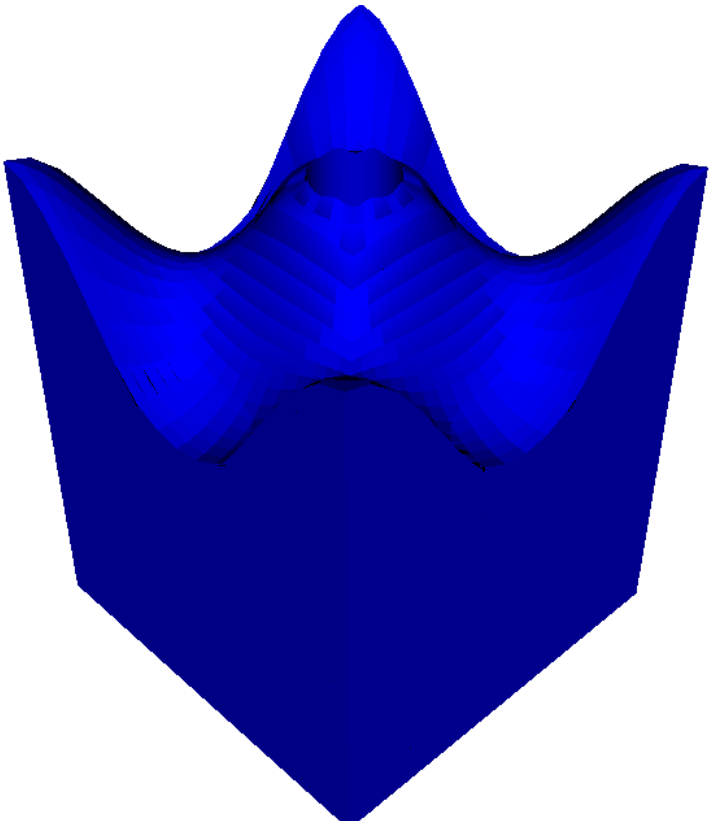
ISOGEOMETRIC POTENTIAL FLOW ANALYSIS OF LINEAR SURFACE WAVES

J.H.A. MEIJER



Maritime and Transportation Technology
Mechanical, Maritime and Materials Engineering
TU Delft

October 2018

SUPERVISORS:
Dr. ir. I. Akkerman
Ir. M.F.P. ten Eikelder
Prof. dr. ir. A.P. van 't Veer

LOCATION:
Delft

ISOGEOMETRIC POTENTIAL FLOW ANALYSIS OF LINEAR SURFACE WAVES

J.H.A. MEIJER

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on November 16, 2018 at 13:30

Student number:	4171292	
Thesis committee:	Prof. dr. ir. A. P. van 't Veer,	TU Delft
	Dr. ir. I. Akkerman,	TU Delft, supervisor
	Ir. M. F. P. ten Eikelder,	TU Delft, supervisor
	Dr. ir. A. A. Kana,	TU Delft

As the famous philosopher Jagger once said
'You can't always get what you want'.

— Dr. Gregory House

You're right, but as it turns out
'If you try, sometimes you get what you need'.

— Dr. Lisa Cuddy

ABSTRACT

In numerical methods, correct geometry description and mesh refinement are a challenge. By using a more geometrically based Finite Element Analysis (FEA) type method called 'Isogeometric Analysis' (IGA), exact geometry description can be attained, even on coarse meshes. Furthermore, mesh refinement is relatively easy, since no communication with a geometry description is necessary. From a maritime perspective, this method seems to be very interesting. Therefore, a first step towards an all-inclusive IGA framework for potential flow problems of ships and offshore structures is made by using IGA to solve linear free surface waves in a bounded 2D and 3D domain.

The goal of this thesis is twofold. The first goal is finding out which of three weak formulations is best suited for further development. The second goal is testing the advantages of IGA in a potential frame work. A secondary goal is testing MFEM, the C++ finite element library that was used. These goals were reached by testing the three formulations on (1) a sloshing wave, (2) an airy wave and (3) a step wave in a square tank of 1 m^2 and (4) a sloshing wave in a cubic tank of 1 m^3 with a cylinder in the middle. The first two have analytical solutions that can be used for verification and validation, the last two are used to compare standard FEA with IGA.

The three weak formulations are formed by transforming the strong problem definition into three different weak forms. The main difference between these three being the way the boundary conditions are implemented. The first, reduced formulation is formed by combining the free surface boundary conditions, the second, mixed formulation by implementing all three boundary conditions directly and the third decoupled formulation by decoupling the problem into a free surface and an interior part. The first formulation is the simplest, but is hard to extend towards more complicated problems and is therefore used as a reference solution. The mixed and decoupled formulations are more complicated but can be extended.

The first two tests showed that the reduced and mixed formulations have identical results. These results were very accurate: the wave period could be calculated accurately for coarse meshes and, very important, energy was conserved perfectly. The results for the decoupled formulation were significantly worse: more refined meshes were needed to calculate the wave period accurately and the energy showed periodic behaviour. The last two tests demonstrated that IGA offers results comparable to FEA for less degrees of freedom. The more difficult geometry of the fourth problem was much better represented by IGA.

ACKNOWLEDGMENTS

Some people have been very important in the process of writing this thesis and this should not remain without praise. First of all I would like to thank my supervisors Ido Akkerman and Marco ten Eikelder for their support, criticism and experience. Without them, this thesis would have been impossible to complete. Then, I would like to extend my gratitude towards all my friends, housemates and bandmates. I would especially like to thank Gerben Dekker for all those lunch and coffee breaks full conversation about sports, politics and almost any other subject. Finally, and above all, I want to thank the people that have been at the root of it all: my parents.

Moos Meijer, October 29 2018

CONTENTS

1 INTRODUCTION 1
1.1 Previous research . . . 1
1.2 Advantages of IGA in maritime engineering . . . 2
1.3 Reading guide . . . 3

I CONCEPTS AND METHODS

2 AN INTRODUCTION TO INCOMPRESSIBLE POTENTIAL FLOW ANALYSIS 6
2.1 Derivation of the governing equations . . . 6
2.2 Solving the potential flow . . . 8
3 AN INTRODUCTION TO FINITE ELEMENT ANALYSIS 10
3.1 Weak formulation . . . 10
3.2 Discretisation of the domain . . . 11
3.3 Discretisation of the weak formulation . . . 11
3.4 Basis functions . . . 13
4 AN INTRODUCTION TO ISOGEOMETRIC ANALYSIS 16
4.1 A historic perspective on IGA . . . 16
4.2 B-Splines . . . 18
4.3 NURBS . . . 20
4.4 Numerical analysis with NURBS . . . 22

II PROBLEM DEFINITION

5 STRONG AND WEAK FORMS 26
5.1 Strong form . . . 26
5.2 Weak formulation one: reduced form . . . 27
5.3 Weak formulation two: mixed form . . . 28
5.4 Formulation three: decoupled form . . . 29
6 DISCRETISATION OF THE WEAK FORMS 31
6.1 Spatial discretisation . . . 31
6.2 Spatial discretisation of the reduced form . . . 32
6.3 Spatial discretisation of the mixed formulation . . . 32
6.4 Spatial discretisation of the decoupled form . . . 33
6.5 Temporal discretisation . . . 34
6.6 Post processing . . . 35
7 MFEM CODE OVERVIEW 38
7.1 MFEM time solver . . . 38
7.2 Practical limitations of the MFEM package . . . 39

III RESULTS

8 TEST CASE I: TWO-DIMENSIONAL SLOSHING WAVE 42
8.1 Initial conditions . . . 42
8.2 Analytical solution . . . 43
8.3 Results for the three formulations . . . 44

9 TEST CASE II: TWO-DIMENSIONAL AIRY WAVE 51
9.1 Initial conditions . . . 51
9.2 Analytical solution . . . 52
9.3 Results for the three weak formulations . . . 53
9.4 Results for different wavelengths . . . 60
10 TEST CASE III: TWO-DIMENSIONAL STEP WAVE 62
10.1 Initial condition . . . 62
10.2 Test parameters . . . 62
10.3 Results and analysis . . . 63
11 TEST CASE IV: THREE-RDIMENSIONAL SLOSHING WAVE 68
11.1 Meshes . . . 68
11.2 Initial conditions and test parameters . . . 69
11.3 Results . . . 69
11.4 Illustration of the resulting sloshing wave . . . 73
12 CONCLUSIONS AND FUTURE WORK 80

IV APPENDIX

A SLOSHING WAVE 84
A.1 Prove that the initial conditions satisfy the boundary conditions . . . 84
A.2 Derivation of the kinetic and potential energy . . . 84
B AIRY WAVE 87
B.1 Prove that the initial conditions satisfy the boundary conditions . . . 87
B.2 Derivation of the kinetic and potential energy of the airy wave . . . 89
C MFEM CODE DESCRIPTION 91
C.1 Code initialization . . . 91
C.2 Initialization of the gridfunctions . . . 91
C.3 Wave operator . . . 92

BIBLIOGRAPHY 97

LIST OF FIGURES

Figure 1.1 Example of a Maxsurf CAD-model 2
Figure 3.1 Example of a mesh of tetrahedrons 12
Figure 3.2 Piecewise linear interpolation of $y = \cos(x)$ . . 12
Figure 3.3 First order Lagrange functions 14
Figure 3.4 Lagrange polynomials of degree 1, 2 and 3 . . 15
Figure 4.1 Example of a 'spline' 17
Figure 4.2 Example of a B-spline curve 17
Figure 4.3 Examples of B-spline basis functions of order 0, 1 and 2 19
Figure 4.4 B-spline basis function of order 2 20
Figure 4.5 Example of a NURBS curve 21
Figure 8.1 Illustration of the initial value of ϕ for the sloshing wave 42
Figure 8.2 Time traces of potential ϕ at $(x, z) = (0.5, 1)$ - Implicit midpoint solver with $n = 9$ and $\Delta t = 0.05T_p^a$ 45
Figure 8.3 Time trace of E_{pot}, E_{kin} and E_{tot} - mixed form with $n = 9$ and $\Delta t = 0.05T_p^a$ 46
Figure 8.4 Time trace of E_{pot}, E_{kin} and E_{tot} - decoupled form with $n = 9$ and $\Delta t = 0.05T_p^a$ 47
Figure 8.5 Convergence of T_p for the sloshing wave 49
Figure 8.6 Convergence of E_{tot} at $t = t_3$ 50
Figure 9.1 Illustration of the initial value of η for the airy wave 52
Figure 9.2 Time traces of potential ϕ at $(x, z) = (0.5, 1)$ - Implicit midpoint solver with $n = 7$ and $\Delta t = 0.05T_p^a$ 54
Figure 9.3 Time trace of E_{pot}, E_{kin} and E_{tot} - mixed form with $n = 7$ and $\Delta t = 0.05T_p^a$ 56
Figure 9.4 Time trace of E_{pot}, E_{kin} and E_{tot} - decoupled form with $n = 25$ and $\Delta t = 0.0125T_p^a$ 57
Figure 9.5 Convergence of T_p for the airy wave 59
Figure 9.6 Convergence of E_{kin} 59
Figure 9.7 T_p for varying wave number k 61
Figure 10.1 Initial wave height along the free surface for the step wave 63
Figure 10.2 Time trace of E_{pot}, E_{kin} and E_{tot} - mixed form with $n = 17$ and $\Delta t = 0.025$ s 64
Figure 10.3 Convergence plot for $E_{tot}(t = 0)$ 65
Figure 10.4 Time traces of η at $(x, z) = (0.25, 1)$ for three mesh sizes - mixed form 66

Figure 10.5 Convergence of the maximum wave height at $(x, z) = (0.25, 1)$ 67
Figure 11.1 Geometry for the fourth test case: a cubic tank with a monopile 68
Figure 11.2 Examples of the mesh for the tank with monopile 70
Figure 11.3 Time trace of E_{pot}, E_{kin} and E_{tot} - mixed form . 72
Figure 11.4 Convergence plot of E^0_{tot} and ΔE_{tot} 73
Figure 11.5 Time traces of ϕ and η at $(0.25, 0.25, 1)$ - mixed form - refined once 74
Figure 11.6 Time traces of ϕ and η at $(0.25, 0.25, 1)$ - mixed form - refined twice 75
Figure 11.7 Time traces of ϕ and η at $(0.25, 0.25, 1)$ - mixed form - refined thrice 76
Figure 11.8 Convergence plot of η_{max} at (0.25, 0.25, 1.0). . . 77
Figure 11.9 Illustration of the 3D sloshing wave 78

LIST OF TABLES

Table 4.1 Comparison of finite element analysis and iso-geometric analysis based on NURBS [39] . . . 23
Table 8.1 Analytical solution for the sloshing wave . . . 43
Table 8.2 Parameters for the sloshing wave problem . . . 44
Table 8.3 Parameters for the sloshing wave problem . . . 44
Table 8.4 T_p of the sloshing wave 48
Table 8.5 E_{tot} at $t = 3T_p$ of the sloshing wave 48
Table 9.1 Analytical solution for the airy wave 53
Table 9.2 Parameters for the airy wave problem 53
Table 9.3 Parameters for the airy wave problem 54
Table 9.4 T_p of the airy wave 58
Table 9.5 E_{tot} at $t = 3T_p$ of the airy wave 58
Table 9.6 Test matrix for wavelength comparison 60
Table 9.7 Constant parameters for wavelength comparison 61
Table 10.1 Parameters for the step wave problem 63
Table 10.2 Parameters for step wave experiments 63
Table 11.1 Parameters for three-dimensional wave experiments 69
Table 11.2 Parameters for the three-dimensional sloshing wave problem 71
Table C.1 Command line options for the codes 92

ACRONYMS

BEM	Boundary Element Method
CAD	Computer-Aided Design
CAE	Computer Aided Engineering
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
DOFs	Degrees-Of-Freedom
FE	Finite Element
FEA	Finite Element Analysis
FEM	Finite Element Method
LLNL	Lawrence and Livermore National Laboratory
IGA	Isogeometric Analysis
NURBS	Non-uniform Rational B-Splines
PDE	Partial Differential Equation

1 INTRODUCTION

In shipbuilding, prediction of ship resistance and behaviour was, is and always will be one of the main challenges. How to make this prediction as easy, fast and accurate as possible, was, is and always will be one of the main questions. Several methods have been used since the first people sailed the earth: from the risky old school method of trial and error, via model towing tank tests to modern-day Computational Fluid Dynamics (CFD) calculations. Today, most analyses are done numerically. Either with the slower, but more accurate, CFD-solvers or by using faster, but less accurate potential flow analysis methods. In both methods, the correct representation of geometry and mesh refinement of this geometry can be a challenge. Therefore, a first step towards the incorporation of the concept of Isogeometric Analysis (IGA) into shipbuilding is made in this thesis. The ultimate goal is an 'all-inclusive' IGA-framework for potential flow problems of ships and offshore structures.

1.1 PREVIOUS RESEARCH

IGA was introduced by Hughes et al. in 2005 [39, 43]. It is an analysis method that is similar to Finite Element Analysis (FEA), but is more geometrically based. This is achieved by using Non-uniform Rational B-Splines (NURBS) as basis functions and in the mesh definition. NURBS are commonly used in Computer-Aided Design (CAD)-programs like 'Maxsurf' [56] (see figure 1.1) and 'Rhinoceros' [64]. By doing so, the geometry can be represented exactly, even on a coarse mesh. Furthermore, mesh refinement does not change the geometry. This offers several potential advantages over traditional Finite Element Method (FEM).

After its introduction in 2005, the isogeometric analysis methodology has gained a lot of interest by researchers. This has lead to a firm groundwork of the approach. The mathematical study of NURBS-based IGA can be found in [5]. Other theoretical contributions include the development of IGA with T-splines [9, 25], with LR B-splines [44], (hierarchical) refinement of IGA [13, 29, 58, 66, 71] and the contruction of discrete differential forms [16] and stable elements [17, 19]. Several succesful algorithms have been proposed to speed up the computation time [3, 33, 40, 65].

In the recent years the isogeometric analysis methodology has been succesfully used in various fields of engineering. Without trying to be complete, a few are mentioned: electromagnetics [18, 70], blood flow

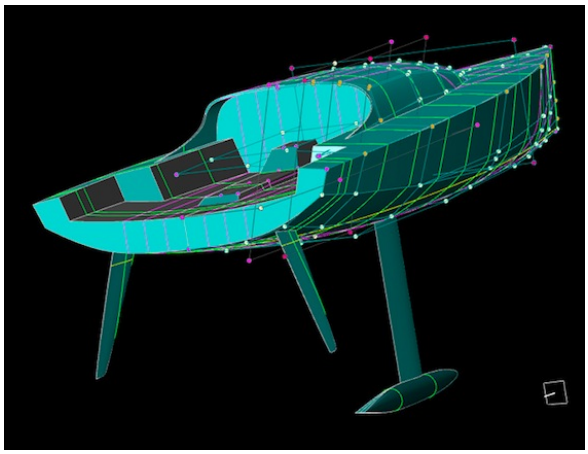


Figure 1.1: Example of a Maxsurf CAD-model [24]

and heart-valve computations [6, 8, 36, 37, 45, 73], shape optimization [12, 21, 62, 67, 72] and structural mechanics and fatigue [10, 11, 57, 59, 68]. In the maritime field, isogeometric analysis is used for fluid flow computations and fluid-structure interaction. The isogeometric methods are often employed with the variational multiscale (VMS) paradigm [26, 27, 38, 41, 42]. For applications in incompressible turbulence see e.g. [1, 4, 7, 23, 27, 35, 47, 51, 63], for free surface flow and FSI consult [2, 48, 51].

Isogeometric analysis has been used in conjunction with Boundary Element Method (BEM). For instance, Belibassakis et al. developed an isogeometric BEM to calculate the wave-resistance in [14] and [15]. The method is used to calculate the wave resistance of a prolate spheroid, a Wigley hull and a Series 60 hull. Other examples can be found in [31, 49, 50, 61].

1.2 ADVANTAGES OF IGA IN MARITIME ENGINEERING

In the maritime world, CAD-programs are very often used to design ships and other structures. NURBS are very well equipped to represent the rounded and fluent shapes that are omnipresent in maritime engineering. In order to predict the behaviour of the design, numerical calculations of (for instance) the resistance and hydrodynamic coefficients are often performed. Numerical calculations need some kind of numerical version of the geometry in the form of a 'mesh'. However, rounded shapes are hard to represent with traditional mesh types such as tetrahedrons. Usually, they can only be approximated. This results in not-exact geometries, the need to go back to the original CAD-model if the mesh needs to be refined and a changing geometry as the mesh is refined. Using the techniques of IGA, that actually originate from shipbuilding, could solve these problems. Possible applications are for instance structural mechanics, CFD-calculations and potential flow

analysis. In this thesis a first step in the direction of potential flow analysis is made. A possible example of an application would be a wave resistance estimation code that can be used for hull optimization.

1.3 READING GUIDE

This thesis is divided into three parts. The *first part* focusses on the introduction of concepts that are used: potential flow (chapter 2), finite element analysis (chapter 3) and isogeometric analysis (chapter 4). These three concepts are complex enough for years of studying, so descriptions are far from complete and only serve as a short introduction. In the *second part*, these concepts are used to solve the problem: the estimation of linear free surface waves in a bounded domain. Three different weak forms are derived and explained in chapter 5. These weak forms are discretised in space and time in chapter 6. In chapter 7, some limitations of the used MFEM package are explained. The results are presented and analysed in the *third part*. The three formulations are tested on several two and three dimensional problems in chapters 8 to 11 to see the differences between the formulations and to see the difference between IGA and 'standard' FEA. The report ends with a chapter dedicated to the conclusions and by giving some recommendations for future work.

Part I

CONCEPTS AND METHODS

The methods that are used in this thesis are introduced in this part. These three methods are potential flow analysis (chapter 2), finite element analysis (chapter 3) and isogeometric analysis (chapter 4). Since all three subjects are big enough for complete books and years of studying, only a practical introduction is given to them so the reader can understand them well enough to understand the way the results are obtained.

2

AN INTRODUCTION TO INCOMPRESSIBLE POTENTIAL FLOW ANALYSIS

Potential flow is a relatively simple model for fluid flow that is obtained by assuming *irrotationality*. With this assumption, it is possible to come to analytical solutions of several flow problems. Furthermore, it can be used to set up numerical models that are cheaper than the more complicated CFD-models. However, viscous effects are neglected. These characteristics lead to a model that is very useful in some applications, but also has its limitations. For instance: it can be used to estimate the wave-making resistance of a ship hull or the lift produced by an airplane wing. It could also be used for optimization purposes. However, viscous effects like drag cannot be determined.

In this chapter, a short introduction to incompressible potential flow analysis is given. Many articles and books are written on this topic (see for instance [46]), so the interested reader is referred there. The derivation of the governing equations is given in the first paragraph. Several ways to solve these equations are given in the second paragraph.

2.1 DERIVATION OF THE GOVERNING EQUATIONS

Incompressible, Newtonian fluids with constant viscosity can be described with the incompressible (so density ρ and viscosity μ are constant) version of the Navier-Stokes equations [46, p. 16]:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2.1a}$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) = -\nabla p + \mu \nabla^2 \boldsymbol{u} + \rho \boldsymbol{f} \tag{2.1b}$$

where $\boldsymbol{u}$ is the velocity vector of the fluid in the direction of the principal axes in m/s, ρ is the density of the fluid in kg/m^3, t is the time in s, p is the pressure N/m^2, μ is the viscosity coefficient in N·s/m^2 and $\boldsymbol{f}$ is the vector of the volume forces in the direction of the three principal axes in N.

From dimensional analysis, the non-dimensional *Reynolds* number, which is a measure of the ratio between the inertial and viscous forces, can be formulated:

$$Re = \frac{\rho V L}{\mu}, \tag{2.2}$$

where V and L are a reference speed and length. If this Reynolds number is sufficiently large, the viscous terms are very small compared to the inertial ones. Usually, this is the case in the turbulent flows

around ships. However, the shear flow derivative ($\nabla^2 u$) is large near solid boundaries, so the $\mu\nabla^2 u$ term cannot be neglected there. The flow is therefore often divided into two regions: the outer flow region, where viscous effects are small, and the boundary layer, where they are not.

It can be shown that, for high Reynolds numbers, vorticity ξ

$$\xi = \nabla \times u$$

is only generated in the small boundary layer and that it is convected along with the flow faster than it can be diffused into the outer region. Therefore, if the outer region is rotation-free, it will remain that way. So:

$$\nabla \times u = 0$$

or

$$\frac{\partial w}{\partial y} = \frac{\partial v}{\partial z}, \qquad \frac{\partial u}{\partial z} = \frac{\partial w}{\partial x} \qquad \text{and} \qquad \frac{\partial v}{\partial x} = \frac{\partial u}{\partial y}.$$

This means the integral of the velocity along an arbitrary line C is the exact differential of a potential ϕ:

$$\int_C \boldsymbol{u} \cdot d\boldsymbol{l} = \int_C u dx + v dy + w dz = \phi(x, y, z).$$

So the velocity is equal to the gradient of this potential:

$$\boldsymbol{u} = \nabla\phi \tag{2.3}$$

This can be substituted in the continuity equation for incompressible fluids (2.1a) to form the linear differential equation known as *Laplace's equation* that can be used to obtain the velocity field of an irrotational, inviscid, incompressible flow:

$$\nabla \cdot \boldsymbol{u} = \nabla \cdot \nabla\phi = \nabla^2\phi = \Delta\phi = 0.$$

By applying the right boundary conditions, the velocity field of a certain problem can be obtained. For instance, finding the flow that results from a solid body B moving with velocity $\boldsymbol{u}_B$ through an infinite fluid. In this case, a no-penetration and a far-field boundary condition are applied to obtain the so-called *Neumann exterior problem*:

Find ϕ such that:

$$\delta\phi = 0, \tag{2.4a}$$

$$\frac{\partial\phi}{\partial n} = \boldsymbol{n} \cdot \boldsymbol{u}_B, \qquad \text{on } \Gamma_B \tag{2.4b}$$

$$\nabla\phi = 0. \qquad \text{at infinity} \tag{2.4c}$$

where $\boldsymbol{n}$ is the normal vector of body B.

The pressure in an inviscid incompressible fluid can be determined with *Bernoulli's equation*:

$$E + \frac{p}{\rho} + \frac{1}{2}\boldsymbol{u}^2 + \frac{\partial \phi}{\partial t} = C(t), \tag{2.5}$$

So the value of the left-hand side of (2.5) is a function of time only. Bernoulli's equation can be derived from *Euler's equation*:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = \boldsymbol{f} - \frac{\nabla p}{\rho} \tag{2.6}$$

and by assuming only conservative body forces $\boldsymbol{f}$ act on the body, so they can be written as

$$\boldsymbol{f} = -\nabla E. \tag{2.7}$$

If the body force is, for instance, gravity, then this equates to $E = -gz$.

2.2 SOLVING THE POTENTIAL FLOW

Since the Laplace equation (2.4a) is a linear differential equation, the linear combination ϕ_{sum} of two solutions ϕ_1 and ϕ_2 is also a solution:

$$\phi_{\text{sum}} = \sum_{i=1}^{2} c_i \phi_i$$

This property can be used to form new solutions by combining basic solutions like 'point sources', 'point doublets', 'polynomials' and 'vortices' [46, p. 58-71]. This is a way to obtain analytical solutions and has, for instance, been used to calculate the lift of wings in the early days of aeronautics. However, this is usually only possible after simplifying the geometry and the boundary conditions.

In order to be able to solve for more complex problems, numerical techniques have been developed. An example is the 'boundary element' or 'panel' method. In the panel method, singular elements, e.g. sources σ and doublets μ, are distributed over the surface of boundary S_B:

$$\phi_{\text{tot}} = -\frac{1}{4\pi} \int_{S_B} \left(\sigma \frac{1}{r} - \mu \boldsymbol{n} \cdot \nabla \frac{1}{r} \right) dS + \phi_\infty \tag{2.8}$$

where ϕ_∞ is the free-stream potential.

Equation (2.8) is reduced to a set of linear algebraic equations by dividing the body S_B into N surface panels and N_w wake panels on each of which the boundary condition is specified on a 'collocation point'. This point is, for instance, the center of the panel. For each collocation point a linear algebraic equation with N unknown sources and doublets can be set up:

$$\sum_{i=1}^{N} A_i \mu_i = -\sum_{i=1}^{N} B_i \sigma_i.$$

This can be done for all N collocation points, which leads to a matrix-vector system of the form

$$\boldsymbol{A}\boldsymbol{\mu} = -\boldsymbol{B}\boldsymbol{\sigma}.$$

where matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ contain the *influence coefficients*. The solution of the the Laplace equation is not unique from a mathematically point of view: the combination of sources and doublets is arbitrary and the amount of circulation around surface S_B has to be fixed based upon physical considerations. Physical problems can then be modelled by applying (a mix of) Neumann and Dirichlet boundary conditions and solving the system numerically.

FEM is another possible numerical method that can be used to solve potential flow. The big difference with BEM is that the complete domain is discretised instead of only the boundaries. This results in larger matrices and therefore, usually, in more expensive computations. However, the matrices that result from BEM are 'fuller' than those in FEM, which can counter this characteristic. FEM will be described in more detail in the following chapter.

3

AN INTRODUCTION TO FINITE ELEMENT ANALYSIS

Finite Element Analysis (FEA) is a numerical method that can be used to solve engineering and mathematical problems numerically. It is applied in many fields. A few examples are structural analysis, option pricing, fluid-structure interactions and flow problems [28, p. v]. Complete studies and books are dedicated to this subject, see for instance [28, 74]. A short introduction to the practical side of FEA is given in this chapter.

In general, FEA can be divided in four steps:

1. Write the physical problem in a mathematical form, usually as a Partial Differential Equation (PDE).
2. Discretise the derived mathematical form and the geometry.
3. Solve the resulting discretised equations.
4. Post-process the results to asses them.

In this chapter, the second step in this process is explained. In the first paragraph the concept of *weak formulations* is introduced and explained. The second paragraph gives a short introduction on the discretisation of the problem domain into elements. How the weak formulation in the first paragraph can be discretised is explained in the third paragraph. In the last paragraph, some examples of *basis functions* that are used for this discretisation are given.

3.1 WEAK FORMULATION

The most common approach to model engineering problems are Partial Differential Equations (PDEs). For instance, the Laplace problem:

$$-\Delta\phi = f. \tag{2.4a}$$

These governing equations are usually referred to as the *strong form*. The strong form is continuous and therefore needs to be discretised in order to solve it numerically. For many problems, it is not possible to formulate a general solution to this strong form. The strong form is therefore transformed to a so-called *weak form* for which accurate approximate solutions can be found. By doing this, the boundary conditions are incorporated and the needed order of continuity of the elements is reduced. The strong form can be transformed into a weak form in four steps [74, p. 47]:

1. Multiply each PDE with a (weighting) function.
2. Integrate this product over the domain of the problem.
3. Reduce the maximum order of derivatives in this integral by integration by parts.
4. If possible, introduce the boundary conditions.

As an example, this is done for the Laplace equation (2.4a). First, it is multiplied by a arbitrary weighting function w that can be chosen conveniently:

$$w \cdot -\Delta\phi = w \cdot f.$$

This is then integrated over the domain Ω:

$$\int_\Omega w \cdot (-\Delta\phi) d\Omega = \int_\Omega w \cdot f d\Omega.$$

Using integration by parts and assuming that the boundary terms are 0, this can be written as:

$$\int_\Omega (\nabla w) \cdot (\nabla\phi) d\Omega = \int_\Omega \cdot w \cdot f d\Omega. \tag{3.1}$$

3.2 DISCRETISATION OF THE DOMAIN

In order for a computer to be able to approximate a continuous mathematical problem with infinite degrees of freedom, it needs to be discretised on a mesh in order to get a finite number of points. In finite element methods a certain physical domain, a tank of water for instance, is subdivided into *elements*. The simplest form of these elements are lines (one dimension), triangles (two dimensions) and tetrahedrons (three dimensions, see figure 3.1). Together, these elements form the mesh. The degrees of freedom are then associated with the nodes of this mesh. The continuous problem can then be approximated by numerical computations of these degrees of freedom.

The generation of good quality meshes is a complicated process and is a complete study on its own. Some reviews of mesh generation techniques are given in [34, 52].

3.3 DISCRETISATION OF THE WEAK FORMULATION

The weak formulation can be discretised by approximating it with the *interpolation operator* [28]:

$$g(\boldsymbol{x}) \approx g^h(\boldsymbol{x}) \equiv \sum_{i=1}^{n} \gamma_i N_i(\boldsymbol{x}). \tag{3.2}$$

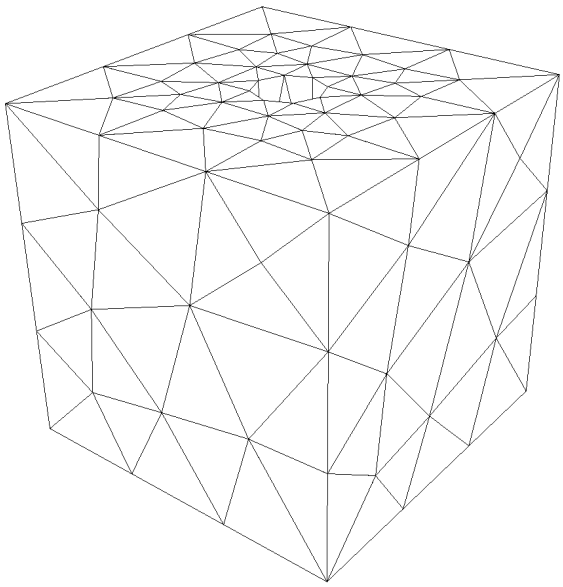


Figure 3.1: Example of a mesh of tetrahedrons

Here, $g^h(\boldsymbol{x})$ is the approximation of the continuous function $g(\boldsymbol{x})$, n is the number of mesh points, γ_i are the global degrees of freedom and $N_i(\boldsymbol{x})$ are the so called basis functions. There are many possible basis functions, see for some examples paragraph 3.4. The simplest example is probably a piecewise linear function with coefficients $\gamma_i = g(x_i)$, where x_i is the i^{th} node in the mesh. An example of such an discretisation is shown in figure 3.2. Here, a simple cosine is sampled on a mesh of 10 equally spaced points and interpolated by assuming the change in y is linear with x.

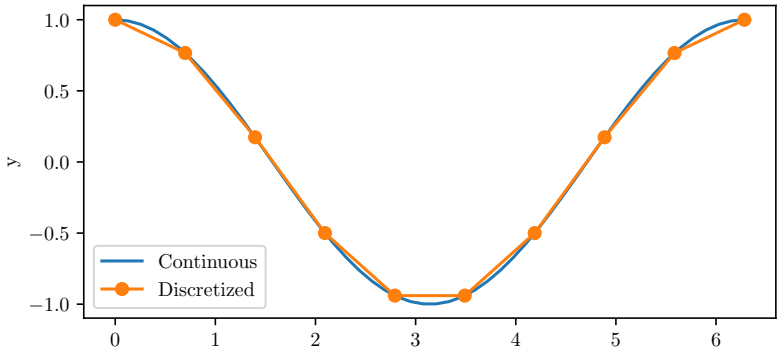


Figure 3.2: Piecewise linear interpolation of $y = \cos(x)$

Using the interpolation operator (3.2) to approximate ϕ with ϕ^h in the weak form of the Laplace equation (3.1), results in:

$$\phi \approx \phi^h \equiv \sum_{i=1}^{n} \phi_i N_i(\boldsymbol{x}). \tag{3.3}$$

Equation (3.1) must hold for all possible weightings w, so it can be chosen freely. [1] Therefore, it is defined as follows:

$$w \equiv N_i(\boldsymbol{x}). \tag{3.4}$$

Substituting (3.3) and (3.4) into (3.1) results in:

$$\int_\Omega \left(\nabla \sum_{i=1}^{n} \phi_i N_i(\boldsymbol{x}) \right) \cdot \left(\nabla N_j(\boldsymbol{x}) \right) d\Omega = \int_\Omega f(\boldsymbol{x}) N_j(\boldsymbol{x}) d\Omega$$

Swapping the summation and integration results in n equations of the form:

$$\phi_i \int_\Omega \left(\nabla N_i(\boldsymbol{x}) \cdot \nabla N_j(\boldsymbol{x}) \right) d\Omega = \int_\Omega f(\boldsymbol{x}) N_j(\boldsymbol{x}) d\Omega.$$

This can be written in matrix form:

$$\boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{b},$$

where the entries of matrix $\boldsymbol{A}$, A_{ij}, are given by

$$A_{ij} = \int_\Omega \nabla N_i(\boldsymbol{x}) \cdot \nabla N_j(\boldsymbol{x}) d\Omega,$$

the entries of vector $\boldsymbol{b}$ are given by

$$b_i = \int_\Omega f(\boldsymbol{x}) N_i(\boldsymbol{x}) d\Omega,$$

and entries of vector $\boldsymbol{\phi}$ are ϕ_i. This system can be solved with linear algebra.

3.4 BASIS FUNCTIONS

For the discretisation with the interpolation operator (3.2), a basis function $N_i(x)$ is needed. This function serves as a trial solution or approximating function. Many basis functions are available. Examples are Lagrange, Crouzeix-Raviart, Raviart-Thomas, Nédélec [28, p. 25-30] and Legendre functions.

For figure 3.2, the piecewise linear $\mathbb{P}_1$ Lagrange finite elements are used. These are defined elementwise for $i \in \{0, ..., n+1\}$ as [28]:

$$N_i(\boldsymbol{x}) = \begin{cases} \frac{1}{h_i - 1}(x - x_{i-1}), & \text{if } x \in I_{i-1} \\ \frac{1}{h_i}(x_{i+1} - x), & \text{if } x \in I_i \\ 0. & \text{otherwise} \end{cases} \tag{3.5}$$

1 This is not completely true: there are mathematical requirements that have to be met. All these requirements are met by the basis functions used in this thesis.

Here, x_i are the vertices of a certain mesh, I_i is the interval $[x_i, x_{i+1}]$ and $h_i = x_{i+1} - x_i$ is the grid size. An example of the resulting 'hat-functions' is given in figure 3.3 for the vertices $x_i = [0, 0.5, 1, 2, 4]$.

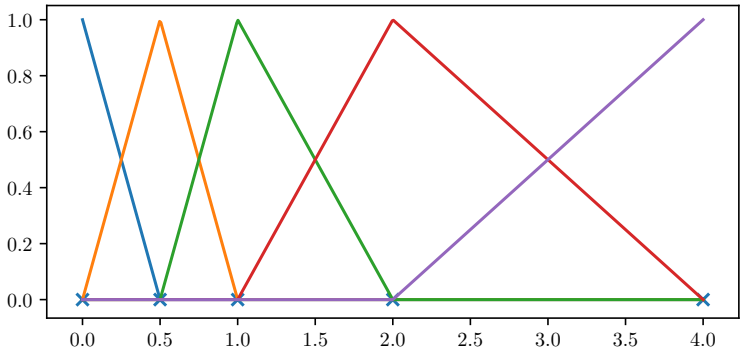


Figure 3.3: First order Lagrange functions

Higher order Lagrange polynomials are shown in figure 3.4 on the interval $[0, 1]$ with $k + 1$ equally spaced nodes (each extra degree also needs an extra node).

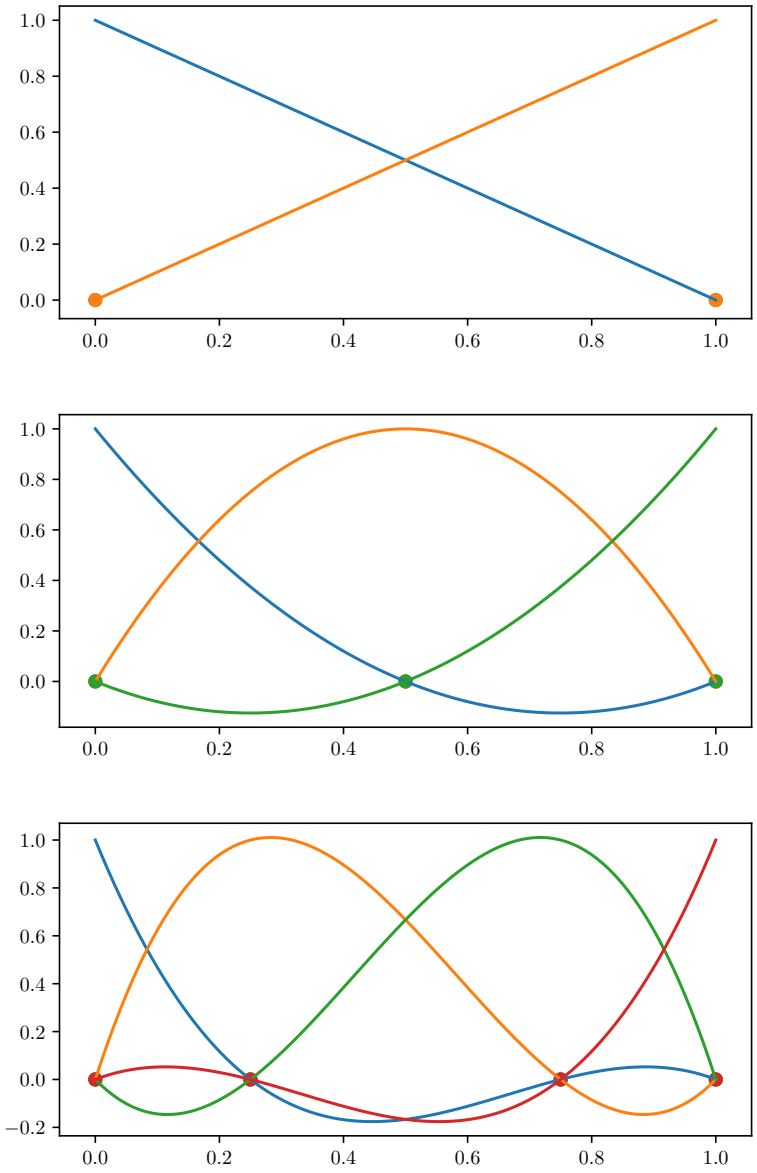


Figure 3.4: Lagrange polynomials of degree 1, 2 and 3

4

AN INTRODUCTION TO ISOGEOMETRIC ANALYSIS

In this chapter, the isogeometric analysis concept is introduced. A more extensive description can be found in literature such as the 2005 paper by Hughes [39] and the book 'Isogeometric Analysis: Toward Integration of CAD and FEA' by Cottrell et al. [43].

In the first paragraph, a short history of IGA is given. B-splines, functions that stand at the basis of NURBS, are explained in the second paragraph. The third paragraph is dedicated to the explanation of NURBS and its characteristics. How these B-splines and NURBS are used to discretise PDEs and geometry is explained in the last paragraph.

4.1 A HISTORIC PERSPECTIVE ON IGA

In the days before the advance of computers, ships were designed by hand. In order to be able to draw continuous, fluent hull shapes, so-called 'splines' were used. These were flexible strips of timber or steel that were bent and held in place by weights, such as in figure 4.1 [69]. The uniform distribution of stress in the spline results in smooth lines that can be used to draw fair hull shapes with low drag. Making a big leap forward to mid-twentieth century, the first days of the computer, car manufacturers like Citroën, Renault and General Motors were looking for ways to store and communicate car shapes. Mathematicians de Casteljau, Bézier and De Boor came up with ways to do just this in the form of Casteljau's algorithm, Bézier curves and B-splines (figure 4.2). These are mathematically quite complex, but are very simple to use by designers. The shape of the curves can be altered by moving 'control points' around in a way that is similar to the physical weights that were used in splines. In the 1980's, NURBS were developed by aerospace company Boeing. NURBS are basically a generalization of B-splines. Since then, it has become the standard in CAD-programs like 'Maxsurf' [56] and 'Rhinoceros' [64].

With the advance of computers, also Computer Aided Engineering (CAE) came up: engineers wanted to be able to solve partial differential equations numerically. One way to do this is Finite Element Analysis (FEA). The development of FEA started in the 1950s and 1960s, so approximately 20 years before NURBS were developed. A numerical description of geometry is at the basis of FEA, but the geometric description is very different than in CAD. In the present day, a typical work flow of an engineer therefore includes the generation of a geometric representation that can be used in FEA from a representation made in a CAD-program. Sometimes this generation can be automated,

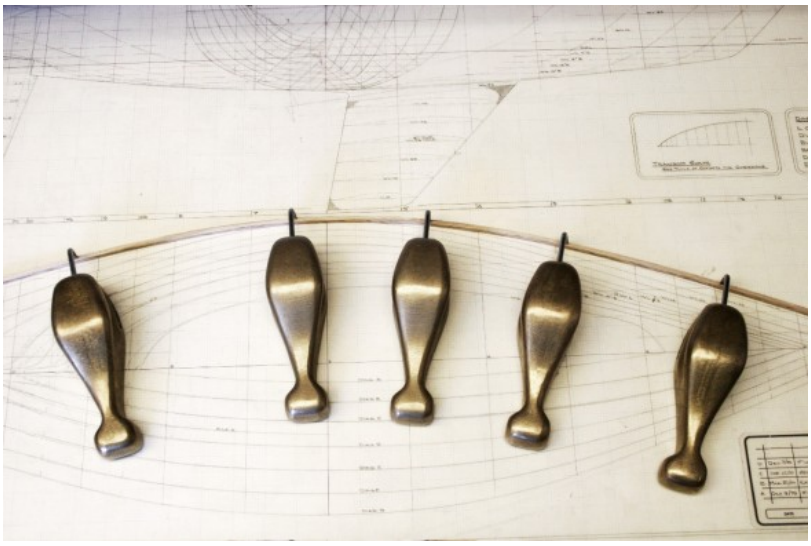


Figure 4.1: Example of a 'spline' [69]

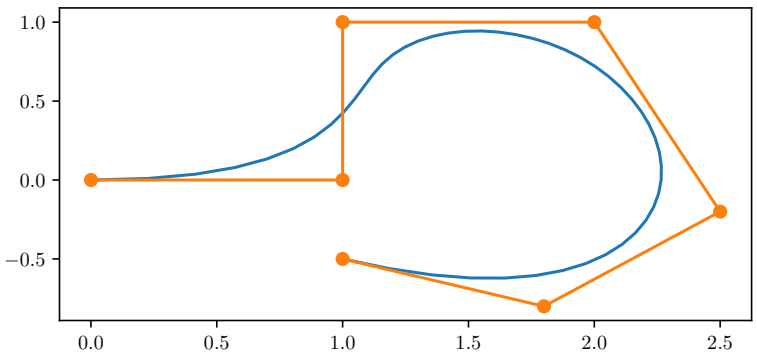


Figure 4.2: Example of a B-spline curve

but most of the times semi-automatically is the best option. Hughes [39] estimates "that about 80% of the overall analysis time is devoted to mesh generation". Transforming a CAD geometry to one that is suitable for FEA usually results in some kind of geometry approximation and the need for communication with the CAD-model if the mesh needs to be refined. Hughes argues that, since the size of the CAE industry is about 20% of that of the CAD industry, the best way forward is "to attempt to change FEA into something more CAD-like". Hughes et al. do just this by using NURBS in their method that they called IGA [39].

4.2 B-SPLINES

At the basis of NURBS are B-Spline basis functions and curves. A B-spline curve of degree p is defined as a linear combination of control points $\boldsymbol{P_i}$ and B-spline basis functions $N_{i,p}(x)$:

$$\boldsymbol{C}(u) = \sum_{i=0}^{n} N_{i(u)} \boldsymbol{P}_i. \tag{4.1}$$

The B-spline basis functions $N_{i,p}(x)$ can be calculated recursively as follows:

Let $U = \{u_0, ..., u_m\}$ be a sequence of non-decreasing real numbers ($u_i \leq u_{i+1}$, $i = 0, ..., m-1$). The basis function $N_{i,p}$ is then defined as:

$$N_{i,0}(u) = \begin{cases} 1, & \text{if } u_i \leq u < u_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{4.2a}$$

For p is 1, 2, 3, ..., they are defined by

$$\begin{aligned} N_{i,p}(u) = &\frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) \\ &+ \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \end{aligned} \tag{4.2b}$$

$N_{i,p}$ is the i^{th} B-spline basis function of degree p and u_i is the i^{th} value of the knot vector U. In case (4.2a) yields $0/0$ the function is defined to be 0.

U is called the 'knot-vector' and its entries u_i are the 'knots'. A knot span is the interval between two succeeding knots. Knots do not need to all have distinct values: $U = \begin{pmatrix} 0, & 0, & 0, & 1, & 1, & 1 \end{pmatrix}$ is a perfectly valid (and often used) knot vector where the knots 0 and 1 have multiplicity 3. They also do not need to be spaced uniformly: $U = \begin{pmatrix} 0, & 0, & 1/5, & 6/7, & 1, & 1 \end{pmatrix}$ is also valid. Knot multiplicity is used to control the local continuity of the curve. The knot spacing can be used to increase the resolution locally. Usually, the knot vector is chosen to lie between 0 and 1, but this is not necessary. Furthermore, the first and last value of the knot vector usually have multiplicity $p+1$. This reduces the continuity at the endpoints from C^{p-1} to C^0 and makes the B-spline 'clamped'. Examples of B-spline basis functions for the knot vector $U = \begin{pmatrix} 1/4, & 2/4, & 3/4, & 1 \end{pmatrix}$ of order 0, 1 and 2 are given in figure 4.3. Another example is given for the knot vector $U = \begin{pmatrix} 0, & 0, & 0, & 1/5, & 2/5, & 2/5, & 6/7, & 1, & 1, & 1 \end{pmatrix}$ in figure 4.4.

A spline can be formed from the defined B-spline basis functions with (4.1). The control polygon $\boldsymbol{P}$ has $(m - p - 1)$ points, where m is the number of knots. For instance, figure 4.2 is made by using the

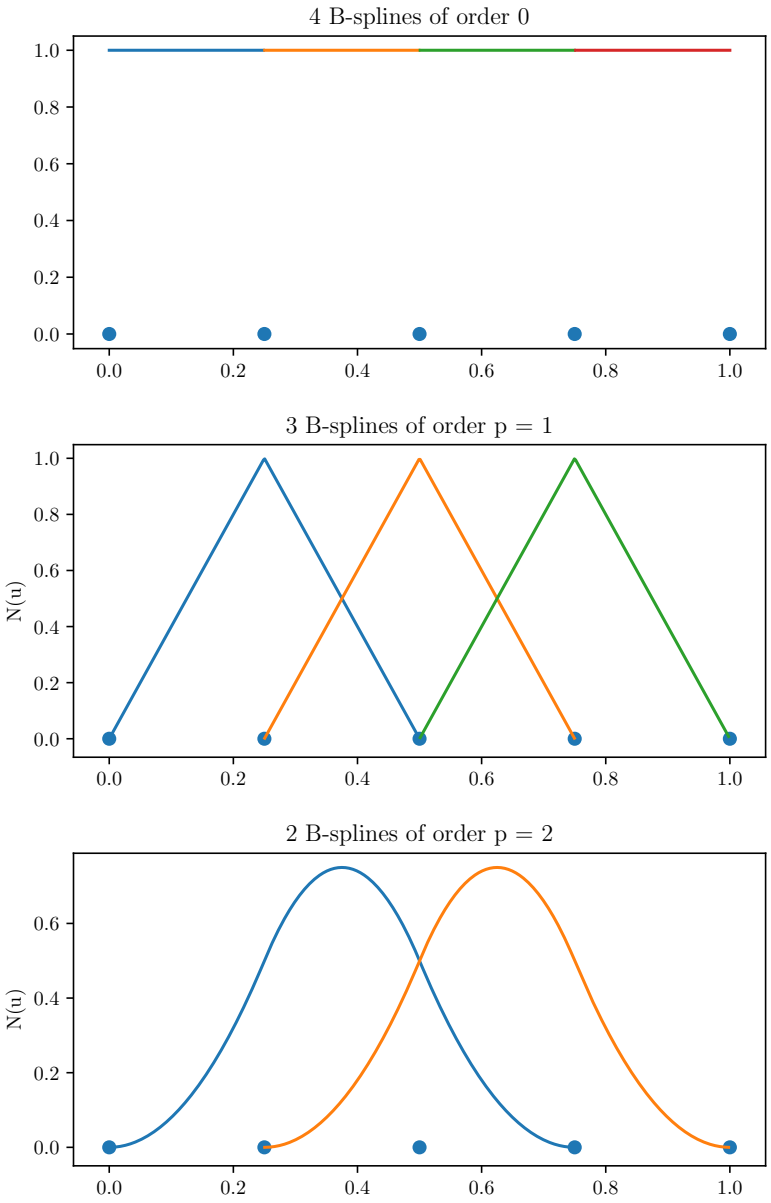


Figure 4.3: Examples of B-spline basis functions of order 0, 1 and 2

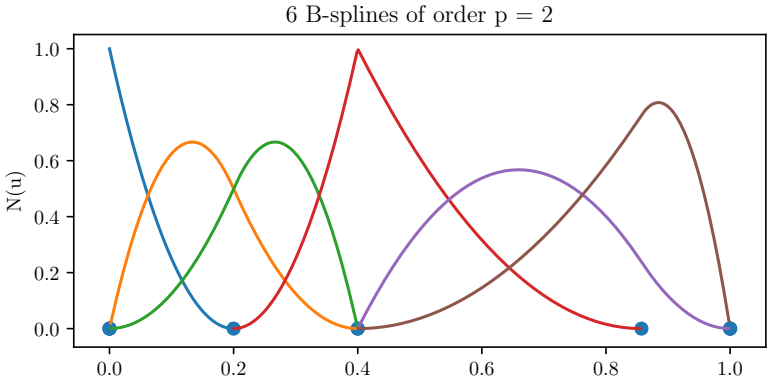


Figure 4.4: B-spline basis function of order 2

B-splines defined by knot vector $U = \{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$, order $p = 3$ and using the control net

$$\boldsymbol{P} = \begin{pmatrix} 0 & 1 & 1 & 2 & 2.5 & 1.8 & 1 \\ 0 & 0 & 1 & 1 & -0.2 & -0.8 & -0.5 \end{pmatrix}.$$

The derivative of a B-spline curve can be determined with

$$\boldsymbol{C}'(u) = \sum_{i=0}^{n} N_{i,p}^{(k)}(u) p \frac{\boldsymbol{P}_{i+1} - \boldsymbol{P}_i}{u_{i+p+1} - u_{i+1}}. \tag{4.3}$$

B-spline surfaces are defined by taking a linear combination of the control points in the control polygon and the tensor product of two basis functions:

$$S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,p}(v) \boldsymbol{P}_{i,j}. \tag{4.4}$$

4.3 NURBS

B-splines have the major drawback of not being able to represent conic sections. This problem was solved by the introduction of *NURBS*. A NURBS-curve is a rational function of B-splines. Rational functions are fractions of which both the numerator and denominator are polynomials. A NURBS-curve is formed with

$$\boldsymbol{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \boldsymbol{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i}, \tag{4.5}$$

where w_i are values called *weights*, usually chosen between 0 and 1. B-spline curves are a subset of NURBS-curves where the weights are all set to 1. An example of such a NURBS curve is plotted in figure 4.5 together with the B-spline curve described previously. Weights

are $w = [1, 0.5, 1, 0.5, 1, 0.5, 1]$ and the same knot vector and control polygon are used. It is clearly visible that the NURBS-curve is further away from points with smaller weights and closer to points with weight 1 with respect to the B-spline-curve. This offers quite intuitive CAD-modelling.

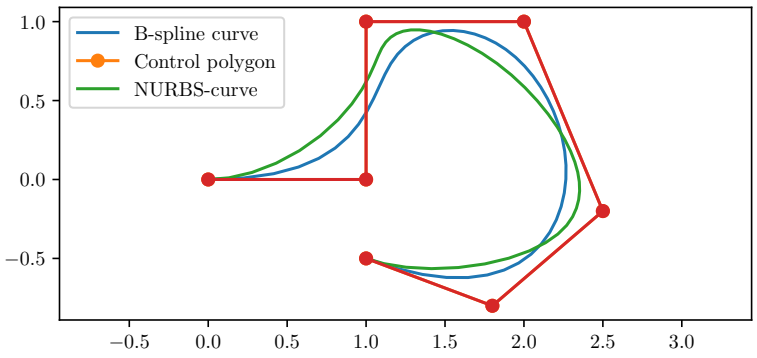


Figure 4.5: Example of a NURBS curve

NURBS-surfaces can be calculated in similar fashion:

$$\boldsymbol{S}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}\boldsymbol{P}_i}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)} \tag{4.6}$$

Some characteristics of NURBS and B-splines are mentioned in [32] and [60]:

1. Local support: $N_{i,p}(u) = 0$ if u is outside of the half open interval $[u_i, u_{i+p+1})$. This property makes it possible to deform the geometry locally. If one control point is moved or its weight changed, it only affects the curve on the interval $u \in [u_i, u_{i+p+1})$.
2. Within a given knot span $[u_j, u_{j+1})$ at most $p + 1$ B-splines are non-zero.
3. Non negativity: the basis splines are never smaller than 0.
4. All derivatives of $N_{i,p}(u)$ exist in the interior of a knot span. At a knot, it is $p - k$ times continuously differentiable, with k the multiplicity of a knot.
5. If the B-splines are clamped they form a partition of unity: $\sum_{i=0}^{n} N_{i,p} = 1$
6. The B-splines attain exactly one maximum value, except if $p = 0$.
7. The basis functions always form a linear independent basis.

8. Variation diminishing property: if the degree of a B-spline is increased, a change in the control points has a smaller influence on the curve.

These characteristics lead to the following geometric characteristics of NURBS-curves:

1. The first and last points of the NURBS-curve coincide with the first and last points of the control polygon.
2. An affine transformation of the curve can be applied by applying the transformation to the control points.
3. The curve can be differentiated infinitely in the interior of a knot span and $p - k$ times at a knot with multiplicity k.
4. Variation diminishing: the curves always lie in the convex hull spanned by the control polygon and the curve always 'wiggles' less then the control polygon.
5. No line or plane has more intersections with the curve than with control polygon.

4.4 NUMERICAL ANALYSIS WITH NURBS

A NURBS-mesh can be formed by defining NURBS-curves (2D) or -surfaces (3D) and use these to construct a coarse mesh of two or three dimensional NURBS-elements. By doing this, the domain is divided into elements by the knot spans. The geometry of the domain can then be formed by defining the control points. The unknown fields (in this case potential ϕ_i and wave height η_i, but in other problems these could for instance be temperature or displacement) are approximated with (3.2) with the basis functions that are also used for the geometry definition. This is known as the *isoparametric* concept. The coefficients in the control polygon $\boldsymbol{P}$ are the control variables, or degrees-of-freedom. The matrices and arrays that were formed in paragraph 3.3 for FEA, can be formed in the same way for NURBS meshes.

The variation diminishing property of NURBS ensures that *Gibbs phenomena* are not present in IGA. Gibbs phenomena are the oscillations in the finite element interpolation that occur when the interpolation function is used to fit discontinuous data. A nice comparison between similar and dissimilar concepts of FEA and IGA is made in table 1 of [39], which is repeated in this thesis in table 4.1.

FINITE ELEMENT ANALYSIS	ISOGEOMETRIC ANALYSIS
Nodal points	Control points
Nodal variables	Control variables
Mesh	Knots
Basis interpolates nodal points and variables	Basis does not interpolate control points and variables
Approximate geometry	Exact geometry
Gibbs phenomena	Variation diminishing
Subdomains	Patches
Compact support	
Partition of unity	
Isoparametric concept	
Affine covariance	
Patch tests satisfied	

Table 4.1: Comparison of finite element analysis and isogeometric analysis based on NURBS [39]

Part II

PROBLEM DEFINITION

The methods described in the first part are used to formulate the linear, free surface potential flow problem in this second part. The problem is first defined in a strong form and then turned into three different weak forms in chapter five. To solve these weak forms numerically, they are discretised in space and time in chapter six.

5

STRONG AND WEAK FORMS

In this thesis, the main problem is the calculation of free surface waves in a tank via potential theory, introduced in chapter 2. By using potential theory, viscous effects are neglected. As explained in chapter 3, the way to do this in FEA and IGA is write a governing set of equations in the form of partial differential equations, usually called the *strong form*. To solve this strong form numerically, it is written in a weak form that is subsequently discretised.

Three different weak forms are discussed: a *reduced* form, a *mixed* form and a *decoupled* form. The reduced form is quite simple to solve, but hard to expand towards (for instance) non-linear waves and will serve as a reference solution, the decoupled formulation is often used in literature and the mixed formulation offers a different approach to the problem. The difference between these three is the way the boundary conditions are imposed. Behaviour of the three formulations will be compared in later chapters to see the differences and to see which one can be best used for further development.

5.1 STRONG FORM

The theory of potential flow is explained in chapter 2. As is shown there, the main equation in potential flow is the Laplace equation (2.4a) that needs to be satisfied on the whole domain Ω of the problem:

$$\Delta\phi = 0.$$

In order to define the problem, boundary conditions need to be defined. Two different boundary types are defined: the free surface boundary Γ_{fs} and the remaining, non free surface boundary Γ_{nfs}. Boundary conditions can be set up by assuming pressure is constant across the free surface and that a particle on the free surface will stay there. Furthermore, it is not possible for fluid to go through watertight boundaries.

A dynamic boundary condition is imposed to ensure pressure is constant across the free surface Γ_{fs}. This is done using Bernoulli's equation (2.5). Setting $E = g\eta$, $p/\rho = C(t)$ and noting that $u^2 = \nabla\phi \cdot \nabla\phi$, results in the dynamic boundary condition

$$\phi_t + \frac{1}{2}\nabla\phi \cdot \nabla\phi + g\eta = 0,$$

where ϕ_t the partial derivatives of ϕ with respect to time and g is the gravitational acceleration.

A particle at the free surface stays at the surface, if the normal velocity of a particle on this free surface is equal to the normal velocity of the surface. This can be used to form the kinematic boundary condition:

$$\eta_t + \nabla_2\phi\nabla_2\eta - \phi_z = 0.$$

where η_t is the partial derivatives of η with respect to time, ϕ_z is the partial derivative of ϕ in vertical direction and ∇_2 is the two-dimensional gradient operator.

On the non free surface part of the boundary Γ_{nfs}, a no-penetration boundary conditions is imposed. This ensures no fluid can go through the walls:

$$\boldsymbol{n} \cdot \nabla\phi = 0$$

The complete strong formulation of the problem is:

$$\Delta\phi = 0 \qquad \text{in } \Omega \tag{5.1a}$$
$$\phi_t + \frac{1}{2}\nabla\phi \cdot \nabla\phi + g\eta = 0 \qquad \text{on } \Gamma_{fs} \tag{5.1b}$$
$$\eta_t + \nabla_2\phi\nabla_2\eta - \phi_z = 0 \qquad \text{on } \Gamma_{fs} \tag{5.1c}$$
$$\boldsymbol{n} \cdot \nabla\phi = 0 \qquad \text{on } \Gamma_{nfs} \tag{5.1d}$$

Here, Ω is the domain of the tank, Γ_{nfs} is the non free surface part of the boundary of the domain and Γ_{fs} is the free surface part of the boundary.

These equations are linearised by assuming the amplitude of the waves is small with respect to the wavelength. The quadratic terms $\nabla\phi \cdot \nabla\phi$ and $\nabla_2\phi \cdot \nabla_2\phi$ are consequently even smaller and can be neglected. This results in the linearised strong form that will be used in the rest of this thesis:

$$\Delta\phi = 0 \qquad \text{in } \Omega \tag{5.2a}$$
$$\phi_t + g\eta = 0 \qquad \text{on } \Gamma_{fs} \tag{5.2b}$$
$$\phi_z = \eta_t \qquad \text{on } \Gamma_{fs} \tag{5.2c}$$
$$\boldsymbol{n} \cdot \nabla\phi = 0 \qquad \text{on } \Gamma_{nfs} \tag{5.2d}$$

5.2 WEAK FORMULATION ONE: REDUCED FORM

The first weak formulation is formed by reducing the number of boundary conditions. This can be done by combining the boundary conditions on the free surface Γ_{fs}: (5.2b) and (5.2c). This is done by differentiating (5.2b) with respect to time, which results in:

$$\phi_{tt} + g\eta_t = 0$$

Using (5.2c) to substitute η_t with ϕ_z, this leads to the following set of equations:

$$\Delta\phi = 0 \qquad \text{in } \Omega \tag{5.3a}$$
$$\phi_{tt} + g\phi_z = 0 \qquad \text{on } \Gamma_{fs} \tag{5.3b}$$
$$\boldsymbol{n} \cdot \nabla\phi = 0 \qquad \text{on } \Gamma_{nfs} \tag{5.3c}$$

This strong form can be written in weak form by multiplying both sides by weighting function w and integrating over Ω. This results in

$$\int_\Omega w(\Delta\phi)d\Omega = 0.$$

The maximum order can be reduced with a rearranged form of Green's first identity:

$$\int_\Omega w\Delta\phi d\Omega = \oint_\Gamma w(\mathbf{n} \cdot \nabla\phi)d\Gamma - \int_\Omega \nabla w \cdot \nabla\phi d\Omega$$

The circle integral around the boundary is equal to the sum of the integral over the free surface Γ_{fs} and the integral over the non free surface boundaries Γ_{nfs}

$$\oint_\Gamma w(\mathbf{n} \cdot \nabla\phi)d\Gamma = \int_{\Gamma_{fs}} w(\mathbf{n} \cdot \nabla\phi)d\Gamma + \int_{\Gamma_{nfs}} w(\mathbf{n} \cdot \nabla\phi)d\Gamma$$

The boundary term on Γ_{nfs} is equal to 0, because of the no-penetration boundary condition (5.3c). The free surface boundary term can be substituted by $-\frac{1}{g}\phi_{tt}$, since it follows from the linearisation that $\boldsymbol{n} \cdot \nabla\phi = \phi_z$ and from (5.3b) it follows that $\phi_z = -\frac{1}{g}\phi_{tt}$. So:

$$\oint_\Gamma w(\mathbf{n} \cdot \nabla\phi)d\Gamma = -\int_{\Gamma_{fs}} w\left(\frac{1}{g}\phi_{tt}\right)d\Gamma$$

The reduced weak form can then be written as:

$$\int_\Omega \nabla w \cdot \nabla\phi d\Omega + \frac{1}{g}\int_{\Gamma_{fs}} w \cdot \phi_{tt} d\Gamma = 0$$

Using inner product notation, the reduced weak form can be written as:
Find ϕ such that

$$(\nabla w, \nabla\phi) + \frac{1}{g}(w, \phi_{tt})_{\Gamma_{fs}} = 0. \tag{5.4}$$

5.3 WEAK FORMULATION TWO: MIXED FORM

A second weak formulation can be formed where the two free surface boundary conditions are not combined. Instead, strong formulation

(5.2) is used. This results in a *mixed* weak formulation in which both the potential ϕ and the wave height η need to be solved.

Again multiplying the Laplace equation with w, integrating over Ω and applying Green's first identity results in

$$\int_\Omega \nabla w \cdot \nabla \phi d\Omega + \frac{1}{g} \int_{\Gamma_{fs}} w \cdot (\boldsymbol{n} \cdot \nabla \phi) d\Gamma = 0.$$

Boundary condition (5.2c) can be used to substitute $\boldsymbol{n} \cdot \nabla \phi$ with η_t. The result is written in inner product notation as

$$(\nabla w, \nabla \phi) - (w, \eta_t)_{\Gamma_{fs}} = 0. \tag{5.5}$$

The other boundary condition on the free surface (5.2b) needs to be satisfied as well. A possible weak form can be obtained by multiplying it by weighting function $\left(\xi + \frac{\alpha}{g} w\right)$ and integrating along the free surface:

$$\left(\xi + \frac{\alpha}{g} w, \phi_t + g\eta\right)_{\Gamma_{fs}} = 0 \tag{5.6}$$

Equations (5.5) and (5.6) multiplied by $1/2$ can be added to obtain the final second weak formulation of the system:

Find ϕ and η such that

$$(\nabla w, \nabla \phi) - (w, \eta_t)_{\Gamma_{fs}} + \frac{1}{2}\left(\xi + \frac{\alpha}{g} w, \phi_t + g\eta\right)_{\Gamma_{fs}} = 0 \tag{5.7}$$

The coefficients $1/2$ and weighting function $\xi + \frac{\alpha}{g} w$ are chosen to obtain good coercivity, boundedness and accuracy. These are numerical concepts that will not be discussed any further in this thesis.

5.4 FORMULATION THREE: DECOUPLED FORM

A third option is decoupling the interior and free surface problem which results in two weak forms instead of one. The interior problem can be written in strong form as:

$$\Delta \phi = 0 \qquad \text{in } \Omega \tag{5.8a}$$
$$\phi = \tilde{\phi} \qquad \text{on } \Gamma_{fs} \tag{5.8b}$$
$$n \cdot \nabla \phi = 0 \qquad \text{on } \Gamma_{nfs} \tag{5.8c}$$

Where $\tilde{\phi}$ is defined on the free surface and follows from the strong form of the free surface problem:

$$\phi_t + g\eta = 0 \qquad \text{on } \Gamma_{fs} \tag{5.8d}$$
$$\eta_t = \phi_z \qquad \text{on } \Gamma_{fs} \tag{5.8e}$$

The weak form associated with the interior problem can be obtained in similar fashion as for the reduced form. However, the free surface

is not part of the interior problem, so the boundary term on the free surface $\left(\int_{\Gamma_{fs}} n \cdot \nabla\phi d\Gamma\right)$ is not part of the equation.

The weak form for the free surface problem can be obtained by multiplying (5.8d) with weighting function w and (5.8e) with weighting function $g^2/\alpha^2\xi$. This is again done to obtain good coercivity, boundedness and accuracy. The resulting weak forms are:

Find ϕ and η such that

$$(\nabla w, \nabla\phi) = 0 \qquad \text{in } \Omega \tag{5.9a}$$

for the interior problem and

$$(w, \phi_t + g\eta)_{\Gamma_{fs}} + \frac{g^2}{\alpha^2}(\xi, \eta_t)_{\Gamma_{fs}} = \frac{g^2}{\alpha^2}(\xi, \phi_z)_{\Gamma_{fs}} \qquad \text{on } \Gamma_{fs} \tag{5.9b}$$

for the free surface problem.

6 DISCRETISATION OF THE WEAK FORMS

In chapter 5 the governing equations were formulated in strong form (5.2). This strong form was transformed into three different weak forms: (5.4), (5.7) and (5.9). The main difference between the three is the way the boundary conditions are imposed. In order to solve these weak formulations with FEA, the weak forms need to be discretised. The discretisation of the variables and weighting functions are defined in the first paragraph and applied to the three weak forms in paragraphs two to four. The implicit temporal discretisation of these equations is defined in the fifth paragraph. To asses the results, a discrete approximation of the energy is used. This is defined in the last paragraph.

6.1 SPATIAL DISCRETISATION

The continuous variables potential ϕ and wave height η are approximated with the interpolation operator (3.2). So:

$$\phi(\boldsymbol{x},t) \approx \phi^h(\boldsymbol{x}) \equiv \sum_{i=0}^{n} \phi_i(t) N_i(\boldsymbol{x}), \tag{6.1a}$$

$$\eta(\boldsymbol{x},t) \approx \eta^h(\boldsymbol{x}) \equiv \sum_{i=0}^{n} \eta_t(t) N_i(\boldsymbol{x}). \tag{6.1b}$$

Here, $N_i(\boldsymbol{x})$ are the basis functions and ϕ_i and η_i are the degrees of freedom. In FEA, the basis functions could be Lagrange polynomials for instance. In IGA, these would be NURBS.

The discrete approximations of the several partial derivatives are defined equivalently:

$$\phi_t(\boldsymbol{x},t) \approx \phi_t^h(\boldsymbol{x},t) \equiv \sum_{i=0}^{n} \dot{\phi}_i(t) N_i(\boldsymbol{x}), \tag{6.1c}$$

$$\phi_{tt}(\boldsymbol{x},t) \approx \phi_{tt}^h(\boldsymbol{x},t) \equiv \sum_{i=0}^{n} \ddot{\phi}_i(t) N_i(\boldsymbol{x}), \tag{6.1d}$$

$$\phi_z(\boldsymbol{x},t) \approx \phi_z^h(\boldsymbol{x},t) \equiv \sum_{i=0}^{n} \phi_i(t) \frac{\partial}{\partial z} N_i(\boldsymbol{x}), \tag{6.1e}$$

$$\eta_t(\boldsymbol{x},t) \approx \eta_t^h(\boldsymbol{x}) \equiv \sum_{i=0}^{n} \dot{\eta}_i(t) N_i(\boldsymbol{x},t). \tag{6.1f}$$

Weighting functions w and ξ are defined as

$$w \equiv N_i(\boldsymbol{x}), \tag{6.2a}$$
$$\xi \equiv N_i(\boldsymbol{x}). \tag{6.2b}$$

These approximations are used to discretise the three weakforms (5.4), (5.7) and (5.9).

6.2 SPATIAL DISCRETISATION OF THE REDUCED FORM

The reduced weak form

$$(\nabla w, \nabla \phi) + \frac{1}{g}(w, \phi_{tt})_{\Gamma_{fs}} = 0. \tag{5.4, repeated}$$

can be discretised with (6.1a) and (6.1d) and by setting w according to (6.2a). This results in

$$\left(\nabla N_i, \nabla \sum_{j=0}^{n} \phi_j N_j\right) + \frac{1}{g}\left(N_i, \sum_{j=0}^{n} \ddot{\phi}_j N_j\right)_{\Gamma_{fs}} = 0.$$

This results in

$$\sum_{j=0}^{n} \phi_j \left(\nabla N_i, \nabla N_j\right)_{\Omega} + \sum_{j=0}^{n} \ddot{\phi}_j \frac{1}{g}\left(N_i, N_j\right)_{\Gamma_{fs}} = 0.$$

which can be written in matrix-vector form as

$$\boldsymbol{K}\boldsymbol{\phi} + \frac{1}{g}\boldsymbol{M}\ddot{\boldsymbol{\phi}} = \boldsymbol{0} \tag{6.3}$$

where the entries of stiffness matrix $\boldsymbol{K}$ are

$$K_{ij} = \left(\nabla N_i, \nabla N_j\right)_{\Omega},$$

and the entries of mass matrix $\boldsymbol{M}$ are

$$M_{ij} = \left(N_i, N_j\right)_{\Gamma_{fs}}.$$

Vectors $\boldsymbol{\phi}$ and $\ddot{\boldsymbol{\phi}}$ have entries ϕ_i and $\ddot{\phi}_i$.

6.3 SPATIAL DISCRETISATION OF THE MIXED FORMULATION

Following the same methodology as for the reduced formulation, the mixed form

$$(\nabla w, \nabla \phi) - (w, \eta_t)_{\Gamma_{fs}} + \frac{1}{2}\left(\xi + \frac{\alpha}{g} w, \phi_t + g\eta\right)_{\Gamma_{fs}} = 0 \quad \text{(5.7, repeated)}$$

can be discretised. This is done with (6.1a), (6.1c), (6.1b) and (6.1f). The bilinear forms in (5.7) can be written as

$$
\begin{aligned}
(\nabla w, \nabla \phi) &\approx \sum_{i=0}^{n} \phi_i \left(\nabla N_i, \nabla N_j\right) = \boldsymbol{K} \cdot \boldsymbol{\phi}, \\
(w, \eta_t)_{\Gamma_{fs}} &\approx \sum_{i=0}^{n} \dot{\eta}_i \int_{\Gamma_{fs}} N_i \cdot N_j d\Gamma = \boldsymbol{M} \cdot \boldsymbol{\eta}, \\
(\xi, \phi_t)_{\Gamma_{fs}} &\approx \sum_{i=0}^{n} \dot{\phi}_i \int_{\Gamma_{fs}} N_i \cdot N_j d\Gamma = \boldsymbol{M} \cdot \dot{\boldsymbol{\phi}}, \\
(w, \phi_t)_{\Gamma_{fs}} &\approx \sum_{i=0}^{n} \dot{\phi}_i \int_{\Gamma_{fs}} N_i \cdot N_j d\Gamma = \boldsymbol{M} \cdot \dot{\boldsymbol{\phi}}, \\
(\xi, \eta_t)_{\Gamma_{fs}} &\approx \sum_{i=0}^{n} \dot{\eta}_i \int_{\Gamma_{fs}} N_i \cdot N_j d\Gamma = \boldsymbol{M} \cdot \dot{\boldsymbol{\eta}}, \\
(\xi, \eta)_{\Gamma_{fs}} &\approx \sum_{i=0}^{n} \eta_i \int_{\Gamma_{fs}} N_i \cdot N_j d\Gamma = \boldsymbol{M} \cdot \boldsymbol{\eta}, \\
(w, \eta)_{\Gamma_{fs}} &\approx \sum_{i=0}^{n} \eta_i \int_{\Gamma_{fs}} N_i \cdot N_j d\Gamma = \boldsymbol{M} \cdot \boldsymbol{\eta}.
\end{aligned}
$$

The weak form must hold for all test functions w and ξ. Therefore, they are alternately set to 0 to be able to write them in matrix-vector form. In other words: w is set according to (6.2a) and ξ to 0, after which w is set to 0 and ξ to (6.2b). By doing this, (5.7) can be written in matrix-vector form as:

$$
\boldsymbol{A}\dot{\boldsymbol{x}} = -\boldsymbol{B}\boldsymbol{x}, \tag{6.4}
$$

where matrices

$$
\boldsymbol{A} = \begin{bmatrix} \frac{1}{2}\frac{\alpha}{g}\boldsymbol{M} & -\boldsymbol{M} \\ \frac{1}{2}\boldsymbol{M} & 0 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{K_w} & \frac{1}{2}\alpha\boldsymbol{M} \\ 0 & \frac{1}{2}g\boldsymbol{M} \end{bmatrix}.
$$

And vectors

$$
\dot{\boldsymbol{x}} = \left[\dot{\phi}_0, \ldots, \dot{\phi}_n, \dot{\eta}_0, \ldots, \dot{\eta}_n\right]^T \qquad \text{and} \qquad \boldsymbol{x} = \left[\phi_0, \ldots, \phi_n, \eta_0, \ldots, \eta_n\right]^T.
$$

6.4 SPATIAL DISCRETISATION OF THE DECOUPLED FORM

Again using the same methodology, the decoupled form

$$
(\nabla w, \nabla \phi) = 0 \tag{5.9a, repeated}
$$

$$
(w, \phi_t + g\eta)_{\Gamma_{fs}} + \frac{g^2}{\alpha^2}(\xi, \eta_t)_{\Gamma_{fs}} = \frac{g^2}{\alpha^2}(\xi, \phi_z)_{\Gamma_{fs}} \tag{5.9b, repeated}
$$

can be discretised. This is done by using the discretisations used for the mixed form with the addition of the approximation of (ξ, ϕ_z) by

$$
(\xi, \phi_z) \approx \sum_{i=0}^{n} \phi_i \int_{\Gamma_{fs}} N_i \cdot \frac{\partial}{\partial z} N_j d\Gamma = \boldsymbol{F} \cdot \boldsymbol{\phi}.
$$

Furthermore, stiffness matrix $\boldsymbol{K}_{nfs}$ is formed by setting the rows in stiffness matrix $\boldsymbol{K}$ that are associated with values of ϕ at the free-surface to 0. This again leads to a system that can be written as:

$$\boldsymbol{A}\dot{\boldsymbol{x}} = -\boldsymbol{B}\boldsymbol{x}, \tag{6.6}$$

where this time

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{M} & 0 \\ 0 & \frac{g^2}{\alpha^2}\boldsymbol{M} \end{bmatrix} \quad \text{and} \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{K}_{nfs} & g\boldsymbol{M} \\ \frac{g^2}{\alpha^2}\boldsymbol{F} & 0 \end{bmatrix}.$$

6.5 TEMPORAL DISCRETISATION

The first, reduced weak formulation is a second order PDE, whereas the second (mixed) and third (decoupled) formulations are first order. They therefore need slightly different temporal discretisations. The first order mixed and decoupled formulations are discretised in time first, followed by the reduced formulation.

MIXED AND DECOUPLED FORMULATION In order to calculate the time derivative of the mixed and decoupled form implicitly, the value of $\boldsymbol{x}$ at time step $n + a$ is approximated by

$$\boldsymbol{x}^{n+a} = \boldsymbol{x}^n + a\Delta t\dot{\boldsymbol{x}}^{n+a} \tag{6.7}$$

where Δt is the time step between time step n and $n+1$ and $0 \leq a \leq 1$. The value of η^{n+a} is approximated equivalently. Substituting this in (6.4) or (6.6) results in

$$\boldsymbol{A}\dot{\boldsymbol{x}}^{n+a} + \boldsymbol{B}\left(\boldsymbol{x}^n + \Delta t\dot{\boldsymbol{x}}^{n+a}\right) = 0.$$

which can be written as

$$(\boldsymbol{A} + \Delta t\boldsymbol{B}) \cdot \dot{\boldsymbol{x}}^{n+a} = -\boldsymbol{B} \cdot \boldsymbol{x}^{\boldsymbol{n}}.$$

So the derivative at time step $n + a$ is

$$\dot{\boldsymbol{x}}^{n+a} = (\boldsymbol{A} + \Delta t\boldsymbol{B})^{-1} \cdot \boldsymbol{B} \cdot \boldsymbol{x}^n. \tag{6.8}$$

The value of $\boldsymbol{x}^{n+1}$ can then be found with

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \Delta t\dot{\boldsymbol{x}}^{n+a} \tag{6.9}$$

For the implicit midpoint method, a is set to $\frac{1}{2}$. Setting it to 1 results in the backward Euler method.

REDUCED FORMULATION Since the reduced formulation is a second order PDE, it needs a little different approach to the time discretisation. An implicit method can be found by assuming

$$\boldsymbol{\phi}^{n+a} = \boldsymbol{\phi}^n + a\Delta t\dot{\boldsymbol{\phi}}^{n+a},$$
$$\dot{\boldsymbol{\phi}}^{n+a} = \dot{\boldsymbol{\phi}}^n + a\Delta t\ddot{\boldsymbol{\phi}}^{n+a}.$$

The value of $\ddot{\boldsymbol{\phi}}^{n+a}$ can be found by substituting these into (6.3):

$$\boldsymbol{K}\left(\boldsymbol{\phi}^n + a\Delta t\dot{\boldsymbol{\phi}}^n + (a\Delta t)^2\ddot{\boldsymbol{\phi}}^{n+1}\right) + \frac{1}{g}\boldsymbol{M}\ddot{\boldsymbol{\phi}}^{n+a} = \boldsymbol{0}$$

So the value of $\ddot{\boldsymbol{\phi}}^{n+a}$ can be found with

$$\ddot{\boldsymbol{\phi}}^{n+a} = \left(\frac{1}{g}\boldsymbol{M} + a^2\Delta t^2\boldsymbol{K}\right)^{-1}\left(-\boldsymbol{K}\left(\boldsymbol{\phi}^n + a\dot{\boldsymbol{\phi}}^n\right)\right). \tag{6.10}$$

In reality, the inverse is not really determined. Instead, the system is solved linearly. The resulting $\ddot{\boldsymbol{\phi}}^{n+a}$ is used to calculate $\boldsymbol{\phi}^{n+1}$ and $\dot{\boldsymbol{\phi}}^{n+1}$ with

$$\boldsymbol{\phi}^{n+1} = \boldsymbol{\phi}^n + \Delta t\dot{\boldsymbol{\phi}}^{n+a}, \tag{6.11a}$$
$$\dot{\boldsymbol{\phi}}^{n+1} = \dot{\boldsymbol{\phi}}^n + \Delta t\ddot{\boldsymbol{\phi}}^{n+a}. \tag{6.11b}$$

6.6 POST PROCESSING

Energy conservation is an important measure of the behaviour of the formulations. It will therefore be used to assess the results of the three weak forms. In order to do this, a numerical approximation of the energy is needed. This numerical approximation is explained in this paragraph.

The energy can subdivided in potential and kinetic energy (E_{pot} and E_{kin}). The total energy E_{tot} is defined as the sum of these two. The numerical approximation of the kinetic energy is explained first, followed by the potential energy.

KINETIC ENERGY The density of the water is assumed to be constant and a measure for the kinetic energy therefore is

$$E_{kin} = \frac{1}{2}\int_\Omega (\nabla\phi)^2\, d\Omega, \tag{6.12}$$

To determine the kinetic energy numerically, potential ϕ is again approximated by

$$\phi^h(\boldsymbol{x}, t) \equiv \sum_n \phi_i(t)N_i(\boldsymbol{x}).$$

The kinetic energy can then be approximated with

$$\begin{aligned} E^h_{kin} &= \frac{1}{2}\int_\Omega \left(\nabla \phi^h\right)^2 d\Omega \\ &= \frac{1}{2}\int_\Omega \left(\nabla \sum_{i=0}^{n} \phi_i N_i\right)^2 d\Omega \\ &= \frac{1}{2}\int_\Omega \left(\nabla \sum_{i=0}^{n} \phi_i N_i\right) \cdot \left(\nabla \sum_{j=0}^{n} \phi_j N_j\right) d\Omega \\ &= \frac{1}{2}\sum_{i=0}^{n}\phi_i \sum_{j=0}^{n}\phi_j \int_\Omega (\nabla N_i)\cdot(\nabla N_j)\, d\Omega, \end{aligned}$$

which can be written in matrix-vector notation as

$$E^h_{kin} = \frac{1}{2}\boldsymbol{\phi}^T \boldsymbol{K}\boldsymbol{\phi}, \tag{6.13}$$

where stiffness matrix $\boldsymbol{K}$ and vector $\boldsymbol{\phi}$ are equal to the ones in paragraph 6.2, 6.3 and 6.4 without the free surface rows set to 0.

POTENTIAL ENERGY The potential energy E_{pot} is a relative quantity. In this thesis, it is considered relative to the still fluid. So, $E_{pot} = 0$ if the wave height $\eta = 0$ over the whole free surface. A measure can then be found with

$$E_{pot} = \int_{\Gamma_{fs}} \frac{1}{2} g \eta^2 d\Gamma \tag{6.14}$$

which can be written as

$$E_{pot} = \int_{\Gamma_{fs}} \frac{1}{2g} \phi_t^2 d\Gamma \tag{6.15}$$

with boundary condition (5.2b).

For the reduced formulation, the potential energy can be determined numerically by discretising ϕ_t in (6.15) by assuming it can be approximated as

$$\phi_t^h(\boldsymbol{x}, t) \equiv \sum_{i=0}^{n} \dot{\phi}_i(t) N_i(\boldsymbol{x}).$$

The potential energy can then be approximated by

$$\begin{aligned} E^h_{pot} &= \int_{\Gamma_{fs}} \frac{1}{2g}\phi_t^2 d\Gamma \\ &= \frac{1}{2g}\int_{\Gamma_{fs}} \left(\sum_{i=0}^{n} \dot{\phi}_i N_i\right)^2 d\Gamma, \\ &= \frac{1}{2g}\int_{\Gamma_{fs}} \left(\sum_{i=0}^{n} \dot{\phi}_i N_i\right) \cdot \left(\sum_{j=0}^{n} \dot{\phi}_j N_j\right) d\Gamma, \\ &= \frac{1}{2g}\sum_{i=0}^{n} \dot{\phi}_i \sum_{j=0}^{n} \dot{\phi}_j \int_{\Gamma_{fs}} N_i \cdot N_j d\Gamma, \end{aligned}$$

which can be written in matrix-vector notation as

$$E^h_{pot} = \frac{1}{2g}\dot{\boldsymbol{\phi}}^T \boldsymbol{M} \dot{\boldsymbol{\phi}}, \tag{6.16}$$

where mass matrix $\boldsymbol{M}$ is equal to the one in paragraph 6.2 and

$$\dot{\boldsymbol{\phi}} = \left[\dot{\phi}_0, \ldots, \dot{\phi}_n\right]^T.$$

For the mixed and decoupled formulation, wave height η is available, so (6.14) is used. It is discretised by approximating η as

$$\eta^h \equiv \sum_{i=0}^{n} \eta_i(t) N_i(\boldsymbol{x}).$$

Substituting this in (6.14) results in

$$\begin{aligned} E^h_{pot} &= \frac{1}{2}g\int_{\Gamma_{fs}} \left(\sum_{i=0}^{n} \eta_i N_i\right)^2 dx, \\ &= \frac{1}{2}g\int_{\Gamma_{fs}} \left(\sum_{j=0}^{n} \eta_j N_j\right) \cdot \left(\sum_{i=0}^{n} \eta_i N_i\right) dx, \\ &= \frac{1}{2}g\sum_{i=0}^{n} \eta_i \sum_{j=0}^{n} \eta_j \int_{\Gamma_{fs}} N_j \cdot N_i dx, \end{aligned}$$

which can be written in matrix-vector form as

$$E_{pot} = \frac{1}{2}g\boldsymbol{\eta}^T \boldsymbol{M}_w \boldsymbol{\eta}. \tag{6.17}$$

7

MFEM CODE OVERVIEW

The systems derived in chapter 6 will be solved with open-source, finite element C++ library MFEM [55]. This library was developed at Lawrence and Livermore National Laboratory (LLNL) and a list of contributors is available at `mfem.org/about/`. It offers support for a variety of 2D and 3D finite element spaces, most importantly for this thesis: NURBS spaces for isogeometric analysis. Furthermore, multiple time solvers are readily available.

Some adaptions are made to the block matrices that were derived in chapter 5. These adaptions are explained in the first paragraph. Furthermore, Some practical problems were encountered. These are explained in the second paragraph where the reasons and consequences are explained. For a more in depth discussion of the MFEM codes, the reader is directed towards appendix C.

7.1 MFEM TIME SOLVER

In the MFEM package, many ODE solvers are available. These solvers need a function in which the time derivative is calculated. In this thesis, only implicit solvers are used. Implicit formulations of the three weak forms are formed in paragraph 6.5: (6.10) for the reduced formulation and (6.8) for the mixed and decoupled formulations. The matrices that are needed for the reduced formulation are used without any changes. This is not the case for the matrices for the other two formulations. For these two formulations, a dummy Laplace problem will be solved such that η is not only defined on the free surface, but also on the interior. This is done to prevent the matrices from being singular and can be used in a later stage to solve non-linear waves. The dummy Laplace problem is defined as:

Find η such that

$$\Delta\tilde{\eta} = 0 \qquad \text{in } \Omega \tag{7.1a}$$

$$\tilde{\eta} = \eta \qquad \text{on } \Gamma_{fs} \tag{7.1b}$$

$$\boldsymbol{n} \cdot \nabla\tilde{\eta} = 0 \qquad \text{on } \Gamma_{nfs} \tag{7.1c}$$

where η follows from the the main problem. This results in

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{K_{AC}} & \frac{1}{2}\alpha\boldsymbol{M_{BC}} \\ 0 & \frac{1}{2}g\boldsymbol{M_{BD}} + \boldsymbol{K_{BD}} \end{bmatrix} \tag{7.2}$$

for the mixed formulation and in

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{K_{AC}} & g\boldsymbol{M_{BC}} \\ \frac{g^2}{\alpha^2}\boldsymbol{F_{AD}} & \boldsymbol{K_{BD}} \end{bmatrix} \tag{7.3}$$

for the decoupled formulation.

A second order implicit midpoint solver is not available in MFEM, the reduced formulation will therefore be solved with a generalized-α solver [22] with ρ_∞ set to 1. By doing this, the generalized-α solver is effectively equivalent to the implicit midpoint solver. Formulations two and three will be solved with the implicit midpoint solver.

7.2 PRACTICAL LIMITATIONS OF THE MFEM PACKAGE

Most techniques and functions are implemented for both 'standard' finite element meshes and NURBS meshes. However, some things are not implemented for NURBS meshes yet. This is not because it is impossible to implement these, but it is simply not done yet. These limitations did influence the research.

1. Periodic NURBS meshes are not available.
2. There is no method to calculate the integral $\int_{\Gamma_{fs}} N_i \cdot \frac{\partial}{\partial z} N_j d\Gamma$ for NURBS meshes. This integral used to form F matrix in the decoupled formulation.

The consequence of this second limitation that it is unfortunately not possible to test the third, decoupled formulation with an NURBS mesh.

Another difficulty is visualisation. At this moment, the lightweight visualisation program GLVis [30], also developed at LLNL, can be used to visualize the meshes and the results. It supports NURBS. However, it does not offer the same possibilities as the heavier visualisation program Visit [20]. Visit does not fully support NURBS yet. Supervisor Akkerman is working on an adaption of Visit that does support NURBS, but this is not openly available. Where possible, GLVis is therefore used, but some figures could only be made in Visit.

Part III

RESULTS

The three weak forms that were formulated in part II are tested on two two-dimensional problems and one three-dimensional problem. These problems are defined by their initial conditions and the used mesh. The two-dimensional problems are a *sloshing wave* (chapter 8) and an *airy wave* (chapter 9). These two problems have analytical solutions that can be used to verify and validate the results. Since these two problems are relatively simple, they are not used to test the differences between 'standard' FEA and IGA. This test is performed in chapters **??** and 11. In chapter **??**, a step wave is used In chapter 11, a three-dimensional showcase is given in which the mixed weak form is used to see the differences between IGA and standard FEA.

8

TEST CASE I: TWO-DIMENSIONAL SLOSHING WAVE

The first verification of the code is done by simulating a sloshing-type wave with standard finite element analysis. This is a simple problem for which an analytical solution is available. That solution can be used to verify and validate the weak forms and to verify the developed C++ codes. The wave is simulated by setting the right initial conditions. These are described in the first paragraph. The analytical solution is derived in the second paragraph. In the third paragraph the results for the three different formulations are given and compared to the analytical solution.

8.1 INITIAL CONDITIONS

In the sloshing wave problem, one wave in a bounded square tank of 1 m² is simulated. This tank is discretised with a quadrilateral mesh. The problem is initialized by setting the potential to

$$\phi(x, z, t = 0) = a \cos(kx) \cosh(kz), \tag{8.1}$$

where a is the wave amplitude and k is the wave number defined as

$$k \equiv \frac{2\pi}{\lambda}, \tag{8.2}$$

where λ is the wavelength. The initial wave height is set to $\eta = 0$. The initial condition is illustrated in figure 8.1, where the domain is deformed with the value of potential ϕ (note: not with wave height η!).

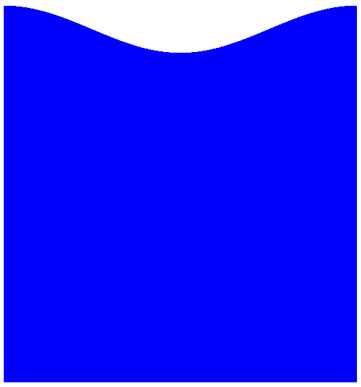


Figure 8.1: Illustration of the initial value of ϕ for the sloshing wave

This initial condition satisfies the Laplace equation (5.2a) and the boundary conditions (5.2b) to (5.2d). Prove of this is given in appendix

VARIABLE	ANALYTICAL VALUE		
ω	$\sqrt{kg\tanh(kz)}$	7.851	rad/s
T_p^a	$2\pi/\omega$	0.8003	s
Amp of E_{kin}^a	$\frac{1}{8}a^2k\sinh(2k)$	0.1126	m^4/s^2
E_{tot}^a	$\frac{1}{8}a^2k\sinh(2k)$	0.1126	m^4/s^2

Table 8.1: Analytical solution for the sloshing wave

A.1. For the initialization, wavelength $\lambda = 1$ m and wave height $a = 0.001$ m are used. This results in a wave number $k = 2\pi$.

8.2 ANALYTICAL SOLUTION

An analytical solution can be found with the help of separation of variables. This is done in appendix A.2. This results in

$$\phi(x,z,t) = a\cos(kx)\cosh(kz)\cos(\omega t) \tag{8.3}$$

where ω is given by the dispersion relation

$$\omega^2 = kg\tanh(kH). \tag{8.4}$$

The mass density is assumed to be constant, so a measure of the kinetic energy at $t = 0$ when $\eta = 0$ can be calculated with (6.12):

$$E_{kin} = \int_\Omega (\nabla\phi)^2\, d\Omega = \int_0^1\int_0^H (\nabla\phi)^2 dzdx. \tag{8.5}$$

which results in

$$E_{kin} = \frac{1}{8}a^2k\sinh(2k) \tag{8.6}$$

The full derivation is also shown in appendix A.2.

This is the maximum value of the kinetic, because the velocity of the wave is at a maximum if $\eta = 0$. Kinetic energy is transformed into potential energy as the wave height η increases. The total energy E_{tot} is defined as the sum of potential and kinetic energy. Since no energy is dissipated, it is constant:

$$E_{tot} = E_{kin} + E_{pot} = constant$$

$E_{kin} = 0$ if the wave is at a peak, so the amplitude of the potential energy is equal to that of the kinetic energy and is $\frac{1}{4}T_p$ behind.

For $a = 0.001$ and $\lambda = 1$ the kinetic energy is calculated with (8.6) and is approximately 0.1126. The derived analytical solutions are summarised in table 8.1.

8.3 RESULTS FOR THE THREE FORMULATIONS

The three different weak formulations that are derived in chapter 5 and discretised in chapter 6 are used to solve the sloshing wave problem. This is done for increasing degrees of freedom, keeping the Courant-Friedrichs-Lewy (CFL) number constant. The CFL number for problems in n dimensions is defined as

$$C = \Delta t \sum_{i=1}^{n} \frac{u_{x_i}}{\Delta x_i}. \tag{8.7}$$

The CFL number is used in the CFL condition, which states that $C \leq C_{max}$ is a necessary condition for convergence. Since the velocity u is constant for constant wave length and depth, the relation between time step Δt and grid size Δx_i is linear.

The formulations are solved with the first and second order versions of the implicit midpoint solvers described in 6.5. The calculations are done for $0 \leq t \leq 3.3T_p^a$. These test parameters are summarised in table 8.2. The number of mesh nodes n in horizontal direction is increased from 5 to 65 in four refinement steps. This results in the test matrix in table 8.3.

PARAMETER	VALUE
Mesh type	Quadrilateral
Solver	Implicit midpoint
Order	2
Final time	$3.3T_p^a$

Table 8.2: Parameters for the sloshing wave problem

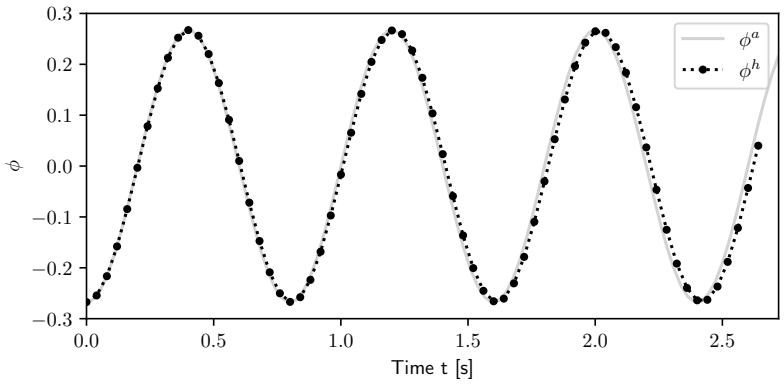
	TEST 1	TEST 2	TEST 3	TEST 4	TEST 5
$\Delta t / T_p^a$	0.1	0.05	0.025	0.0125	0.00625
n	5	9	17	33	65

Table 8.3: Parameters for the sloshing wave problem

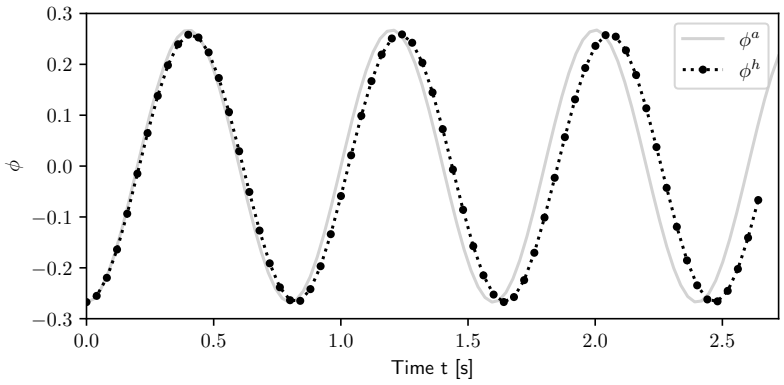
The results will be compared on basis of:

1. Period T_p,
2. Amplitude of the kinetic energy E_{kin},
3. Value of the total energy E_{tot}.

FIRST VERIFICATION A first verification is done by looking at the time traces of the numerically determined potential ϕ^h and the analytically determine ϕ^a at the free surface at $x = 0.5$. As an example, the time trace for $n = 9$ is shown in figures 8.2a (mixed formulation) and 8.2b (decoupled formulation). The results for the the reduced and mixed formulations are very similar, so only the time trace for the second one is shown. Looking at figure 8.2a, it is clear that the reduced and mixed formulations result in a wave that resembles the analytical solution very closely. The time trace in figure 8.2b is very similar as well, but the period is clearly overestimated.



(a) Mixed formulation



(b) Decoupled formulation

Figure 8.2: Time traces of potential ϕ at $(x, z) = (0.5, 1)$ - Implicit midpoint solver with $n = 9$ and $\Delta t = 0.05 T_p^a$

A second verification is done by looking at the potential, kinetic and total energy. These are calculated as described in paragraph 6.6. Typical figures for the three formulations for $n = 9$ are shown in figures 8.3 and 8.4. The results for the reduced and mixed formulations are again almost identical, which is why only the results for the

mixed formulation are shown. Several things can be noted from the figures. Firstly, the total energy is underestimated with all three the formulations. More importantly however: the decoupled formulation shows periodic behaviour of the total energy instead of staying constant. This seems to come from the not purely sinusoidal behaviour of the kinetic energy. A possible reason is the presence of ϕ_z in the decoupled formulation. Since the mesh is quite coarse with only 9 degrees of freedom in both directions, the differential in z-direction is not precise enough. The amplitude of the periodic difference reduces to insignificant values for refinement meshes.

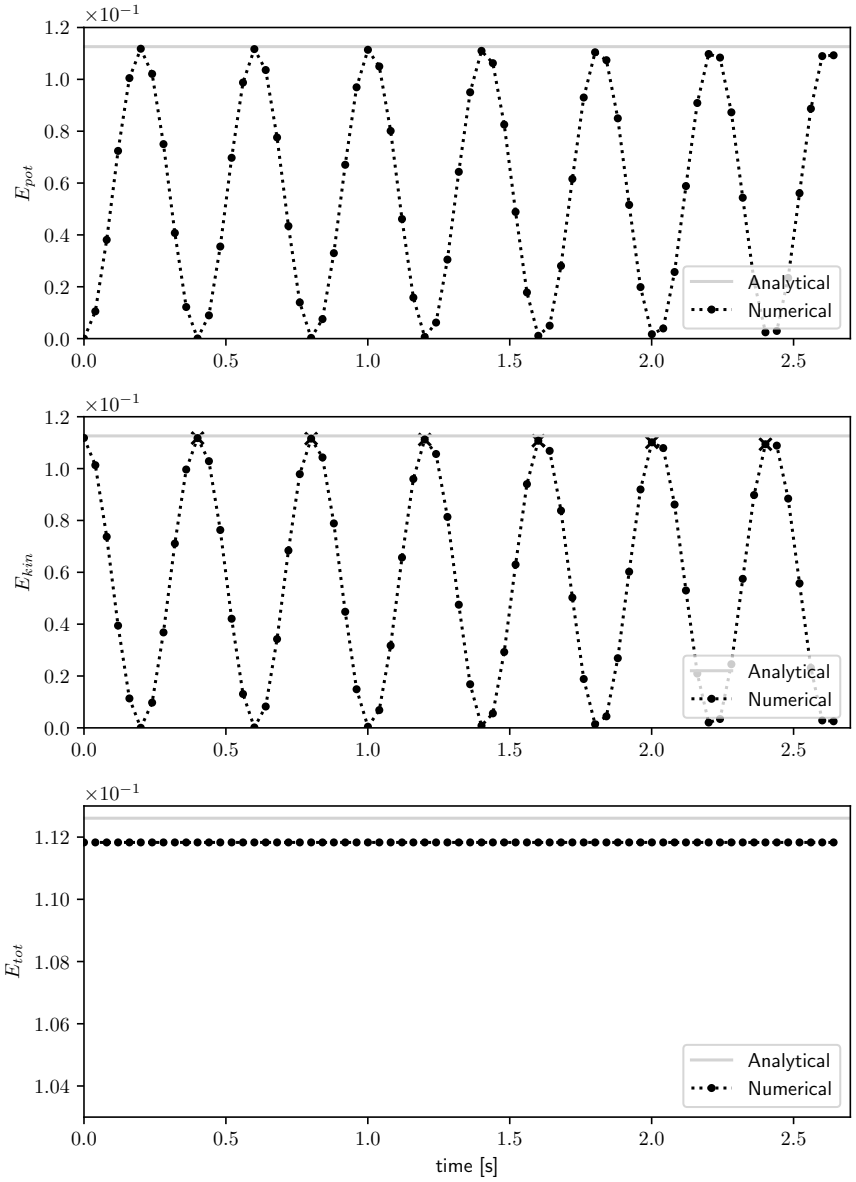


Figure 8.3: Time trace of E_{pot}, E_{kin} and E_{tot} - mixed form with $n = 9$ and $\Delta t = 0.05 T_p^a$

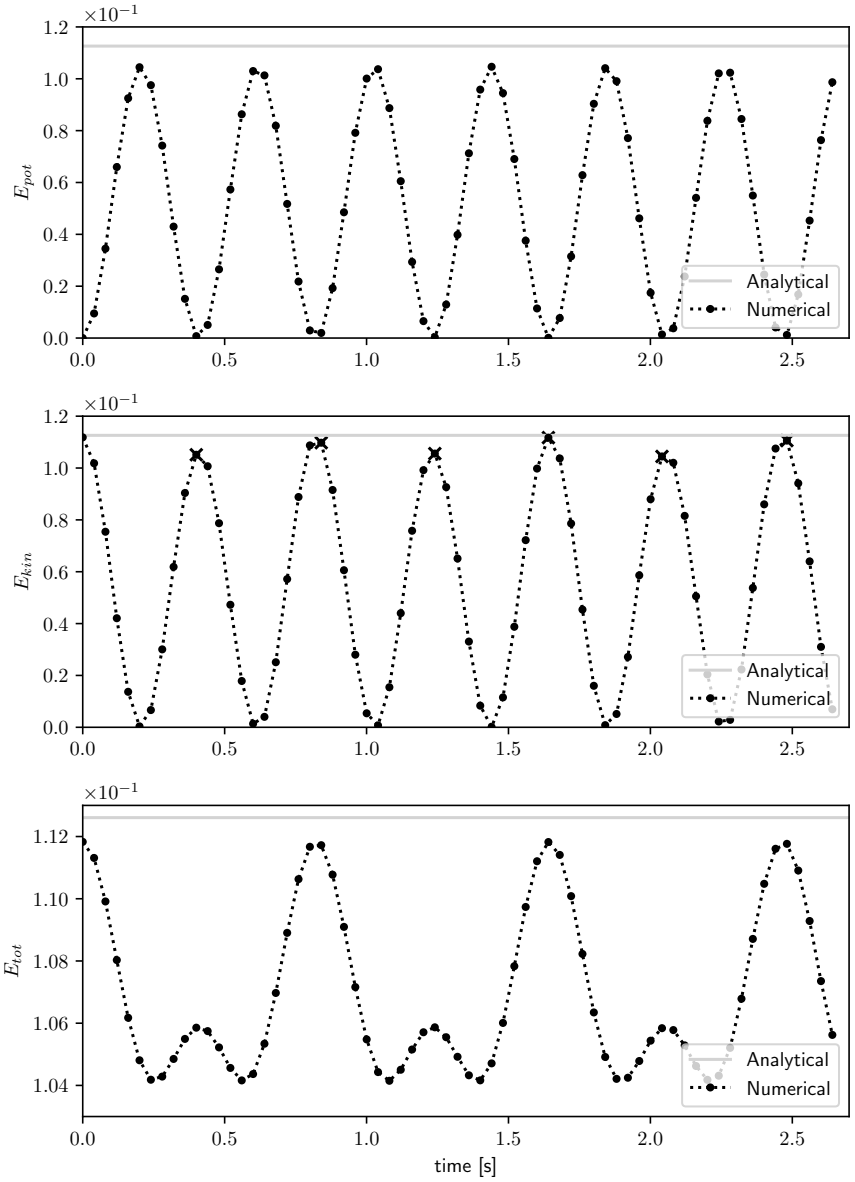


Figure 8.4: Time trace of E_{pot}, E_{kin} and E_{tot} - decoupled form with $n = 9$ and $\Delta t = 0.05 T_p^a$

FURTHER VERIFICATION A more in-depth verification is done by comparing the numerically determined period and $E^h_{tot}(t = t_3)$ to the analytically determined ones. The numerically determined period T^h_p is calculated with

$$T^h_p = \frac{t_3 - t_0}{3}, \tag{8.8}$$

where t_3 is the time of the third peak of the time trace of $\phi(0.5, 1.0)$ and t_0 the time of the first peak.

The values of T^h_p are shown in table 8.4 and figure 8.5. The reduced and mixed formulations give equivalent results and give a very accurate period for grid sizes of $n = 9$ and higher. The total energy E_{tot} is constant for the reduced and mixed formulation for all n. The total energy E_{tot} at $t = t_3$ is shown in figure 8.6. The percentage difference with the analytically detemined value is shown in table 8.5. The numerically determined energy is lower than the analytical value. This underestimation reduces for more refined meshes and is smaller than 0.2% for $n \geq 17$

The decoupled formulation needs a much more refined mesh and smaller time step to calculate the period with similar accuracy. The period is overestimated for smaller time steps. It is calculated accurately for a time step $\Delta t \leq 0.0125 T^a_p$. The total energy E_{tot} is underestimated more than with the reduced and mixed formulations. The difference is clearly visible on coarse meshes.

	$n = 5$	$n = 9$	$n = 17$	$n = 33$	$n = 65$
Analytical	0.8003				
Red. form	0.8403	0.8003	0.8003	0.8003	0.8003
Mixed form	0.8403	0.8003	0.8003	0.8003	0.8003
Dec. form	0.8803	0.8203	0.8103	0.8003	0.8003

Table 8.4: T_p of the sloshing wave

	$n = 5$	$n = 9$	$n = 17$	$n = 33$	$n = 65$
Analytical	0.112607				
Red. form	-3.41%	-0.691%	-0.110%	-0.016%	-0.002%
Mixed form	-3.41%	-0.691%	-0.110%	-0.016%	-0.002%
Dec. form	-16.1%	-0.748%	-0.111%	-0.017%	-0.002%

Table 8.5: E_{tot} at $t = 3T_p$ of the sloshing wave

CONCLUSIONS It can be concluded that the results for the reduced and mixed formulations are equivalent and are quite satisfactory:

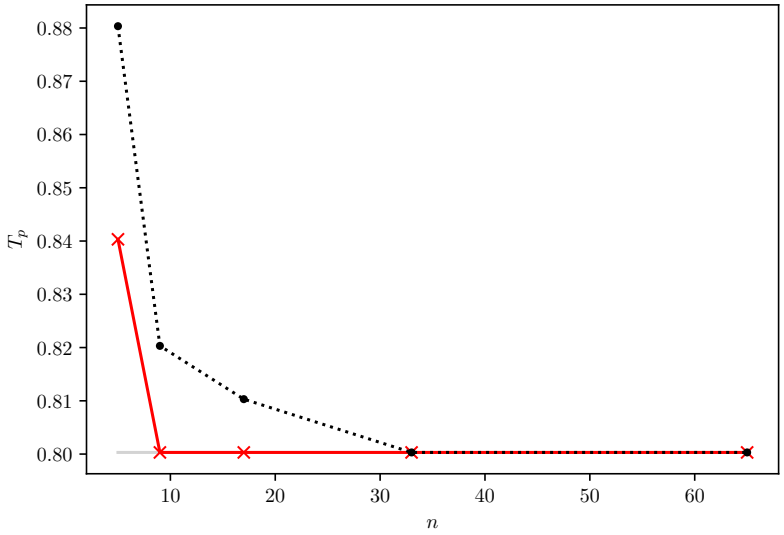


Figure 8.5: Convergence of T_p for the sloshing wave

- Period T_p is calculated accurately for $n \geq 9$ and $\Delta t \leq 0.05 T_p^a$.
- Energy is conserved exactly.
- Total energy E_{tot} is underestimated, but the underestimation is lower than 0.2% if $n \geq 9$.

Results for the decoupled formulation are less accurate:

- Period T_p is calculated accurately for $n \geq 33$ and $T_p \leq 0.0125 T_p^a$.
- Energy is conserved, but shows periodic behaviour. The amplitude is smaller for refined meshes.
- Total energy E_{tot} is underestimated more than for the other two formulations, especially on coarser meshes.

The most probable reason for the underestimation of the total energy E_{tot} is the fact that it is calculated on a discrete mesh. The most probable reason for the less accurate results of the decoupled formulation is the presence of the $\frac{\partial}{\partial z}$ term in the formulation. This derivative cannot be calculated accurate enough on coarse meshes. The results might be improved by using a mesh with smaller elements near the free surface. The derivative can then be calculated more accurately.

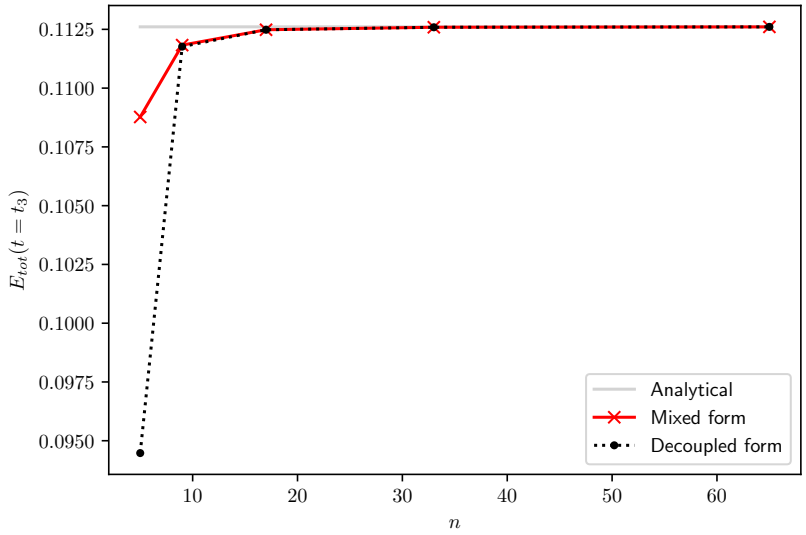


Figure 8.6: Convergence of E_{tot} at $t = t_3$

9 TEST CASE II: TWO-DIMENSIONAL AIRY WAVE

A slightly more complicated problem than the sloshing wave of chapter 8 is the simulation of an *airy wave*. In this problem, a wave does not only change in vertical direction, but also moves in horizontal direction. This problem is therefore used to test the three weak forms more. Again, an analytical solution is available. This is used to calculate the period and the kinetic, potential and total energy analytically. The results of the three weak forms are compared with these analytical ones.

The airy wave problem is initialised by setting the conditions that are described in the first paragraph. A two-dimensional, quadrilateral, periodic mesh is used. This periodic mesh is created by connecting the points on the left edge to those on the right edge so they behave exactly the same. By doing this, everything that goes out of the right edge comes into the left edge and vice versa, effectively creating an infinite mesh. The analytical solution of this problem is given in the second paragraph. In the third paragraph, the results are shown and compared with the analytical solutions. In the last paragraph, the mixed weak formulation is used to calculate the period for different wave lengths. These results are compared to the dispersion relation.

9.1 INITIAL CONDITIONS

A airy wave of length $\lambda = 1$ meter is simulated on a square 1 m² mesh that is periodic in the horizontal (or x-) direction. The standard formulation of an airy wave in two dimensions is

$$\phi(x,z,t) = a\frac{\omega}{k}\frac{\cosh(k(h+z)}{\sinh(kH)}\sin(kx-\omega t), \tag{9.1}$$

where a is the wave amplitude, ω is the wave frequency that can be calculated with the dispersion relation (8.4) and k the wave number that is defined in (8.2). In this equation, $z = 0$ at the still free surface instead of at the bottom. It can be shown that formulation (9.1) satisfies the strong problem definition (5.2). This is done in appendix B.1.

Formulation (9.1) is used to define the initial values by setting $t = 0$:

$$\phi(x,z,t) = a\frac{\omega}{k}\frac{\cosh(k(h+z))}{\sinh(kh}\sin(kx),$$

Wave height η is set to

$$\eta = a\cos(kx), \tag{9.2}$$

which follows from boundary boundary condition (5.2b):

$$\eta = -\frac{1}{g}\phi_t = a\frac{\omega^2}{kg}\frac{\cosh(kh)}{\sinh(kh)}\cos(kx)$$

and noting that $\omega^2/(kg) = 1$ and $\cosh(2\pi)/\sinh(2\pi) \approx 1$, because $k = 2\pi$ and $h = 1$.

This initialisation is illustrated in figure 9.1, where the domain and the elevation η along the free surface are plotted.

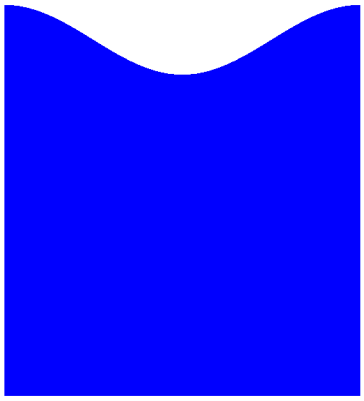


Figure 9.1: Illustration of the initial value of η for the airy wave

9.2 ANALYTICAL SOLUTION

The mass density is again assumed to be constant, so an analytical measure of the kinetic energy can be calculated by splitting (6.12) into an interior and a free surface part:

$$\begin{aligned} E_{kin} &= \int_0^1 \int_{-h}^{\eta} (\nabla\phi)^2 dz dx \\ &= \int_0^1 \left(\int_{-h}^{0} (\nabla\phi)^2 dz + \int_0^{\eta} (\nabla\phi)^2 dz \right) dx \end{aligned} \tag{9.3}$$

An analytical measure of the potential energy can be calculated with (6.14):

$$E_{pot} = \int_0^1 \frac{1}{2} g\eta(x)^2 dx. \tag{6.14, repeated}$$

These equations are linearised by assuming η is small. The kinetic energy can then be approximated with

$$E_{kin} \approx \int_0^1 \int_{-h}^{0} (\nabla\phi)^2 dz dx.$$

This results in

$$E_{kin} = \frac{1}{4} a^2 g.$$

and

$$E_{pot} = \frac{1}{4}a^2 g.$$

The full derivation is given in appendix B.2. The resulting analytical solutions are summarised in table 9.1.

VARIABLE	ANALYTICAL VALUE		
ω	$\sqrt{kg\tanh(kz)}$	7.851	rad/s
T_p^a	$2\pi/\omega$	0.8003	s
E_{kin}^a	$\frac{1}{4}a^2 g$	$2.453 \cdot 10^{-}4$	m^4/s^2
E_{tot}^a	$\frac{1}{4}a^2 g$	$4.905 \cdot 10^{-}4$	m^4/s^2

Table 9.1: Analytical solution for the airy wave

9.3 RESULTS FOR THE THREE WEAK FORMULATIONS

The three weak forms are used to solve the airy wave problem on a square mesh of quadrilateral elements that is periodic in horizontal direction. The periodicity results results in one element less in this direction then in vertical (or z-) direction. Experiments are done using second order basis functions with the first and second order versions of the implicit midpoint solver described in paragraph 6.5. The parameters used for the airy wave experiments are summarised in table 9.2. Experiments are run from $t = 0$ to $t = 3.3T_p^a$. The mesh is refined four times while keeping the CFL number (defined in 8.7) constant. This results in the test-matrix in table 9.3, where n is the number of mesh nodes in horizontal (or x-) direction.

PARAMETER	VALUE
Mesh type	Periodic, quadrilateral
Solver	Implicit midpoint
Order	2
Final time	$3.3T_p^a$

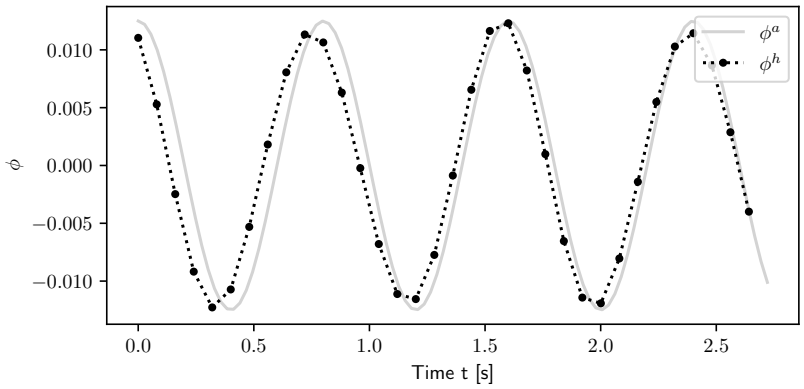
Table 9.2: Parameters for the airy wave problem

FIRST VERIFICATION Just like the previous problem, a first verification is done by looking at the time traces of the numerically determined potential ϕ^h at $x = 0.5$ on the free surface. Examples of this time trace are given in figures 9.2a (mixed formulation) and 9.2b (decoupled formulation). Again, the results for the reduced and mixed formulations are identical, so only the mixed formulation is shown.

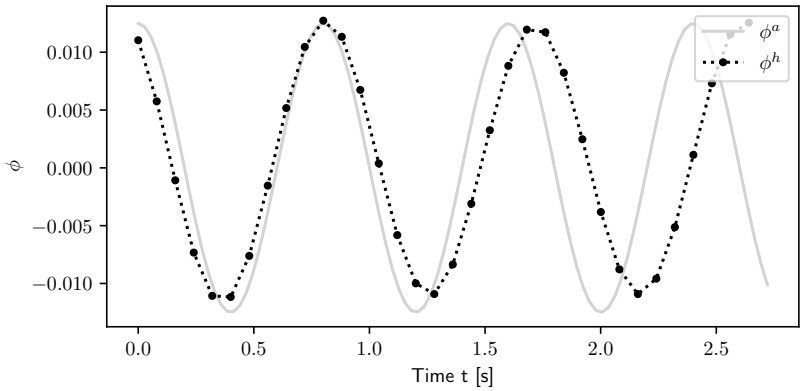
	TEST 6	TEST 7	TEST 8	TEST 9	TEST 10
$\Delta t / T_p^a$	0.1	0.05	0.025	0.0125	0.00625
n	4	7	13	25	49

Table 9.3: Parameters for the airy wave problem

The resulting time traces are similar to those for the sloshing wave problem of chapter 8. The results for the reduced and mixed formulations are just a little bit off, for the decoupled formulation the differences are bigger.



(a) Mixed formulation



(b) Decoupled formulation

Figure 9.2: Time traces of potential ϕ at $(x, z) = (0.5, 1)$ - Implicit midpoint solver with $n = 7$ and $\Delta t = 0.05 T_p^a$

A second verification is done by looking at the potential, kinetic and total energy. Typical time traces are shown in figure 9.3 for the mixed formulation and in figure 9.4 for the decoupled formulation. The energy is again underestimated with both formulations. The reduced

and mixed formulations result in a perfectly constant E_{tot}. The kinetic and potential energy should remain constant, since the shape of the wave does not change. However, they show periodic behaviour. The reason is probably that the wave moves over the mesh and is thereby discretised differently.

The resulting figures for the decoupled formulations are, again, considerably less accurate. A much more refined version is plotted, $n = 25$ instead of $n = 7$, because the limits of the y-axis would otherwise need a much larger reach. The method is energy conservative, but the total energy does not remain constant. Furthermore, the kinetic and potential energies are not constant for the decoupled formulation either. Even worse: the kinetic energy does not even behave purely sinusoidal.

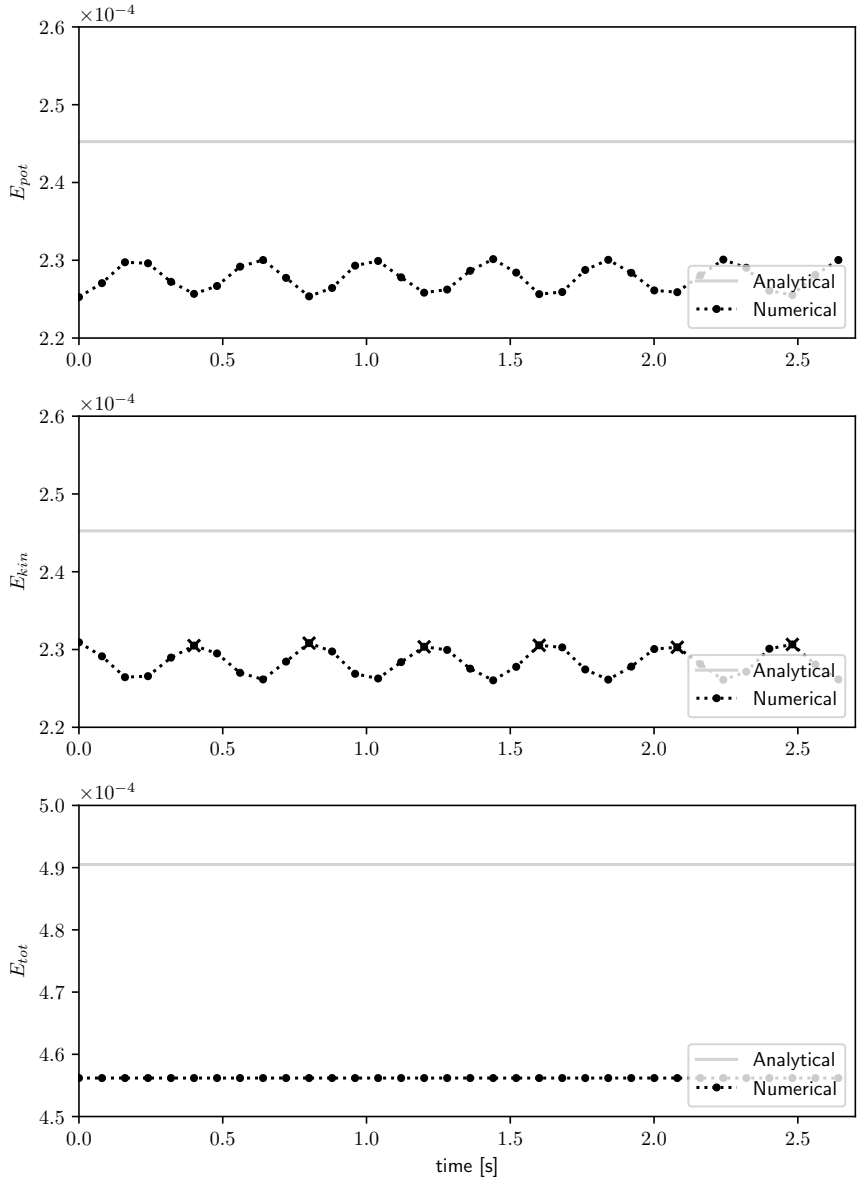


Figure 9.3: Time trace of E_{pot}, E_{kin} and E_{tot} - mixed form with $n = 7$ and $\Delta t = 0.05 T_p^a$

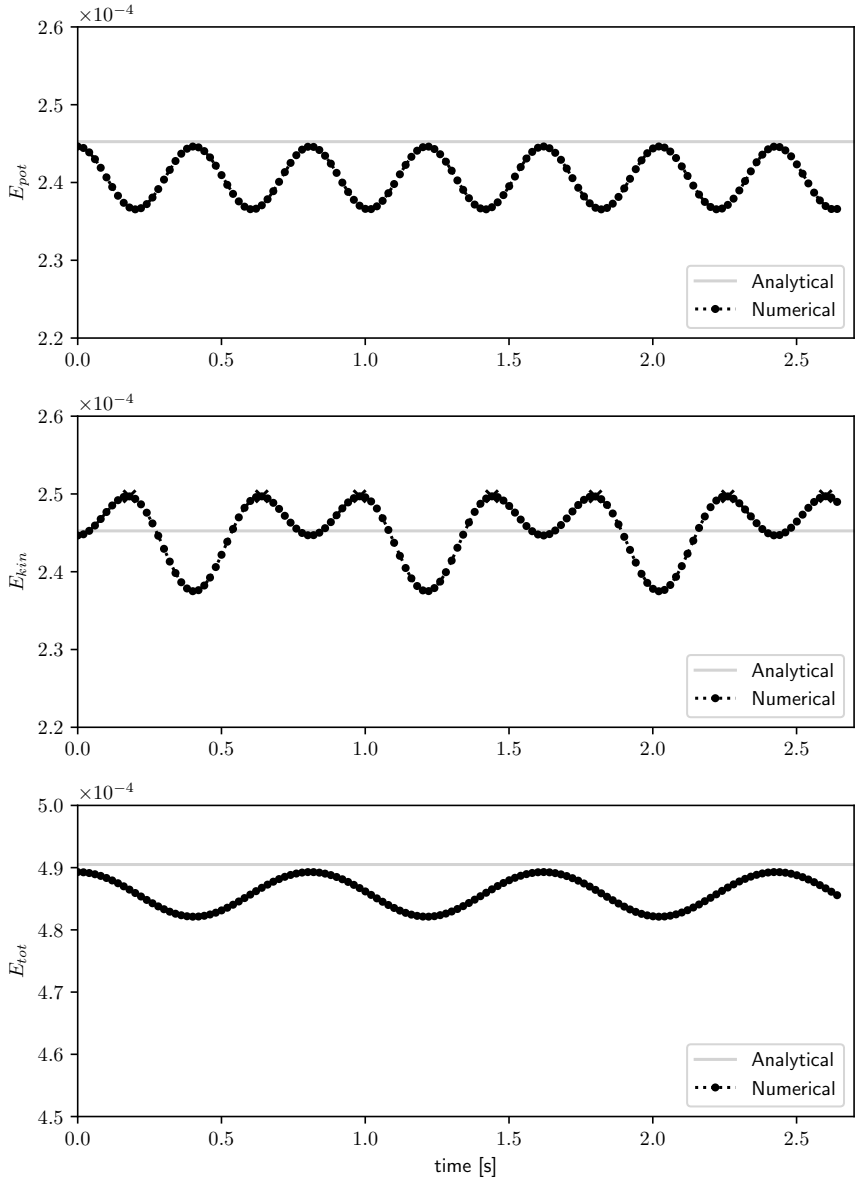


Figure 9.4: Time trace of E_{pot}, E_{kin} and E_{tot} - decoupled form with $n = 25$ and $\Delta t = 0.0125 T_p^a$

FURTHER VERIFICATION The numerically determined period T_p^h, $E_{tot}(t = t_3)$ and ΔE_{kin} are used for further verification. T_p^h is calculated with (8.8). ΔE_{kin} is defined as the difference between the maximum and the minimum value of E_{kin}.

The periods T_p for the five experiments and three formulations are shown in table 9.4 and figure 9.5. The results for the reduced and mixed formulations are identical and are equal to the analytical value for all experiments. The decoupled formulation needs $n \geq 25$ and $\Delta t \leq 0.0125 T_p^a$ for the same accuracy.

The percentage difference between E_{tot} at $t = 3T_p^a$ and the analytically determined E_{tot}^a is shown in table 9.5. All three formulations underestimate the total energy. Results for the reduced and mixed formulations are identical. The difference with the analytical value is smaller than 0.3% if $n \geq 13$ and $\Delta t \leq 0.025 T_p^a$. The decoupled formulation needs one refinement step extra to reach the same accuracy in this respect.

The convergence of the amplitude of the kinetic energy is shown in figure 9.6. It is clear that the amplitude is smaller ánd goes to 0 quicker with the reduced and mixed forms than the decoupled formulation.

	$n = 4$	$n = 7$	$n = 13$	$n = 25$	$n = 49$
Analytical	0.8003				
Red. form	0.8003	0.8003	0.8003	0.8003	0.8003
Mixed form	0.8003	0.8003	0.8003	0.8003	0.8003
Dec. form	0.8403	0.8270	0.8070	0.8003	0.8003

Table 9.4: T_p of the airy wave

	$n = 4$	$n = 7$	$n = 13$	$n = 25$	$n = 49$
Analytical	$4.905 \cdot 10^{-4}$				
Red. form	-7.00%	-1.48%	-0.247%	-0.036%	-0.005%
Mix. form	-7.00%	-1.48%	-0.247%	-0.036%	-0.005%
Dec. form	-8.67%	-2.64%	-0.780%	-0.215%	-0.054%

Table 9.5: E_{tot} at $t = 3T_p$ of the airy wave

CONCLUSIONS Just like in the sloshing wave experiment of chapter 8, the reduced and mixed formulations have identical results. The results for these formulations can be summarised as:

- The period is calculated accurately for all n and Δt.
- Energy is conserved exactly for all n.

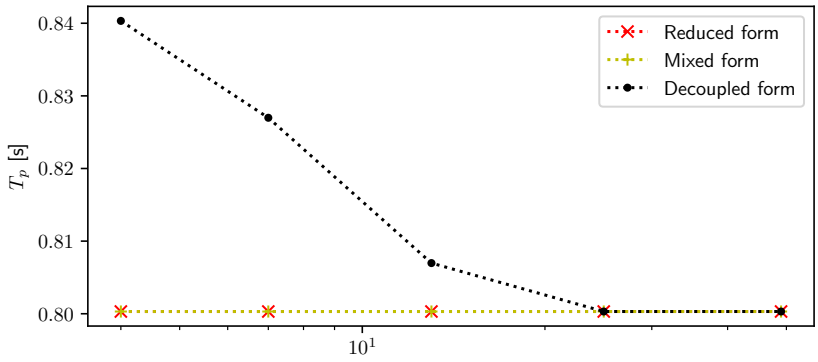


Figure 9.5: Convergence of T_p for the airy wave

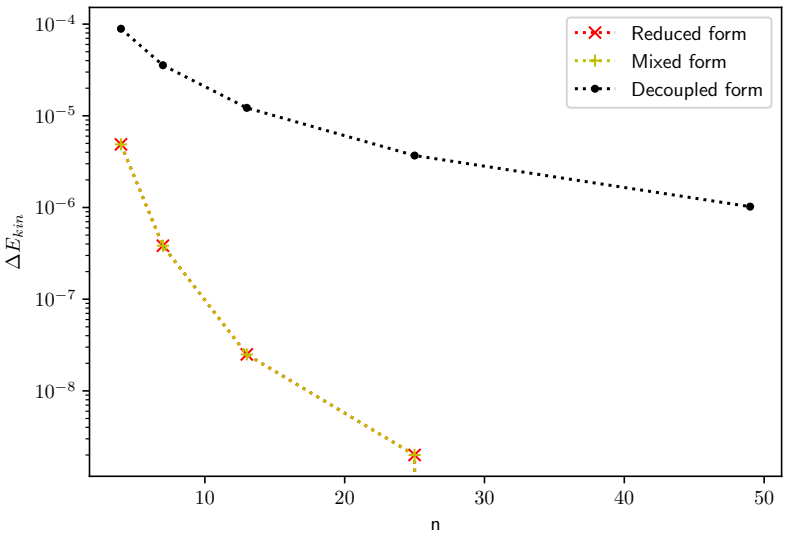


Figure 9.6: Convergence of E_{kin}

- The total energy is underestimated.
- The potential and kinetic energy show periodic behaviour. The amplitude of the kinetic energy is smaller than 10^-7 for $n \geq 13$

And, also like in the previous problem, the results for the decoupled formulation are less accurate:

- The period is calculated accurately for $n \geq 25$ and $\Delta t \leq 0.025 T_p^a$.
- Energy is conserved, but shows periodic behaviour.
- The total energy is underestimated.
- The potential and kinetic energy show periodic behaviour. The minimal amplitude of the kinetic energy is bigger than 10^-6, even for the most refined mesh.

Likely reasons for some of these results are already given in chapter 8. The main difference is the periodic behaviour of the kinetic and potential energy, that should really remain constant for this problem. The most probable reason is the fact that the airy wave moves over the mesh and is therefore discretised a little differently at different time steps, resulting in a periodic difference.

9.4 RESULTS FOR DIFFERENT WAVELENGTHS

The mixed formulation is used to check if T_p can be calculated accurately for different wave numbers. Therefore the wave number is varied by changing the wavelength and the length of the tank. This is done so there is always exactly one wave initialised along the complete free surface. The wave and domain length are varied from 0.5 to 28 meter and are set according to the test matrix in table 9.6. The experiments are performed using the parameters shown in table 9.7.

λ [m]	0.5	1	1.5	2.0	4.0	8.0
k [rad/m]	12.57	6.283	4.189	3.142	1.571	0.7854
λ [m]	12	16	20	24	28	
k [rad/m]	0.5240	0.3927	0.3142	0.2618	0.2244	

Table 9.6: Test matrix for wavelength comparison

RESULTS The period T_p is shown in figure 9.7, for the values of k in table 9.6. The dispersion relation (8.4) is plotted in the same figure. The results are very accurate: no optical difference can be seen.

PARAMETER	VALUE
Solver	Implicit midpoint
Order	2
Time step Δt	$0.025T_p^a$
n	13
Final time	$3.3T_p^a$

Table 9.7: Constant parameters for wavelength comparison

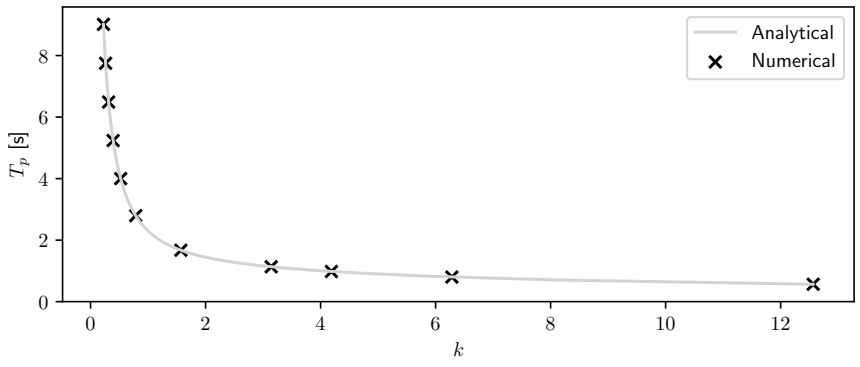


Figure 9.7: T_p for varying wave number k

10 TEST CASE III: TWO-DIMENSIONAL STEP WAVE

The problems described in chapters 8 and 9 have been tested and behaviour has been shown to be satisfactory. In order to see the benefits of IGA and NURBS meshes, a spatially more demanding problem will be thrown at the developed code in this chapter. This is done by initialising it with a step wave. A true Heaviside-like step function would demand a C^0-continuous point at the step. Instead, the step is imposed with a tanh-function. As described in chapter 7, the third formulation does not work on NURBS meshes. Only the reduced and the mixed formulations are therefore tested.

The initial conditions are given in the first paragraph. Tests are done for the settings in the second paragraph. The results of these tests are presented and analysed in the third paragraph.

10.1 INITIAL CONDITION

The step wave is initialised on a tank of 1 m². This tank is meshes in two different waves: with NURBS and with quadrilateral elements. A step wave is initialised by setting the value of the potential to $\phi(x, z, t = 0) = 0$ and the wave height according to

$$\eta = a \tanh(b(x - 0.5)) \frac{\cosh(2\pi z)}{\cosh(2\pi)}, \tag{10.1}$$

or equivalently, the time derivative of the potential ϕ to

$$\frac{d\phi}{dt} = -ag \tanh(b(x - 0.5)) \frac{\cosh(2\pi z)}{\cosh(2\pi)}. \tag{10.2}$$

Here, a is the wave height. The factor b is used to control the 'sharpness' of the step. It is set to 20, which results in a step that is quite sharp but can still be projected onto the coarsest mesh. Setting $a = 0.01$ results in an initial wave height at the free surface as shown in figure 10.1.

10.2 TEST PARAMETERS

The second order implicit midpoint solver is used for the reduced formulation and the first order implicit midpoint solver for the mixed formulation. Calculations are done for $t = 0$ to $t = 5$ seconds. Second order finite elements are used. These settings are summarised in table 10.1.

The number of nodes n in x-direction is increased from 5 to 65 in 4 refinement steps. The CFL number (8.7) is kept constant by decreasing

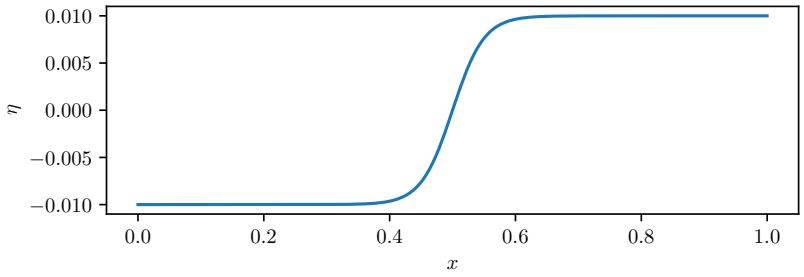


Figure 10.1: Initial wave height along the free surface for the step wave

time step Δt. This results in the test matrix in table 10.2. The corresponding number of Degrees-Of-Freedom (DOFs) is also shown in table 10.2 for the Finite Element (FE) and NURBS meshes. A FE mesh results in more DOFs, because every element only shares the DOFs with other elements that are on the boundaries with those elements.

PARAMETER	VALUE
Mesh types	Quadrilateral NURBS
Solver	Implicit midpoint
Order	2
Final time	5 [s]

Table 10.1: Parameters for the step wave problem

PARAMETERS	VALUES				
Δt	0.1	0.05	0.025	0.0125	0.00625
n	5	9	17	33	65
DOFs FE	81	289	1089	4225	16641
DOFs NURBS	36	100	324	1156	4356

Table 10.2: Parameters for step wave experiments

10.3 RESULTS AND ANALYSIS

As described in paragraph 6.6, the potential, kinetic and total energy can be calculated. Results for two formulations are, again, identical. Typical resulting time traces are shown in figure 10.2. The results for standard FEA and IGA are plotted in the same figure. Both methods show to be exactly energy conservative in this more complicated prob-

lem. The differences between the results for the FE and NURBS meshes are small. For $n = 17$ and $\Delta t = 0.025$ seconds, the measured difference between the total energy for the two mesh types is in the order of 10^{-8}. However, the NURBS mesh results in far less unknown degrees of freedom. So the same accuracy is obtained with less computations. This is illustrated in figure 10.3 where the initial value of the total energy E^0_{tot} is plotted for the five experiments.

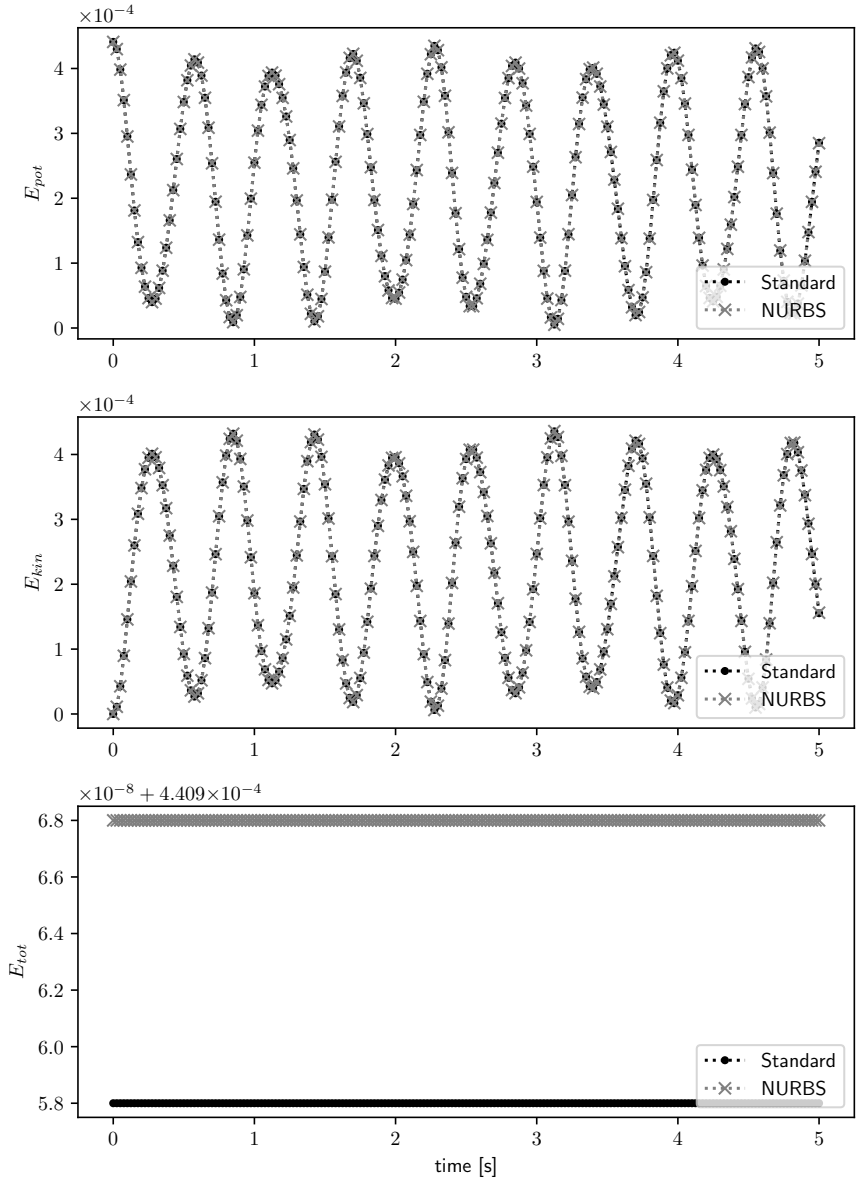


Figure 10.2: Time trace of E_{pot}, E_{kin} and E_{tot} - mixed form with $n = 17$ and $\Delta t = 0.025$ s

From an engineering perspective, it is interesting to look at a single value as well. For instance, the wave height at a certain point. As an example, the wave height at $(x, z) = (0.25, 1)$ is calculated at every

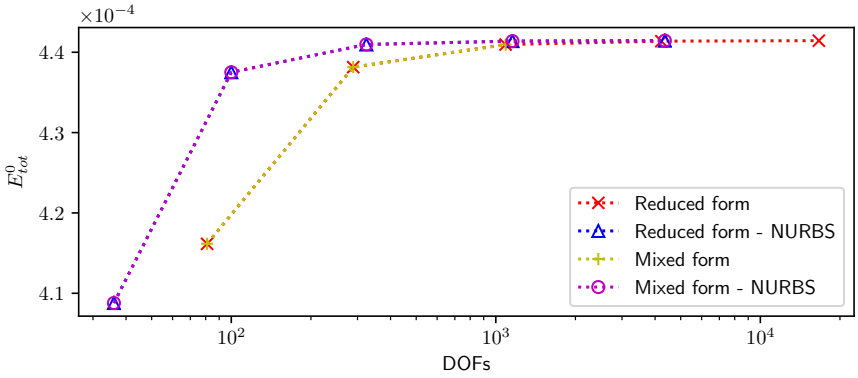
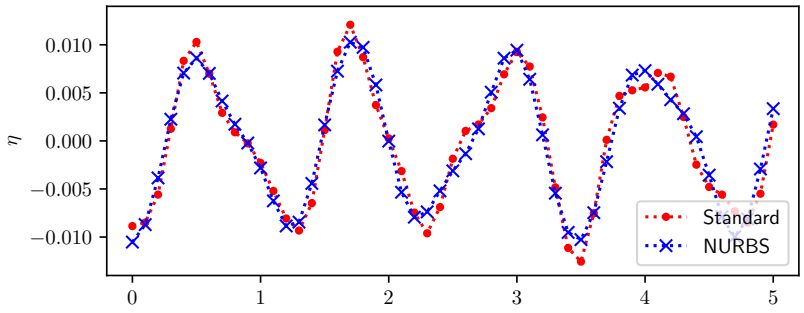


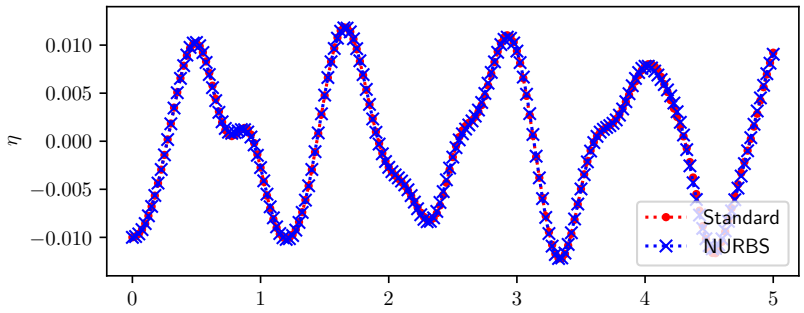
Figure 10.3: Convergence plot for $E_{tot}(t = 0)$

time step with both formulations. The resulting time traces are shown in figure 10.4 for three different mesh sizes and time steps. Differences between the FE and NURBS meshes are small and are only visible for the smallest mesh sizes.

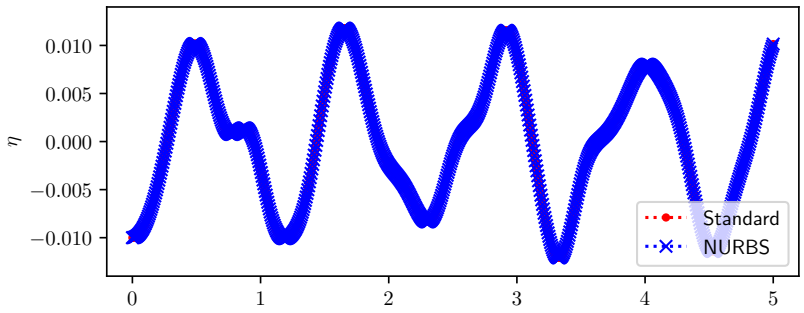
To see differences more clearly and see how the wave height converges, the convergence of the maximum wave height in the first 5 seconds is plotted in figure 10.5. The the time of this maximum is also plotted in this figure. In this plot, it is clearly visible that with a NURBS mesh, the same accuracy can be obtained with less degrees of freedom.



(a) $n = 5$ and $\Delta t = 0.1$



(b) $n = 17$ and $\Delta t = 0.025$



(c) $n = 65$ and $\Delta t = 0.00625$

Figure 10.4: Time traces of η at $(x, z) = (0.25, 1)$ for three mesh sizes - mixed form

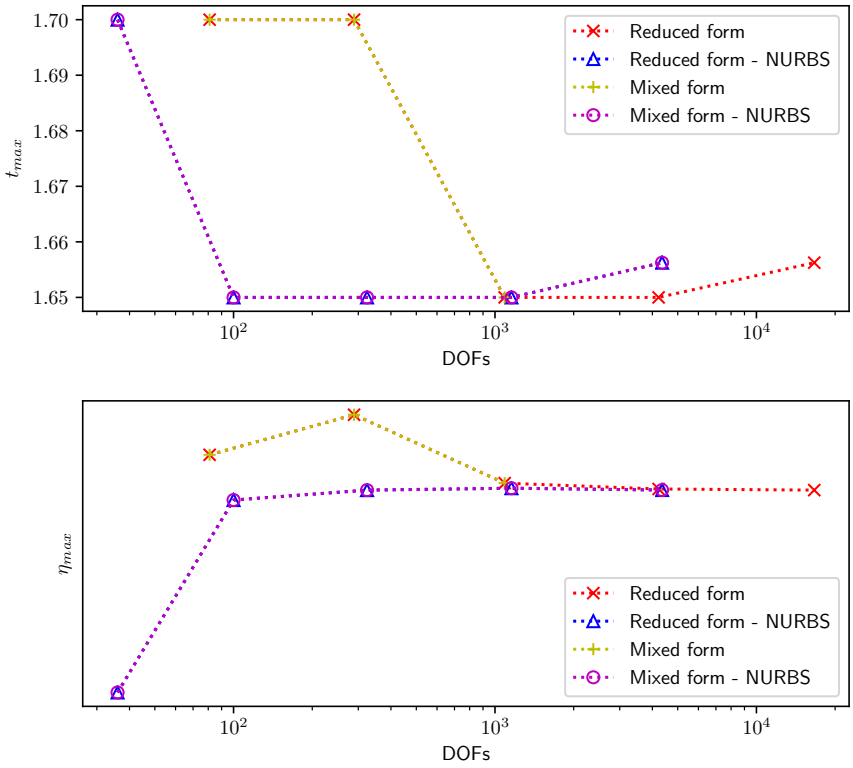


Figure 10.5: Convergence of the maximum wave height at $(x, z) = (0.25, 1)$

11

TEST CASE IV: THREE-RDIMENSIONAL SLOSHING WAVE

As a showcase and further test of the possibilities of IGA and NURBS meshes, a three-dimensional problem is treated as an example in this chapter. In this problem, the behaviour of a sloshing wave in a cubic tank with a monopile in the middle of the tank is calculated. The tank is 1 m^3 and the monopile is a cylinder with a radius of $\sqrt{0.005}$ meter. The geometry is shown in figure 11.1. Only the mixed formulation is used to calculate the behaviour of the waves in the tank during the first 5 seconds. Two mesh types are used: a NURBS mesh and a tetrahedron mesh. These two mesh types are described in the first paragraph. The initial conditions and test parameters are explained in the second and third paragraph. The last paragraph is dedicated to the results of the two meshes.

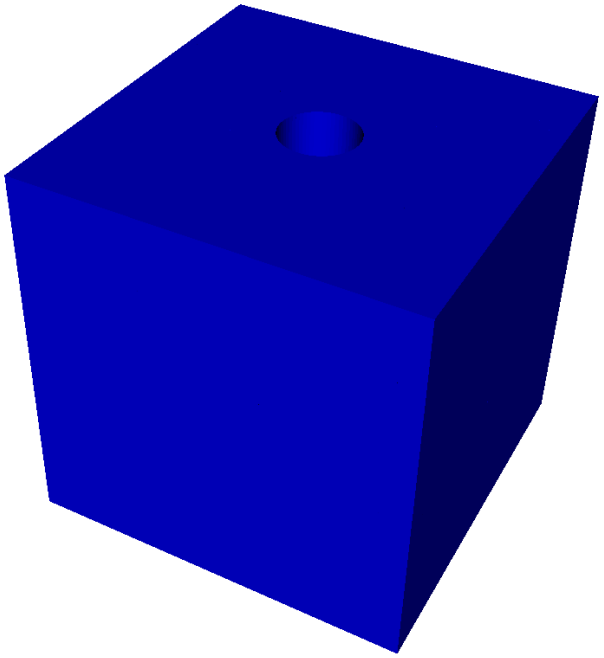


Figure 11.1: Geometry for the fourth test case: a cubic tank with a monopile

11.1 MESHES

Two different meshes are made: a second order NURBS mesh and a tetrahedron mesh. The tetrahedron mesh is made with GMSH, the NURBS mesh was coded by hand. A geometry file was with GMSH made that was used to generate four meshes with an increased refinement. In

PARAMETERS	VALUES			
Δt	0.05	0.025	0.0125	0.00625
n (NURBS)	5	9	17	33
DOFs FE	332	841	3674	23368
DOFs NURBS	192	720	3600	22032

Table 11.1: Parameters for three-dimensional wave experiments

generating the mesh an important advantage and a disadvantage of the NURBS mesh became clear. The advantage is that the NURBS mesh can be refined without going back to a geometry file, because the geometry is exact from the coarsest grid. This is clearly not the case for the tetrahedron mesh, because the cylinder is approximated with several straight lines. The geometry is better represented by increasing the number of lines, but to obtain a more refined mesh the mesh needs to be generated from the defining geometric file all over again. The disadvantage is that the creation of the initial NURBS mesh needed to be done by hand.

The number of mesh nodes n in $x-$direction of the NURBS mesh is increased from 5 to 33 in three refinement step. The time step Δt is decreased to keep the CFL-number (8.7) constant. The parameters for the mesh generation of the tetrahedron mesh are chosen in such a way that it results in a similar amount of degrees of freedom as the NURBS mesh when used with second order finite elements. This results in the text matrix in 11.1. Examples of the two mesh types are given in figure 11.2.

11.2 INITIAL CONDITIONS AND TEST PARAMETERS

A wave is initialised version of the sloshing wave in chapter 8 by projecting a potential that is described by

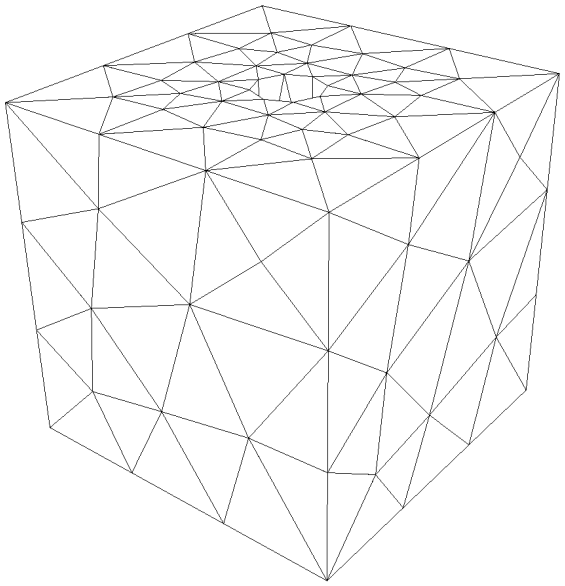
$$\phi(\boldsymbol{x}, t = 0) = 0.001 \cos(2\pi x) \cos(2\pi y) \frac{\cosh(2\pi z)}{\cosh(2\pi)}. \tag{11.1}$$

Wave height η is set to 0.

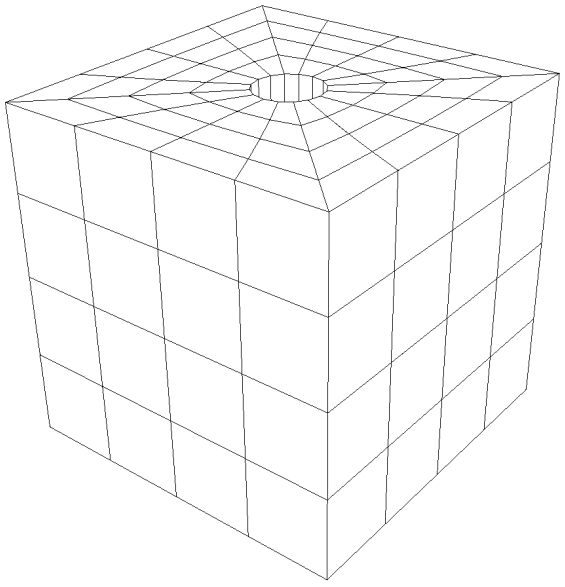
Second order basis functions are used on both mesh types. The wave is simulated until $t = 5$ seconds with a time step chosen according to table 11.1. An implicit midpoint solver is used. These settings are summarised in table 11.2

11.3 RESULTS

The results for the two meshes are compared on the following four points:



(a) Tetrahedron mesh with 841 degrees of freedom



(b) NURBS mesh with 720 degrees of freedom

Figure 11.2: Examples of the mesh for the tank with monopile

PARAMETER	VALUE
Mesh types	Tetrahedron
	NURBS
Solver	Implicit midpoint
Order	2
Final time	5 [s]

Table 11.2: Parameters for the three-dimensional sloshing wave problem

1. First the time traces of the three energy measures are compared to check the overall behaviour.
2. Secondly, a convergence plot is made of the values of the initial total energy.
3. Thirdly: out of engineering interest, the time trace of the wave height at $\boldsymbol{x} = (0.25, 0.25, 1.0)$ is plotted.
4. Finally, the convergence of the value of the last maximum at $\boldsymbol{x} = (0.25, 0.25, 1.0)$ is plotted.

ENERGY An example of the time traces of the kinetic, potential and total energy is shown in figure 11.3. It is clear that even for this even more difficult problem, energy conservation is exact. The convergence plot of the initial value of E_{tot} is shown in figure 11.4. Furthermore, it is clear that the initial projection is better for NURBS meshes. This can be concluded from the fact that E_{tot} at $t = 0$ does not change significantly if the NURBS mesh is refined. This is what was expected and promised, since the geometry is exact even on the coarsest mesh.

WAVE HEIGHT AND POTENTIAL As an example of a value that could be interesting from an engineering perspective, the value of the potential ϕ and the wave height η at $(\boldsymbol{x}) = (0.25, 0.25, 1.0)$ are calculated. The time traces for the coarsest meshes result in very distorted graphs. These are therefore not shown. Results for the other six, more refined meshes are shown in figures 11.5a,11.5b, 11.6a, 11.6b, 11.7a and 11.7b. It can be concluded from these plots that the time trace for the NURBS mesh is smoother and that the time traces seem to converge. As a convergence test, the value of η_{max} is given in figure 11.8, where η_{max} is defined as the value of the peak closest to $t = 1.5$. This is approximately the time of the maximum wave height of the most refined NURBS mesh. The time t at which this maximum occurs is plotted in the same figure. The value of η_{max} has not converged, but is within a band of around 10%.

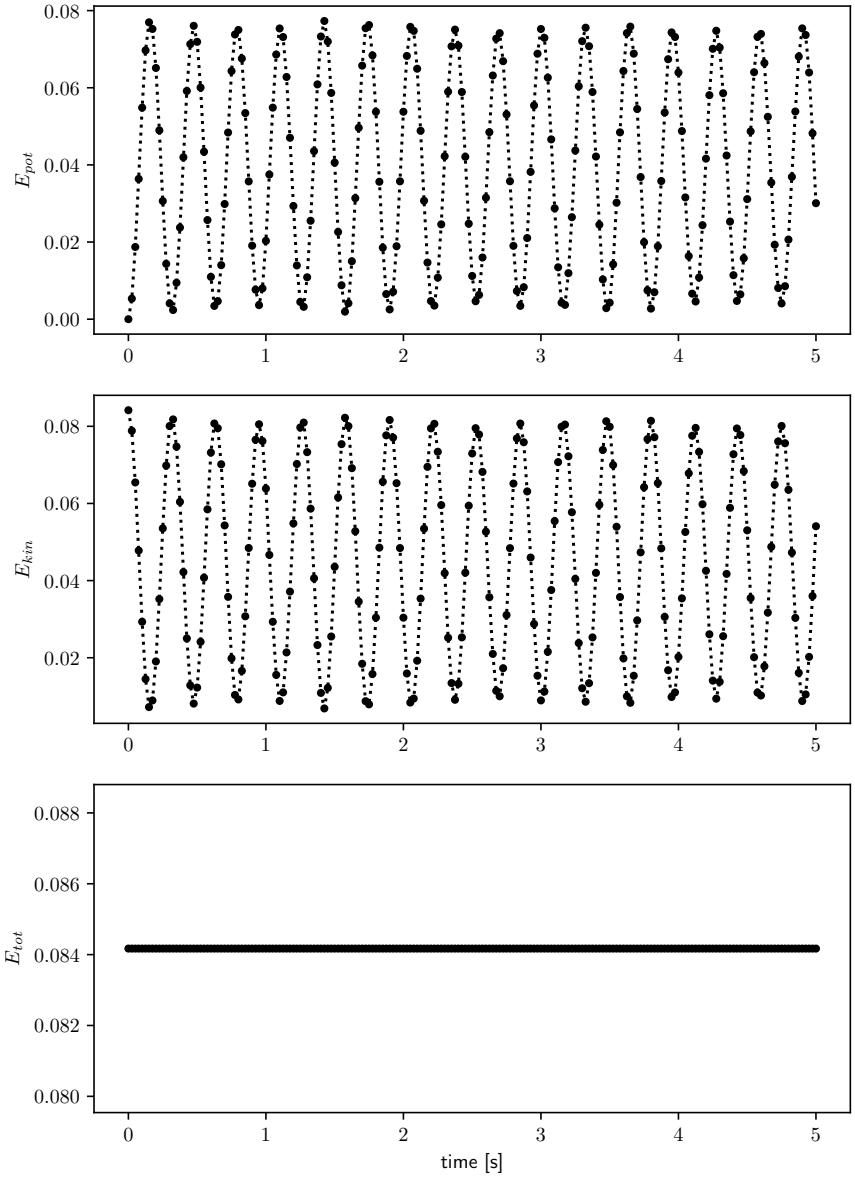


Figure 11.3: Time trace of E_{pot}, E_{kin} and E_{tot} - mixed form with DOFs= 841 and $\Delta t = 0.025$ - standard finite elements

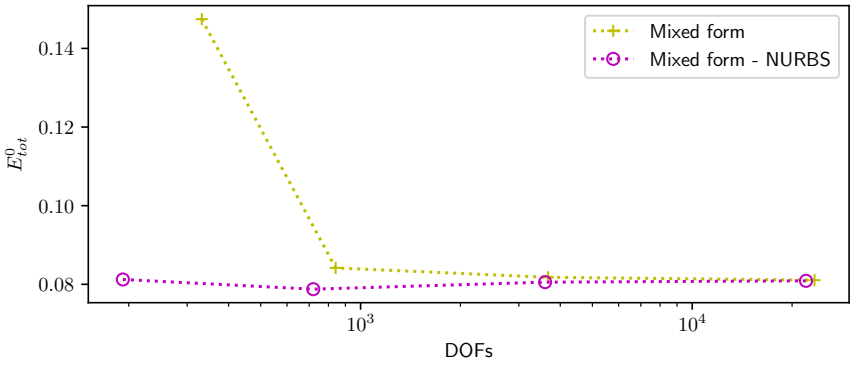
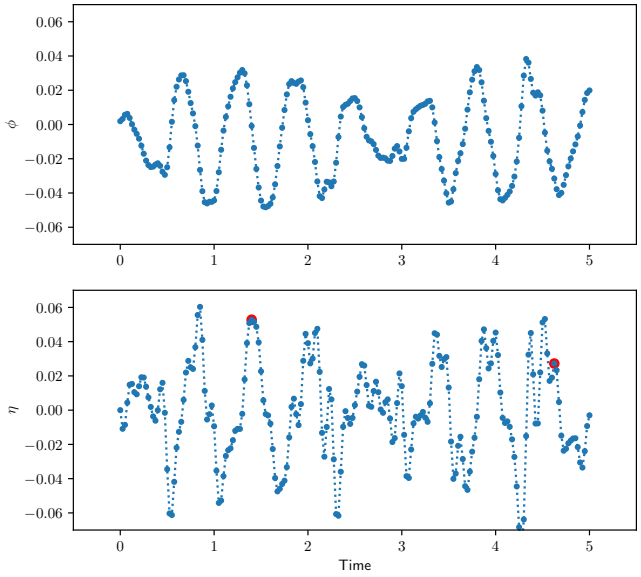


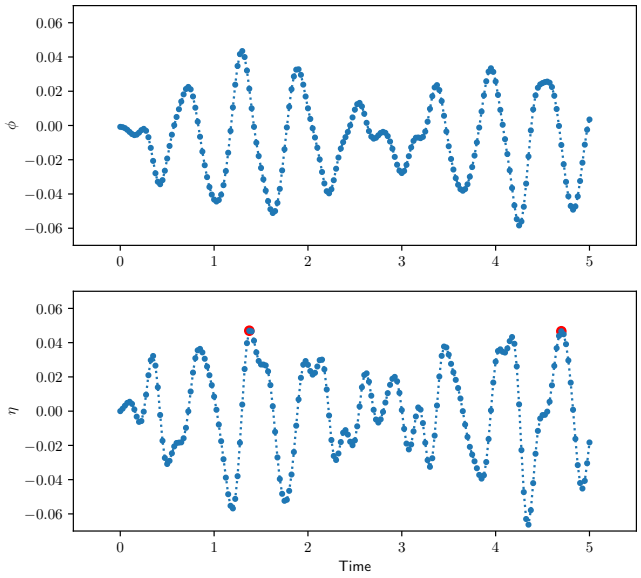
Figure 11.4: Convergence plot of E^0_{tot} and ΔE_{tot}

11.4 ILLUSTRATION OF THE RESULTING SLOSHING WAVE

The resulting sloshing wave can be illustrated with Visit. As an example, a few pictures of the resulting wave are shown in figures 11.9 and 11.9.

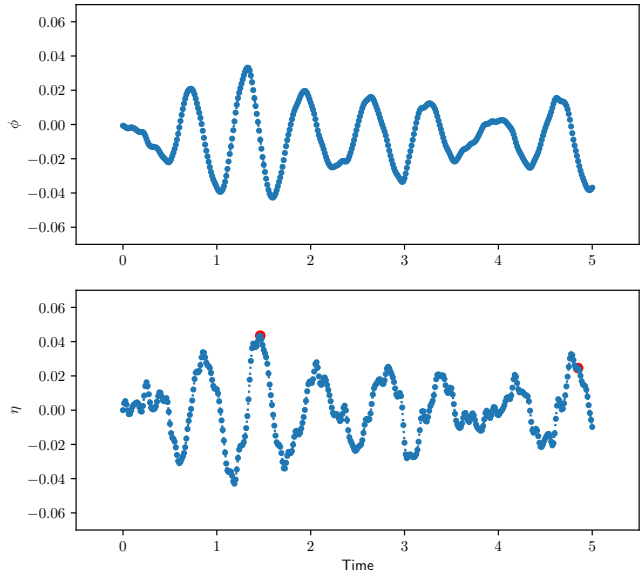


(a) Tetrahedron mesh - DOFs= 841 and $\Delta t = 0.025$

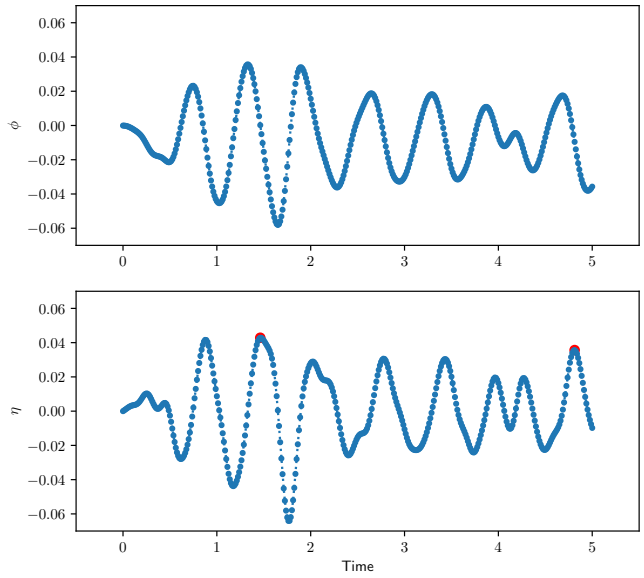


(b) NURBS mesh - DOFs= 720 and $\Delta t = 0.025$

Figure 11.5: Time traces of ϕ and η at $(0.25, 0.25, 1)$ - mixed form - refined once

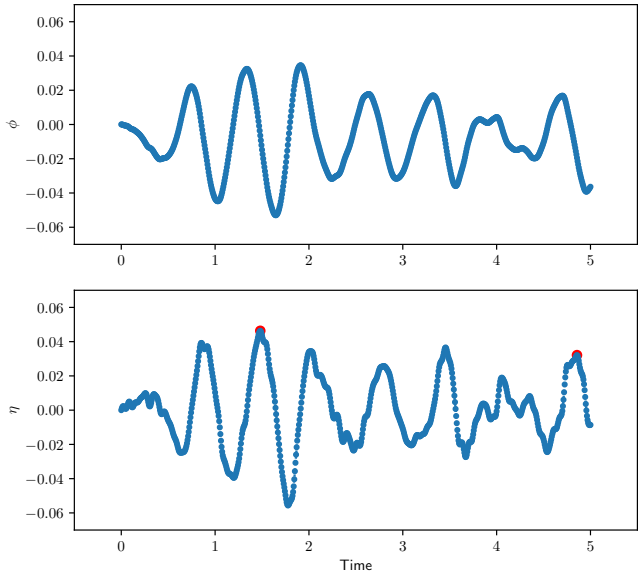


(a) Tetrahedron mesh - DOFs= 3674 and $\Delta t = 0.0125$

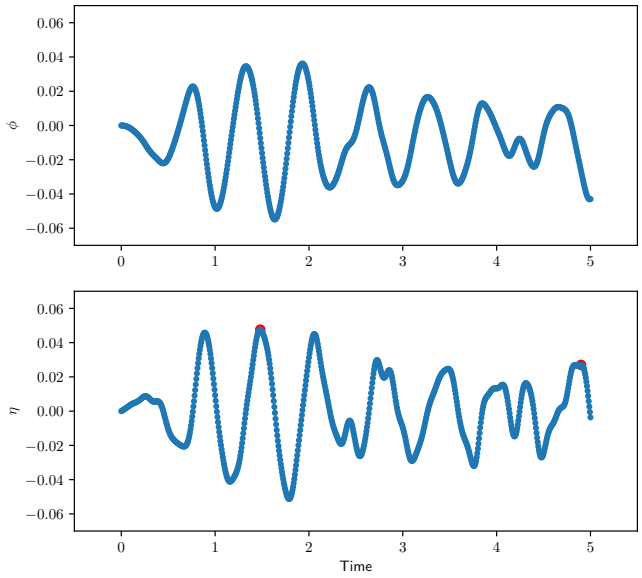


(b) NURBS mesh - DOFs= 3600 and $\Delta t = 0.0125$

Figure 11.6: Time traces of ϕ and η at $(0.25, 0.25, 1)$ - mixed form - refined twice



(a) Tetrahedron mesh - DOFs= 23368 and $\Delta t = 0.00625$



(b) NURBS mesh - DOFs= 22032 and $\Delta t = 0.00625$

Figure 11.7: Time traces of ϕ and η at $(0.25, 0.25, 1)$ - mixed form - refined thrice

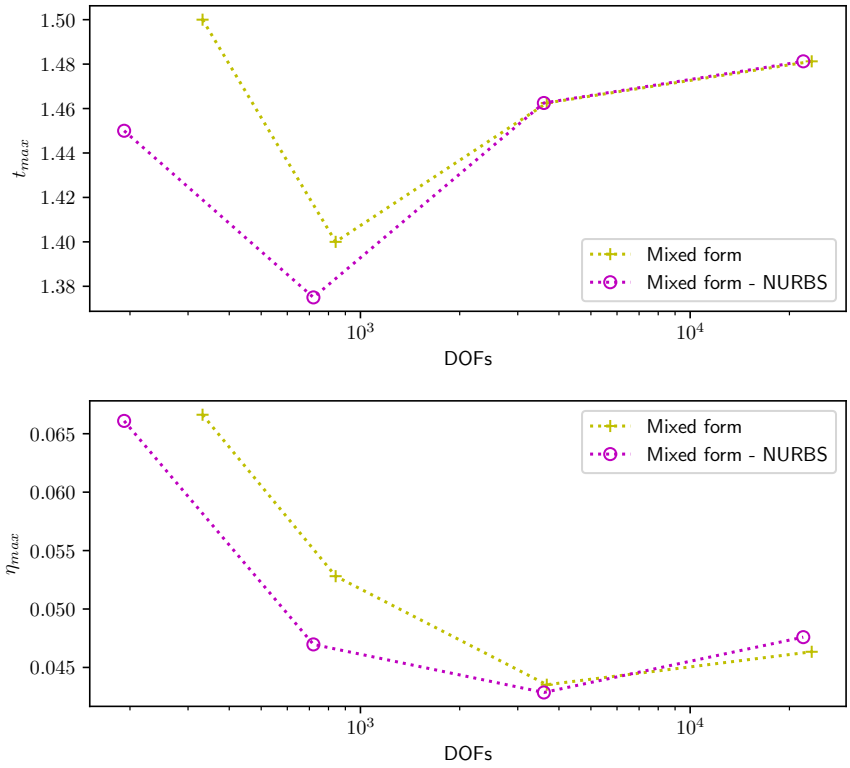
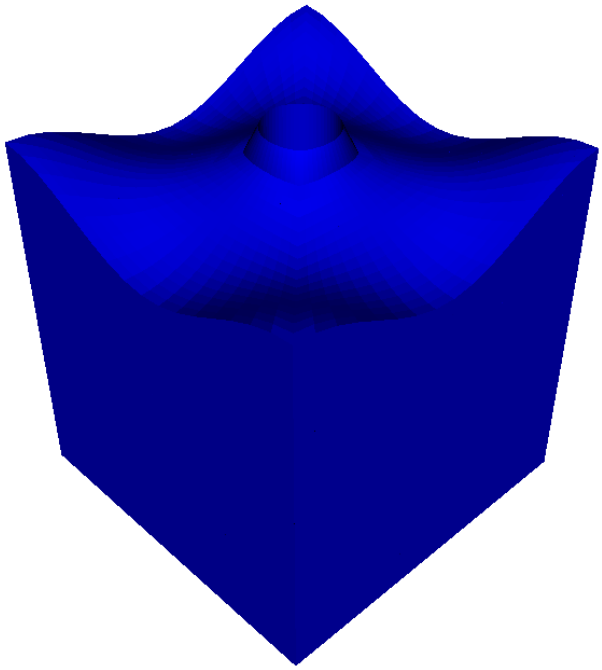
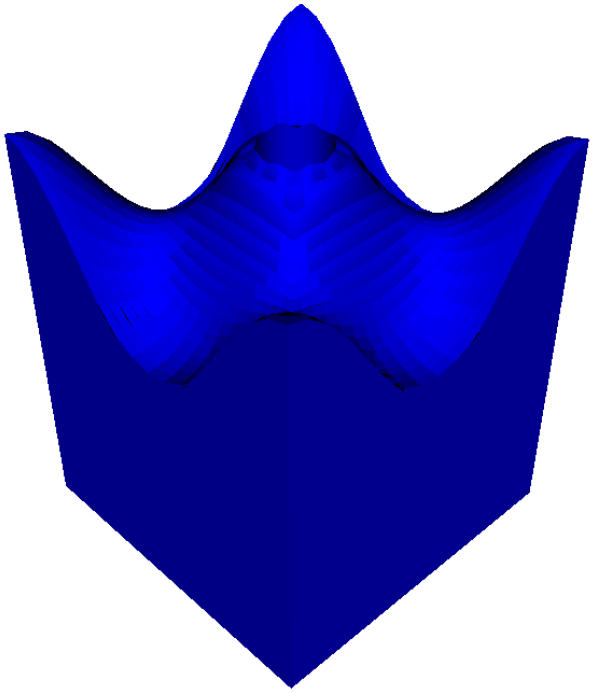


Figure 11.8: Convergence plot of η_{max} at (0.25, 0.25, 1.0).

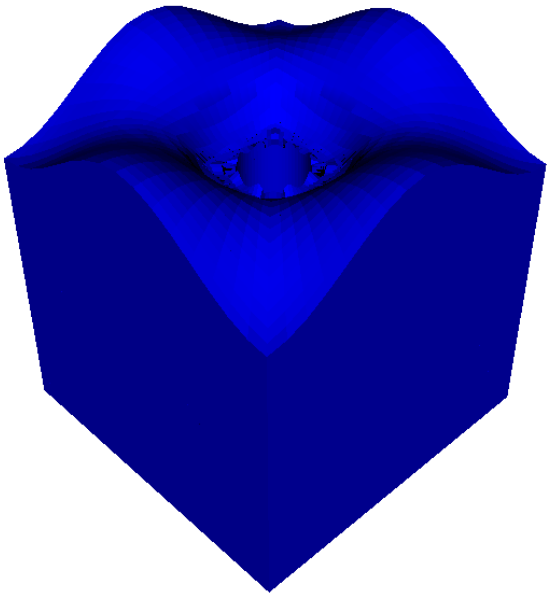


(a) $t = 0.0375$ s

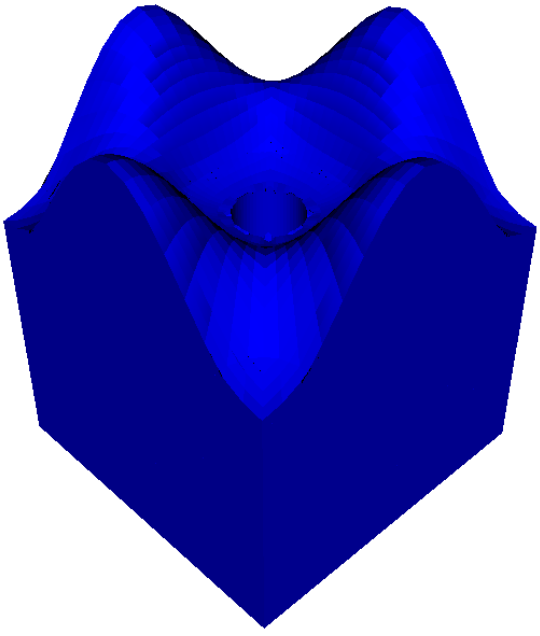


(b) $t = 0.15$ s

Figure 11.9: Illustration of the 3D sloshing wave - mixed formulation with a NURBS mesh, DOFs= 22032 and $\Delta t = 0.00625$s



(c) $t = 0.3875$ s



(d) $t = 1.1$ s

Figure 11.9: Illustration of the sloshing wave - mixed formulation with a NURBS mesh, DOFs= 22032 and $\Delta t = 0.00625$ s

12 CONCLUSIONS AND FUTURE WORK

In this thesis, a first step is made towards an 'all-inclusive' IGA-framework for potential flow problems of ships and offshore structures. The first goal of this research was finding out which of three different weak formulations is best suited for further development. The second goal was testing the advantages of IGA in a potential frame work. A secondary goal is testing MFEM, the C++ finite element library that was used.

FOLLOWED METHODOLOGY The linearised potential free surface problem was first written in strong form. This strong form was subsequently transformed into three different weak formulations by implementing the boundary conditions in three different ways. The first, *reduced* formulation is formed by combining the kinematic and dynamic boundary conditions on the free surface into one boundary condition, the second, *mixed* form by imposing the three boundary conditions directly and the third, *decoupled* form by decoupling the problem into a free surface and an interior problem.

The three formulations are first verified and validated by solving two simple two-dimensional problems that have analytical solutions. This is only done with FEA, because these problems are geometrically too simple to really see the differences with IGA. To see the advantages of IGA, the formulations are subsequently used to solve two more complex problems. The results are compared on the basis of the period and the kinetic/potential energy. The four problems are: (1) a sloshing wave, (2) an airy wave and (3) a step wave in a square tank of 1 m^2 and (4) a sloshing wave in a cubic tank of 1 m^3 with a cylinder in the middle.

RESULTS The formulations are first used to solve the two-dimensional sloshing and airy wave problems. Results of the reduced and mixed formulations are nearly identical and are very satisfactory. The period for the sloshing and airy wave problems are calculated exactly for quite coarse meshes and time step. Furthermore, energy is conserved exactly. This is a big feat and is very promising. Results for the decoupled formulation are less accurate: a more refined mesh and time step are needed to calculate the period with the same precision. Furthermore, the total energy is not constant. Instead, it show periodic behaviour. The most probable reason is the presence of the partial derivative in vertical direction in the formulation. This derivative cannot be calculated with sufficient accuracy on coarser meshes.

The step wave and three-dimensional sloshing wave problems are solved subsequently. Here, some limitations of MFEM became apparent. The current MFEM version does not support periodic NURBS meshes. On top of this, the third, decoupled formulation could not be used with NURBS meshes, because the partial derivative in vertical direction is only implemented for conventional meshes. However, it was already clear from the first tests that the reduced and mixed formulations performed much better. Therefore, only those two are used in the other two experiments. These two experiments made clear that both methods were also energy conservative for more complicated problems. It is again stressed that this a big feat! Comparing the results for the step waves for FEA with those for IGA showed that IGA results in comparable accuracy for less degrees of freedom. So the calculations are less expensive. The other big advantage of IGA became apparent on the geometrically more challenging three-dimensional sloshing wave problem. The results for the coarser meshes were much closer to those for the refined mesh with IGA than with FEA. This was expected, because the coarse NURBS mesh already represents the geometry exactly and refinement does not change the geometry. This is a big advantage, especially on geometries that have rounded shapes, which are ubiquitous in the maritime world.

Concluding, the two goals of this thesis are reached: the mixed formulation is the best formulation to develop further and it is shown that this formulation works well with IGA. Very important to note is the fact that the mixed formulation has shown to conserve energy exactly, even for more complicated geometries and problems.

FUTURE WORK Some missing MFEM features have been identified that would be welcome additions to the library. Firstly, addition of periodic NURBS meshes would be very welcome. Secondly, the decoupled formulation could be made available with NURBS meshes if a method to form the F matrix would be implemented. Additionally, *Visit* has been used to visualize the data. This visualizer does not fully support NURBS. Full support of NURBS is under development by supervisor Akkerman and will hopefully be available in a future release.

Another addition that would be necessary if the developed codes are used to solve real life engineering problems, is an easier method to generate NURBS meshes. A way to use a CAD program, like Rhinoceros, to create a NURBS geometry and be able to export this to a mesh file would be ideal.

Lastly, the code can be extended and enhanced to support external forces and inhomogeneous boundary conditions. This could be used to simulate the sloshing waves in, for instance, a moving LNG tank. Another improvement would be extending the code to be able to simulate non-linear waves.

Part IV

APPENDIX

A

SLOSHING WAVE

A.1 PROVE THAT THE INITIAL CONDITIONS SATISFY THE BOUNDARY CONDITIONS

The Laplace equation in two dimensions is satisfied:

$$\begin{aligned}\Delta\phi &= \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial z^2} \\ &= -ak^2\cos(kx)\cosh(kz) + ak^2\cos(kx)\cosh(kz) \\ &= 0,\end{aligned}$$

That the no-penetration boundary condition (5.2b) is also satisfied can be shown as follows:

$$\boldsymbol{n}\cdot\nabla\phi = \boldsymbol{n}\cdot\begin{bmatrix}-ak\sin(kx)\cosh(kz)\\ ak\cos(kx)\sinh(kz)\end{bmatrix}$$

At the bottom, where $z = 0$ and $\boldsymbol{n} = (0,-1)$, this equates to

$$\boldsymbol{n}\cdot\nabla\phi = (0,-1)\cdot\begin{bmatrix}-ak\sin(kx)\cdot 1\\ ak\cos(kx)\cdot 0\end{bmatrix} = 0$$

and at the left and right side, where $x = 0$ or $x = \lambda$ and $\boldsymbol{n} = (\pm 1, 0)$ to:

$$\boldsymbol{n}\cdot\nabla\phi = (\pm 1,0)\cdot\begin{bmatrix}-ak\sin(kx)\cdot 0\\ ak\cos(kx)\cdot 1\end{bmatrix} = 0$$

The initial surface elevation η can be defined with free surface boundary condition (5.2b):

$$\eta = -\frac{1}{g}\phi_t = -\frac{1}{g}\cdot 0 = 0$$

So (8.1) also satisfies the other free surface boundary condition:

$$\phi_z(x, z=0) = ak\cos(kx)\sinh(k\cdot 0) = 0$$

A.2 DERIVATION OF THE KINETIC AND POTENTIAL ENERGY

It is assumed that the potential can be written as

$$\phi(x,z,t) = X(x,z)T(t). \tag{A.1}$$

Setting $X(x,z)$ equal to the initial condition (8.1) it follows that $T(0) = 1$. A logical trial solution for $T(t) = \cos(\omega t)$. The angular frequency ω can be calculated from the combination of the free surface boundary conditions (5.2b) and (5.2c):

$$\phi_z = -\frac{1}{g}\phi_{tt}$$

Differentiating (8.1) with respect to z results in

$$\phi_z = ak\cos(kx)\sinh(kz)T(t)$$

and differentiating twice with respect to time results in

$$\phi_{tt} = -\omega^2 a\cos(kx)\cosh(kz)T(t).$$

So

$$\frac{1}{g}\omega^2\cosh(kz) = k\sinh(kz),$$

which is equal to well-known dispersion relation:

$$\omega = \sqrt{kg\tanh(kz)}.$$

Since the mass of the water in the tank is constant, a measure of the kinetic energy can be calculated with

$$E_{kin} = \int_A \frac{1}{2}(\nabla\phi)^2 \tag{A.2}$$

Wavelength λ is always a fraction of the length of the tank, so $k = 2n\pi$ with $n = 0, 1, 2, ...$ At $t = 0$ the free surface elevation $\eta = 0$, so:

$$
\begin{aligned}
E_{kin} &= \int_0^1 \int_0^1 \frac{1}{2}\big[(-a \cdot k \sin(kx) \cosh(kz))^2 \\
&\qquad + (a \cdot k \cos(kx) \sinh(kz))^2 \\
&\qquad - \frac{1}{2} a^2 k^2 \sin(2kx) \sinh(2kz)\big] dxdz \\
&= \frac{1}{2} a^2 k^2 \int_0^1 \int_0^1 \big[\sin^2(kx) \cosh^2(kz) + \cos^2(kx) \sinh^2(kz) \\
&\qquad - \sin(2kx) \sinh(2kz)\big] dxdz \\
&= \frac{1}{2} a^2 k^2 \int_0^1 \left[\left(\frac{x}{2} + \frac{1}{4k} \sin(2kx) \right) \cosh^2(kz) \right. \\
&\qquad + \left(\frac{x}{2} - \frac{1}{4k} \sin(2kx) \right) \sinh^2(kz) \\
&\qquad \left. + \frac{1}{2k} \cos(2kx) sinh(2kz) \right]_0^1 dz \\
&= \frac{1}{2} a^2 k^2 \int_0^1 \left[\frac{1}{2} \cosh^2(kz) + \frac{1}{2} \sinh^2(kz) \right] dz \\
&= \frac{1}{2} a^2 k^2 \left[\frac{1}{2} \left(\frac{z}{2} + \frac{1}{4k} \sinh(2kz) \right) \right. \\
&\qquad \left. + \frac{1}{2} \left(-\frac{z}{2} + \frac{1}{4k} \sinh(2kz) \right) \right]_0^1 \\
&= \frac{1}{2} a^2 k^2 \left[\frac{1}{4k} \sinh(2k) \right] \\
&= \frac{1}{8} a^2 k \sinh(2k)
\end{aligned}
$$

B

AIRY WAVE

B.1 PROVE THAT THE INITIAL CONDITIONS SATISFY THE BOUNDARY CONDITIONS

Using (9.1) with $t = 0$ in (2.4a) results in

$$\begin{aligned}\Delta\phi &= \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial z^2} \\ &= -ka\omega\frac{\cosh(k(h+z)}{\sinh(kh)}\sin(kx) + ka\omega\frac{\cosh(k(h+z)}{\sinh(kh)}\sin(kx) \\ &= 0.\end{aligned}$$

So the Laplace equation is satisfied. The no-penetration boundary condition only needs to be satisfied on the bottom, because there are no left and right boundaries. (5.2d) is satisfied, because

$$\boldsymbol{n}\cdot\nabla\phi = \boldsymbol{n}\cdot\left(a\omega\frac{1}{\sinh(kh)}\begin{bmatrix}\cosh(k(h+z))\cos(kx)\\ \sinh(k(h+z))\sin(kx)\end{bmatrix}\right).$$

and on the bottom the normal $\boldsymbol{n} = [0, -1]$ and $z = -h$.

The boundary condition (5.2c)) can be used to define η:

$$\eta = -\frac{1}{g}\phi_t = \frac{1}{g}a\frac{\omega^2}{k}\frac{\cosh(k(h+z))}{\sinh(kh)}\cos(kx-\omega t).$$

This can be used to check the other boundary condition on the free surface (5.2c):

$$\phi_z = a\omega\frac{\sinh(k(h+z)}{\sinh(kh)}\sin(kx-\omega t),$$

which equates to

$$\phi_z(x, z=0, t=0) = a\omega\sin(kx) \tag{B.1}$$

at the surface. Furthermore

$$\eta_t = \frac{1}{g}a\frac{\omega^3}{k}\frac{\cosh(k(h+z))}{\sinh(kh)}\sin(kx-\omega t)$$

which can be rewritten with the help of the dispersion relation (8.4 as:

$$\begin{aligned}
\eta_t(x, z = 0, t = 0) &= \frac{1}{g} a \frac{\omega^3}{k} \frac{\cosh(kh))}{\sinh(kh)} \sin(kx) \\
&= \frac{1}{g} a \frac{\omega \cdot g \cdot k tanh(kh)}{k} \frac{\cosh(kh))}{\sinh(kh)} \sin(kx) \\
&= a\omega \frac{\sinh(kh)}{\cosh(kh)} \frac{\cosh(kh))}{\sinh(kh)} \sin(kx) \\
&= a\omega \frac{\sinh(kh)}{\sinh(kh)} \sin(kx) \\
&= a\omega \sin(kx)
\end{aligned}$$

which is equal to (B.1), so the boundary condition is satisfied.

B.2 DERIVATION OF THE KINETIC AND POTENTIAL ENERGY OF THE AIRY WAVE

$$
\begin{aligned}
E_{kin} &= \int_0^1 \int_{-h}^0 \left[a\omega \frac{\cosh(k(h+z))}{\sinh(kh)} cos(kx) \right. \\
&\qquad \left. + a\omega \frac{\sinh(k(h+z))}{\sinh(kh)} \sin(kx) \right]^2 dzdx \\
&= \omega^2 \frac{a^2}{\sinh^2(kh)} \int_0^1 \int_{-1}^0 \left[\cosh^2(k(h+z)) \cos^2(kx) \right. \\
&\qquad + \sinh^2(k(h+z)) \sin^2(kx) + 2\cosh(k(h+z) \\
&\qquad \left. \cdot \sinh(k(h+z) \cos(kx) \sin(kx) \right] dzdx \\
&= \omega^2 \frac{a^2}{\sinh^2(kh)} \int_0^1 \int_{-1}^0 \left[\frac{1}{2} \cosh(2k(h+z)) - \sin^2(kx) + \frac{1}{2} \right. \\
&\qquad \left. + \frac{1}{2} \sin(2kx) \sinh(2k(h+z)) \right] dzdx \\
&= \omega^2 \frac{a^2}{\sinh^2(kh)} \int_0^1 \left[\frac{1}{2 \cdot 2k} \sinh(2k(h+z)) - \sin^2(kx)z + \frac{1}{2}z \right. \\
&\qquad \left. + \frac{1}{2 \cdot 2k} \sin(2kx) \cosh(2k(h+z)) \right]_{-1}^0 dx \\
&= \omega^2 \frac{a^2}{\sinh^2(kh)} \int_0^1 \left[\frac{1}{4k} \sinh(2kh) - \sin^2(kx) + \frac{1}{2} \right. \\
&\qquad \left. + \frac{1}{4k} \sin(2kx) \cosh(2kh) - \frac{1}{4k} \sin(2kx) \right] dx \\
&= \omega^2 \frac{a^2}{\sinh^2(kh)} \left[\frac{1}{4k} \sinh(2kh)x - \frac{x}{2} + \right. \\
&\qquad \frac{1}{4k} \sin(2kx) + \frac{1}{2}x \\
&\qquad \left. - \frac{1}{8k} \cos(2kx) \cosh(2kh) + \frac{1}{8k} \cos(2kx) \right]_0^1 dx \\
&= \omega^2 \frac{a^2}{\sinh^2(kh)} \left[\frac{1}{4k} \sinh(2kh) \right] \\
&= \omega^2 \frac{a^2}{4k \sinh(kh)}.
\end{aligned}
$$

With the dispersion relation (8.4) and $h = 1$ this can be written as

$$E_{kin} = \frac{1}{4} a^2 g$$

Setting $z = 0$ as the base line, a measure for the potential energy can be calculated with:

$$E_{pot} = Agh = \int_0^1 \eta(x) \cdot \frac{1}{2} \eta dx = \int_0^1 \frac{1}{2} g \eta(x)^2 \tag{B.2}$$

where $\eta(x)$ is defined according to (9.2), this results in

$$E_{pot} = \frac{1}{2}g\int_0^1 \left(\frac{1}{g}a\frac{\omega^2}{k}\frac{1}{\tanh(kh)}\cos(kx)\right)^2 dx$$

.

Which can be rewritten with the dispersion relation (8.4) as:

$$\begin{aligned} E_{pot} &= \frac{1}{2}g\int_0^1 \left(a\cos(kx)\right)^2 dx \\ &= \frac{1}{2}g\int_0^1 \left(a^2\cos^2(kx)\right) dx \\ &= \frac{1}{2}g\left[a^2\left(\frac{x}{2}+\frac{1}{4k}\sin(2kx)\right)\right]_0^1 \end{aligned}$$

So the potential energy is

$$E_{pot} = \frac{1}{4}a^2 g \tag{B.3}$$

and the total energy

$$E_{tot} = E_{kin} + E_{pot} = \frac{1}{2}a^2 g \tag{B.4}$$

C

MFEM CODE DESCRIPTION

The code for all three formulations can be divided into 5 steps:

1. Set up the variables, parameters and other options.
2. Read the mesh file and apply optional refinement and set the boundary conditions.
3. Apply the initial conditions.
4. Set up the wave operator in which the (block)matrices defined in chapter. 6 are set up.
5. Perform time integrations by looping with time step dt.

The first, third and fourth steps are described in more detail in the following paragraphs. The other steps are quite straightforward and are assumed to be self-explanatory.

C.1 CODE INITIALIZATION

The developed MFEM codes can be used to solve many types of problems by offering a range of choices. For instance: what mesh and what solver to use. Furthermore, several parameters, such as the time step and order of the finite elements, can be set to define this problem. These options can be set from the command line. All the available parameters are shown in table C.1.

C.2 INITIALIZATION OF THE GRIDFUNCTIONS

Gridfunctions phi_gf and dphidt_gf (reduced form), phi_gf and eta_gf (mixed and decoupled forms) are defined on the finite element space. The initial conditions are projected onto these gridfunctions. This projection is done in a project function. In this project function, a linear function b and bilinear form integrator M are defined on the finite element space domain. The initial conditions are set as the coefficient for the linear form integrator b. The domain mass integrator M is inverted with a conjugate gradient method which is allowed a maximum of 500 iterations to reach a convergence of 10^{-10} after being preconditioned with a Jacobi-type smoother. The respective grid functions gf are then calculated with

$$gf = \boldsymbol{M}^{-1}\boldsymbol{b}.$$

ID	USAGE	TYPE	COMMENTS
-m	Mesh file name	Const char	
-f	Free surface id number	Array of ints	
-r	Number of refinement	Integer	
-o	Finite element order	Integer	
-s	Solver type 1	Integer	e.g. 10 for gen-α or 32 for imp-midpoint
-Tp	Expected period [s]	Double	
-tf	End time simulation	Double	
-dt	Number of steps per period	Double	
-l	Initialization function library	Const char	
-vs	Visualization steps	Integer	Visualize every n-th step

Table C.1: Command line options for the codes

After this, the vectors phi and dphidt (reduced form) or eta (mixed and decoupled forms) are formed and filled with the true degrees of freedom. These vectors are used in the MFEM solvers to calculate the next time step. The true degrees of freedom of the gridfunctions are then refilled with the values at the new time step.

C.3 WAVE OPERATOR

In chapter 5, the problem is discretized in space and time. This discretization resulted in several (block)matrices. These matrices need to be set up in MFEM in order to solve the problem. This is done in the so-called 'wave operator'. The wave operator has four functions: the initialization, the implicit solver, the explicit solver and a post processor. In the initialization function, the matrices that do not depend on time step Δt are set up. The implicit and explicit functions are used by the MFEM solvers (like the mentioned generalized-α and implicit midpoint solvers) to calculate the values of the time derivatives. In these functions, time step dependent matrices are set up. Furthermore, the functions to calculate $\frac{d^2\phi}{dt^2}$, $\frac{d\phi}{dt}$ and/or $\frac{d\eta}{dt}$ are implemented. In the fourth function, the post processor, the kinetic, potential and total energy are calculated according to paragraphs 6.6. The matrices are set up with the help of the (bi)linear form integrators that are defined on domains or boundaries. Many integrators are readily available in

MFEM. The available integrators are described on the MFEM website ([54] for linear form integrators and [53] for bilinear form integrators). Since the matrices for the three formulations differ significantly, the wave operators for those three are described separately.

WAVE OPERATOR FOR THE REDUCED FORMULATION The reduced formulation is derived in paragraph 6.2 as

$$\boldsymbol{K\phi} + \frac{1}{g}\boldsymbol{M\ddot{\phi}} = \boldsymbol{0}, \qquad \text{(6.3, repeated)}$$

where the entries of stiffness matrix $\boldsymbol{K}$ are

$$K_{ij} = \left(\nabla N_i, \nabla N_j\right)_{\Omega},$$

and the entries of mass matrix $\boldsymbol{M}$ are

$$M_{ij} = \left(N_i, N_j\right)_{\Gamma_{fs}}.$$

The two matrices $\boldsymbol{K}$ and $\boldsymbol{M}$ are formed in the initialization function of the wave operator. Two bilinear forms are needed: a diffusion integrator K for the whole domain Ω and mass integrator M for the free surface boundaryΓ_{fs}. The class of domain integrators is available for both the FE and the NURBS meshes, but the class of boundary face integrators is not available for NURBS meshes. Therefore, the boundary integrator class is used. The $\frac{1}{g}$ factor is included in the mass integrator M. Both integrators are assembled and finalized to form matrices $\boldsymbol{K}$ and $\boldsymbol{M}$. Subsequently, the parameters that are used to estimate the inverse matrix $\boldsymbol{T}^{-1}$ are set: a conjugate gradient method is used with a maximum of 500 iterations to reach a convergence of 10^{-8}. These two values can be changed to obtain more accurate results or, oppositely, a faster code. Lastly, a Jacobi-smoother is set as pre-conditioner.

An explicit expression to calculate $\ddot{\phi}$ at time step $n+1$ cannot be defined, since the initial condition is $\boldsymbol{K\phi} = 0$, so $\boldsymbol{\ddot{\phi}} = \boldsymbol{M}^{-1}\left(-\boldsymbol{K\phi}\right) = 0$. Therefore, the explicit function is set to give the error message "This wave formulation only works for implicit time integrators".

An implicit method is available. A backward Euler method to calculate $\boldsymbol{\ddot{\phi}}^{n+1}$can be written as

$$\begin{aligned}\boldsymbol{\ddot{\phi}}^{n+1} &= \left(\boldsymbol{M} + \Delta t^2 \boldsymbol{K}\right)^{-1}\left(-\boldsymbol{K\phi}^n\right) \\ &= \boldsymbol{T}^{-1}\left(-\boldsymbol{K\phi}^n\right).\end{aligned}$$

In the code, the matrix $\boldsymbol{T}$ is defined as

$$\boldsymbol{T} = \boldsymbol{M} + fac_0\boldsymbol{K},$$

where fac_0 is defined by the used solver. For instance: in a backward Euler solver it would be set to Δt^2. In some methods, the factor fac_0 changes while it is a constant in other methods. Therefore, $\boldsymbol{T}$ is only recalculated if fac_0 has changed.

Finally, the post processor function is set to calculate the kinetic energy with

$$E_{kin} = \frac{1}{2}\boldsymbol{\phi}\left(\boldsymbol{K}\boldsymbol{\phi}\right),$$

the potential energy with

$$E_{pot} = \frac{1}{2}\dot{\boldsymbol{\phi}}\left(\boldsymbol{M}\dot{\boldsymbol{\phi}}\right)$$

and the total energy as a sum of those two:

$$E_{tot} = E_{kin} + E_{pot}.$$

WAVE OPERATOR FOR THE MIXED FORMULATION The discretized form of the mixed formulation is written in paragraph 6.4 as

$$\boldsymbol{A}\dot{\boldsymbol{x}} = -\boldsymbol{B}\boldsymbol{x}, \qquad \text{(6.4, repeated)}$$

where block matrix

$$\boldsymbol{A} = \begin{bmatrix} \frac{1}{2}\frac{\alpha}{g}\boldsymbol{M} & -\boldsymbol{M} \\ \frac{1}{2}\boldsymbol{M} & 0 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{K}_w & \frac{1}{2}\alpha\boldsymbol{M} \\ 0 & \frac{1}{2}g\boldsymbol{M} \end{bmatrix}.$$

The entries of the two block matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are formed in the initialization function. Again, two bilinear forms are defined on the finite element space: a diffusion integrator K on the whole domain and a mass integrator M on the free surface boundary. However, the diffusion integrator K is used to form two different stiffness matrices: one for the potential $\boldsymbol{\phi}$ ($\boldsymbol{K}_\phi$), but also one for wave height $\boldsymbol{\eta}$: K_η. This second matrix is used to solve a dummy Laplace problem for the wave height on the whole domain. This is done to obtain an invertible matrix and can be changed in further development of the code towards the inclusion of non-linear waves. This leads to a slightly altered $\boldsymbol{B}$-matrix that is called $\boldsymbol{Kb}$ in the code. The entries of the stiffness matrix $\boldsymbol{K}_\eta$ that are associated with the free surface are set to 0 so it does not interfere with boundary mass matrix $\boldsymbol{M}$. Subsequently, the parameters that are used to estimate the inverse matrix $\boldsymbol{T}^{-1}$ are set: a generalized minimal residual method (GMRES) method is used which is allowed a maximum of 500 iterations to reach a convergence of 10^{-8}. These two values can be changed to obtain more accurate results or, oppositely, a faster code. Lastly, a Gauß-Seidel smoother is set as pre-conditioner to calculate the inverse of $\boldsymbol{T}$.

The explicit function is again not available and set to give the error message "This wave formulation only works for implicit time integrators".

The implicit function is available and can be written as

$$\begin{aligned} \dot{\boldsymbol{x}}^{\boldsymbol{n+1}} &= \left(\boldsymbol{A} + \Delta t\boldsymbol{Kb}\right)^{-1}\left(-\boldsymbol{Kb}\boldsymbol{x}^n\right) \\ &= \boldsymbol{Tb}^{-1}\left(-\boldsymbol{Kb}\boldsymbol{x}^n\right) \end{aligned}$$

So in order to calculate $\dot{\boldsymbol{x}}^{n+1}$, block matrices $\boldsymbol{Tb}$ and $\boldsymbol{Kb}$ are formed from the stiffness matrices K_ϕ and K_η and the mass matrix $\boldsymbol{M}$:

$$\boldsymbol{Kb} = \begin{bmatrix} \boldsymbol{K_\phi} & \frac{1}{2}\alpha\boldsymbol{M} \\ 0 & \frac{1}{2}g\boldsymbol{M} \end{bmatrix} \tag{C.1}$$

and

$$\boldsymbol{Tb} = \begin{bmatrix} \Delta t\boldsymbol{K_\phi} + \frac{1}{2}\frac{\alpha}{g}\boldsymbol{M} & -\boldsymbol{M} + \frac{1}{2}\Delta t\alpha M \\ \frac{1}{2}M & \frac{1}{2}g\boldsymbol{M} + \boldsymbol{K}_\eta \end{bmatrix} \tag{C.2}$$

At every step, it is checked if time step Δt has changed. If this is the case, the block matrices are recalculated.

In the post processor, the three energy measures are calculated with:

$$\begin{aligned} E_{kin} &= \frac{1}{2}\boldsymbol{\phi}\left(\boldsymbol{K}_\phi\boldsymbol{\phi}\right) \\ E_{pot} &= \frac{1}{2}g\boldsymbol{\eta}\left(\boldsymbol{M\eta}\right) \\ E_{tot} &= E_{kin} + E_{pot} \end{aligned}$$

WAVE OPERATOR FOR THE DECOUPLED FORMULATION The discretized form of the decoupled formulation was derived in paragraph 6.4 as

$$\boldsymbol{A}\dot{\boldsymbol{x}} = -\boldsymbol{B}\boldsymbol{x}, \tag{6.6 repeated}$$

where this time

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{M} & 0 \\ 0 & \frac{g^2}{\alpha^2}\boldsymbol{M} \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{K_{nfs}} & g\boldsymbol{M} \\ \frac{g^2}{\alpha^2}\boldsymbol{F} & 0 \end{bmatrix}.$$

This time, three different bilinear form integrators are needed and set up in the initialization function. Apart from the diffusion and mass integrators an integrator F is needed to determine $\left(N_j, \frac{\partial}{\partial z}N_i\right)_{\Gamma_{fs}}$. This is done with a discontinuous Galerkin diffusion integrator, part of the boundary face integrator class, which is defined as

$$-\left\langle Q\nabla u\cdot\tilde{n}, v\right\rangle + \sigma\left\langle |u|, \{Q\nabla v\cdot\tilde{n}\}\right\rangle + \kappa\left\langle \{h^{-1}Q\}u, v\right\rangle$$

where Q is set to 1 and σ and κ are set to 0. Applied to the free surface where $\tilde{n} = (0, 1)$, this reduces to

$$-\left(v, \frac{\partial}{\partial z}\phi\right).$$

However, this integrator is not implemented in the boundary integrator class and cannot be used with NURBS meshes. Therefore, the decoupled formulation cannot be used with NURBS meshes. Hopefully this integrator will be implemented in a later version of MFEM.

The three integrators (K, M and F) are assembled and finalized and are used to form sparse matrices. The domain diffusion integrator is used to form two different stiffness matrices: $\boldsymbol{K_c}$ and $\boldsymbol{K_m}$. The rows in $\boldsymbol{K_m}$ that are associated with the free surface are eliminated to form $\boldsymbol{K}_{nfs}$. This is not done with $\boldsymbol{K_c}$, so this matrix can be used to calculate the kinetic energy in the post processor. Integrator F is used to form sparse matrix $\boldsymbol{F_m}$ and mass integrator M is used to form sparse matrix $\boldsymbol{M_m}$. The parameters that are used to calculate the inverse of matrix $\boldsymbol{T}$ are set: again, a GMRES solver is used with a tolerance of 10^{-8} and a maximum of 500 iterations to reach convergence. The inverse is preconditioned with a Gauß-Seidel smoother.

For the decoupled formulation, $\dot{\boldsymbol{x}}^n$ can be determined explicitly with

$$\dot{\boldsymbol{x}}^n = \boldsymbol{T}_e^{-1}\left(-Kb_e\boldsymbol{x}^n\right)$$

where blockmatrix

$$\boldsymbol{T}_e = \begin{bmatrix} 0 & g\boldsymbol{M}_m \\ \boldsymbol{F}_m & 0 \end{bmatrix}$$

and

$$\boldsymbol{Kb}_e = \begin{bmatrix} \boldsymbol{K}_m + \boldsymbol{M}_m & 0 \\ 0 & \boldsymbol{K}_m + \boldsymbol{M}_m \end{bmatrix}$$

The inverse $\boldsymbol{T}_e^{-1}$ is calculated with the same settings as the implicit $\boldsymbol{T}^{-1}$. This inversion is done in the initialization function and the inverted matrix can subsequently be used to determine $\dot{\boldsymbol{x}}^n$ in the explicit function of the wave operator.

In the implicit function, $\dot{\boldsymbol{x}}^{n+1}$ is determined with

$$\dot{\boldsymbol{x}}^{n+1} = \boldsymbol{T}^{-1}\left(-\boldsymbol{K}_e\boldsymbol{x}^n\right)$$

where the block matrices are (re)calculated if Δt has changed or in the first run with:

$$\boldsymbol{T} = \begin{bmatrix} \Delta t\boldsymbol{K_m} + \boldsymbol{M_m} & \Delta t g\boldsymbol{M_m} \\ \Delta\frac{g^2}{\alpha^2}\boldsymbol{F_m} & \Delta\boldsymbol{K_m} + \frac{g^2}{\alpha^2}\boldsymbol{M_m} \end{bmatrix}$$

and

$$\boldsymbol{Kb} = \begin{bmatrix} 0 & g\boldsymbol{M_m} \\ \frac{g^2}{\alpha^2}\boldsymbol{F_m} & 0 \end{bmatrix}.$$

Finally, in the post processor the three energy measures are calculated with

$$E_{kin} = \frac{1}{2}\boldsymbol{\phi}\left(\boldsymbol{K_c}\boldsymbol{\phi}\right),$$

$$E_{pot} = \frac{1}{2}g\boldsymbol{\eta}\left(\boldsymbol{M_m}\boldsymbol{\eta}\right)$$

and

$$E_{tot} = E_{kin} + E_{pot}$$

BIBLIOGRAPHY

[1] I. Akkerman, Y. Bazilevs, V.M. Calo, T.J.R. Hughes and S. Hulshoff. 'The role of continuity in residual-based variational multiscale modeling of turbulence'. In: *Computational Mechanics* 41:371–378 (2008).

[2] I. Akkerman, Y. Bazilevs, C. Kees and M. Farthing. 'Isogeometric analysis of free-surface flow'. In: *Journal of Computational Physics* 230:4137–4152 (2011).

[3] F. Auricchio, F. Calabro, T.J.R. Hughes, A. Reali and G. Sangalli. 'A simple algorithm for obtaining nearly optimal quadrature rules for nurbs-based isogeometric analysis'. In: *Computer Methods in Applied Mechanics and Engineering* 249:15–27 (2012).

[4] Y. Bazilevs and I. Akkerman. 'Large eddy simulation of turbulent Taylor-Couette flow using isogeometric analysis and the residual-based variational multiscale method'. In: *Journal of Computational Physics* 229(9):3402 – 3414 (2010).

[5] Y. Bazilevs, L. Beirao da Veiga, J.A. Cottrell, T.J.R. Hughes and G. Sangalli. 'Isogeometric analysis: Approximation, stability and error estimates for *h*-refined meshes.' In: *Mathematical Models and Methods in Applied Sciences* 16:1031–1090 (2006).

[6] Y. Bazilevs, V.M. Calo, Y. Zhang and T.J.R. Hughes. 'Isogeometric fluid-structure interaction analysis with applications to arterial blood flow'. In: *Computational Mechanics* 38:310–322 (2006).

[7] Y. Bazilevs, V.M. Calo, T.J.R. Cottrel J.A. Hughes, A. Reali and G. Scovazzi. 'Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows'. In: *Computer Methods in Applied Mechanics and Engineering* 197:173–201 (2007).

[8] Y. Bazilevs, J.R. Gohean, T.J.R. Hughes, R.D. Moser and Y. Zhang. 'Patient-specific isogeometric fluid-structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device.' In: *em Computer Methods in Applied Mechanics and Engineering* 198:3534–3550 (2009).

[9] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott and T.W. Sederberg. 'Isogeometric analysis using t-splines'. In: *Computer Methods in Applied Mechanics and Engineering* 199(5-8):229–263 (2010).

[10] Y. Bazilevs, X. Deng, A. Korobenko, F. Lanza di Scalea, M.D Todd and S.G. Taylor. 'Isogeometric fatigue damage prediction in large-scale composite structures driven by dynamic sensor data'. In: *Journal of Applied Mechanics* 82(9):091008 (2015).

[11] Y. Bazilevs, M.S. Pigazzini, A. Ellison and H. Kim. 'A new multi-layer approach for progressive damage simulation in composite laminates based on isogeometric analysis and kirchhoff–love shells. part i: basic theory and modeling of delamination and transverse shear'. In: *Computational Mechanics* 62(3):563–585 (2018).

[12] Y. Bazilevs, Y. Zhang, V.M. Calo, S. Goswami, C.L. Bajaj and T.J.R. Hughes. 'Isogeometric analysis of blood flow: a nurbs-based approach'. In: *Computational Modelling of Objects Represented in Images. Fundamentals Methods and Applications: Proceedings of the International Symposium CompIMAGE 2006 (Coimbra, Portugal, 20-21 October 2006), page 91* (2018).

[13] L. Beirao Da Veiga, A. Buffa, J Rivas and Sangalli G. 'Some estimates for h–p–k-refinement in isogeometric analysis.' In: *Numerische Mathematik* 118(2):271–305 (2011).

[14] K.A. Belisbassakis, T.P. Gerostathis, C.G. Politis, P.D. Kaklis, A.I. Ginnis and D.N. Mourkogiannis. 'A novel BEM-ISOGEOMETRIC method with application to the wavemaking resistance problem of bodies at constant speed'. In: *13th Congres of Intl. Maritime Assoc. of Mediterranean* (2009).

[15] K.A. Belisbassakis, T.P. Gerostathis, K.V. Kostas, C.G. Politis, P.D. Kaklis, A.I. Ginnis and C. Feurer. 'A BEM-isogeometricmethod for the shipwave-resistance problem'. In: *Ocean Engineering 60* (2013).

[16] A. Buffa, C. De Falco and G. Sangalli. 'Isogeometric analysis: new stable elements for the stokes equation'. In: *International Journal for Numerical Methods in Fluids* (2010).

[17] A. Buffa, C De Falco and G. Sangalli. 'Isogeometric analysis: stable elements for the 2d Stokes equation'. In: *International Journal for Numerical Methods in Fluids* 65:1407–1422 (2011).

[18] A. Buffa, G. Sangalli and R. Vázquez. 'Isogeometric analysis in electromagnetics: B-splines approximation'. In: *Computer Methods in Applied Mechanics and Engineering* 199(17-20):1143–1152 (2010).

[19] A. Buffa, C De Falco, G. Sangalli and R. Vázquez. 'Isogeometric discrete differential forms in three dimensions'. In: *SIAM Journal on Numerical Analysis* 49:818–844 (2011).

[20] H. Childs et al. *VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data*. 2012.

[21] S. Cho and S.H. Ha. 'Isogeometric shape design optimization: exact geometry and enhanced sensitivity'. In: *Structural and Multidisciplinary Optimization* 38(1):53 (2009).

[22] J. Chung and G.M. Hulbert. 'time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-alpha method'. In: *Applied mechanics 60:371-375* (1993).

[23] O. Colomés, S. Badia, R. Codina and J. Principe. 'Assessment of variational multiscale models for the large eddy simulation of turbulent incompressible flows'. In: *Computer Methods in Applied Mechanics and Engineering* 285:32–63 (2015).

[24] *Dibley Marine Design*. `http://www.dibleymarine.com/dibley/dibley/maxsurf.html`. Last accessed: 03/10/2018.

[25] M.R. Dörfel, B. Jüttler and B. Simeon. 'Adaptive isogeometric analysis by local h-refinement with t-splines'. In: *Computer methods in applied mechanics and engineering* 199(5-8):264–275 (2010).

[26] M.F.P. ten Eikelder and I. Akkerman. 'Correct energy evolution of stabilized formulations: The relation between VMS, SUPG and GLS via dynamic orthogonal small–scales and isogeometric analysis. I: The convective–diffusive context'. In: *Computer Methods in Applied Mechanics and Engineering* 331:259–280 (2018).

[27] M.F.P. ten Eikelder and I. Akkerman. 'Correct energy evolution of stabilized formulations: The relation between VMS, SUPG and GLS via dynamic orthogonal small–scales and isogeometric analysis. II: The incompressible Navier–Stokes equations'. In: *Computer Methods in Applied Mechanics and Engineering* 340:1135–1154 (2018).

[28] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of Finite Elements*. Springer, 2004. ISBN: 0-387-20574-8.

[29] E.J. Evans, M.A. Scott, X. Li and D.C. Thomas. 'Hierarchical t-splines: Analysis-suitability, bézier extraction and application as an adaptive basis for isogeometric analysis.' In: *Computer Methods in Applied Mechanics and Engineering* 284:1–20 (2015).

[30] *GLVis: OpenGL Finite Element Visualization Tool*. `glvis.org`. DOI: `10.11578/dc.20171025.1249`.

[31] A.I. Ginnis, K.V. Kostas, C.G. Politis, P.D. Kaklis, K.A. Belisbassakis, T.P. Gerostathis, M.A. Scott and T.J.R. Hughes. 'Isogeometric boundary-element analysis for the wave-resistance problem using T-splines'. In: *Comput. Methods Appl. Mech. Engrg. 279* (2014).

[32] R. Hiemstra. 'IsoGeometric Mimetic Methods'. Master of Science Thesis. TU Delft, 2011.

[33] R.R. Hiemstra, F. Calabro, D. Schillinger and T.J.R. Hughes. 'Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis'. In: *Computer Methods in Applied Mechanics and Engineering* 316:966–1004 (2017).

[34] K. Ho-Le. 'Finite element mesh generation methods: a review and classification'. In: *Computer-Aided Design* 20.1 (1988), pp. 27–38. ISSN: 0010-4485. DOI: https://doi.org/10.1016/0010-4485(88)90138-8. URL: http://www.sciencedirect.com/science/article/pii/0010448588901388.

[35] J. Holmen, T.J.R. Hughes, A.A. Oberai and G.N. Wells. 'Sensitivity of the scale partition for variational multiscale LES of channel flow'. In: *Physics of Fluids* 16:824–827 (2004).

[36] M.C. Hsu, D. Kamensky, Y. Bazilevs, M.S. Sacks and T.J.R. Hughes. 'Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation'. In: *Computational mechanics* 54(4):1055–1071 (2014).

[37] M.C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M.C.H Wu, J. Mineroff, A. Reali, Y. Bazilevs and M.S. Sacks. 'Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with t-splines and fung-type material models.' In: *Computational mechanics* 55(6):1211–1225 (2015).

[38] T.J.R. Hughes. 'Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods'. In: *Computer Methods in Applied Mechanics and Engineering* 127:387–401 (1995).

[39] T.J.R. Hughes, J.A. Cottrell and Y. Bazilevs. 'Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement'. In: *Comput. Methods Appl. Mech. Engrg 194* (2005).

[40] T.J.R. Hughes, A. Reali and G. Sangalli. 'Efficient quadrature for nurbs-based isogeometric analysis'. In: *Computer methods in applied mechanics and engineering* 199(5-8):301–313 (2010).

[41] T.J.R. Hughes and G. Sangalli. 'Variational multiscale analysis: the fine-scale Green's function, projection, optimization, localization, and stabilized methods'. In: *SIAM Journal of Numerical Analysis* 45:539–557 (2007).

[42] T.J.R. Hughes, G. Feijóo, L. Mazzei and J.B. Quincy. 'The variational multiscale method – A paradigm for computational mechanics'. In: *Computer Methods in Applied Mechanics and Engineering* 166:3–24 (1998).

[43] *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley and Sons, Incorporated, 2009.

[44] K.E. Johannessen, T. Kvamsdal and T. Dokken. 'Isogeometric analysis using lr b-splines'. In: *Computer Methods in Applied Mechanics and Engineering* 269:471–514 (2014).

[45] D. Kamensky, M.C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks and T.J.R. Hughes. 'An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves.' In: *Computer methods in applied mechanics and engineering* 2015 (284:1005–1053).

[46] J. Katz and A. Plotkin. *Low-speed aerodynamics. From wing theory to panel methods*. International Edition 1991. Singapore: McGraw-Hill, Inc., 1991. ISBN: 0-07-050446-6.

[47] B. Koobus and C. Farhat. 'A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes – application to vortex shedding.' In: *Computer Methods in Applied Mechanics and Engineering* 193:1367–1383 (2004).

[48] A. Korobenko, X. Deng, J. Yan and Y. Bazilevs. 'Recent Advances in Fluid–Structure Interaction Simulations of Wind Turbines'. In: *Advances in Computational Fluid-Structure Interaction and Flow Simulation: New Methods and Challenging Computations*. Ed. by Yuri Bazilevs and Kenji Takizawa. Springer International Publishing, 2016, pp. 489–500. ISBN: 978-3-319-40827-9.

[49] K.V. Kostas, A.I. Ginnis, C.G. Politis and P.D. Kaklis. 'Ship-hull shape optimization with a T-spline based BEM–isogeometric solver'. In: *Comput. Methods Appl. Mech. Engrg. 284* (2014).

[50] K.V. Kostas, A.I. Ginnis, C.G. Politis and P.D. Kaklis. 'Shape-optimization of 2D hydrofoils using an Isogeometric BEM solver'. In: *Computer-Aided Design 82* (2017).

[51] E. F. Lins, R.N. Elias, F.A. Rochinha and A.L.G.A. Coutinho. 'Residual-based variational multiscale simulation of free surface flows'. In: *Computational Mechanics* 46:545–557 (2010).

[52] S H Lo. 'Finite element mesh generation and adaptive meshing'. In: *Progress in Structural Engineering and Materials* 4.4 (2002).

[53] *MFEM: Bilinear form integrators*. `http://mfem.org/bilininteg/`. Last accessed: 22/09/2018.

[54] *MFEM: Linear form integrators*. http://mfem.org/lininteg/. Last accessed: 22/09/2018.

[55] *MFEM: Modular Finite Element Methods Library*. `mfem.org`. DOI: `10.11578/dc.20171025.1248`.

[56] *Maxsurf - integrated naval architecture software*. `https://www.maxsurf.net/index.html`. Last accessed: 03/10/2018.

[57] G. Moutsanidis, D. Kamensky, J.S. Chen and Y. Bazilevs. 'Hyperbolic phase field modeling of brittle fracture: Part ii—immersed iga–rkpm coupling for air-blast–structure interaction'. In: *Journal of the Mechanics and Physics of Solids* 121:114–132 (2018).

[58] N Nguyen-Thanh, S.P.A. Nguyen-Xuan H. Bordas and T Rabczuk. 'Isogeometric analysis using polynomial splines over hierarchical t-meshes for two-dimensional elastic solids.' In: *Computer Methods in Applied Mechanics and Engineering* 200(21-22):1892–1908 (2011).

[59] N. Nguyen-Thanh, K. Zhou, X. Zhuang, P. Areias, H. Nguyen-Xuan, y. Bazilevs and T. Rabczuk. 'Isogeometric analysis of large-deformation thin shells using rht-splines for multiple-patch coupling'. In: *Computer Methods in Applied Mechanics and Engineering* 316:1157–1178 (2017).

[60] Les Piegle and Wayne Tiller. *The NURBS book*. Springer, 1995. ISBN: 3-540-55069-0.

[61] C.G. Politis, A.I. Ginnis, P.D. Kaklis, K. Belibassakis and C. Feurer. 'An isogeometric BEM for exterior potential-flow problems in the plane'. In: *Conference: Proceedings of the 2009 ACM Symposium on Solid and Physical Modeling* (2009).

[62] X. Qian. 'Full analytical sensitivities in nurbs based isogeometric shape optimization'. In: *Computer Methods in Applied Mechanics and Engineering* 199(29-32):2059–2071 (2010).

[63] S. Ramakrishnan and S.S. Collis. 'Turbulence control simulation using the variational multiscale method'. In: *AIAA Journal* 42(4):745–753 (2004).

[64] *Rhino 6 features*. `https://www.rhino3d.com/6/features`. Last accessed: 03/10/2018.

[65] D. Schillinger, S.J. Hossain and T.J.R. Hughes. 'Reduced bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis.' In: *Computer Methods in Applied Mechanics and Engineering* 277:1–45 (2014).

[66] D. Schillinger, L. Dede, M.A. Scott, J.A. Evans, E Borden M.J. amd Rank and T.J.R. Hughes. 'An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of nurbs, immersed boundary methods, and t-spline cad surfaces.' In: *Computer Methods in Applied Mechanics and Engineering* 249:116–150 (2012).

[67] Y.D. Seo, H.J. Kim and S.K. Youn. 'Shape optimization and its extension to topological design based on isogeometric analysis'. In: *International Journal of Solids and Structures* 47(11-12):1618–1640 (2010).

[68] T.Q. Thai, T. Rabczuk, Y. Bazilevs and G. Meschke. 'A higher-order stress-based gradient-enhanced damage model based on isogeometric analysis'. In: *Computer Methods in Applied Mechanics and Engineering* 304:584–604 (2016).

[69] A. Townsend. *Dibley Marine Design*. `http://www.alatown.com/spline-history-architecture/`. Last accessed: 04/10/2018. 2015.

[70] R Vázquez and A. Buffa. 'Isogeometric analysis for electromagnetic problems'. In: *IEEE Transactions on Magnetics* 46(8):3305–3308 (2010).

[71] A.V. Vuong, C. Giannelli, B Jüttler and B Simeon. 'A hierarchical approach to adaptive local refinement in isogeometric analysis.' In: *Computer Methods in Applied Mechanics and Engineering* 200(49-52):3554–3567 (2011).

[72] W.A. Wall, M.A. Frenzel and C. Cyron. 'Isogeometric structural shape optimization'. In: *Computer methods in applied mechanics and engineering* 197(33-40):2976–2988 (2008).

[73] Y. Zhang, Y. Bazilevs, S. Goswami, C. Bajaj and T.J.R. Hughes. 'Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow'. In: *Computer Methods in Applied Mechanics and Engineering* 196:2943–2959 (2007).

[74] O.C. Zienkiewics, R.L. Taylor and J.Z. Zhu. *Finite element method: its basis and fundamentals*. Elsevier, 2013. ISBN: 978-1-85617-633-0.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

`https://bitbucket.org/amiede/classicthesis/`

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

`http://postcards.miede.de/`

Thank you very much for your feedback and contribution.

Final Version as of 29th October 2018 (classicthesis v4.6).