

SOLVING THE MILD-SLOPE AND HELMHOLTZ EQUATIONS USING THE VIRTUAL ELEMENT METHOD (VEM), DEALING WITH HIGH ORDER ROBIN BOUNDARY CONDITION

R. Dupont^{1,2} and M. Dauphin³, R. Mottier^{4,5,6}

July 30, 2024

¹ GEOSCIENCES-M, Univ Montpellier, CNRS, Montpellier, France.

² IMAG, Univ Montpellier, CNRS, Montpellier, France.

³ Scuola Superiore Meridionale, Napoli, Italy.

⁴ École nationale des ponts et chaussées, France.

⁵ Commissariat à l'énergie atomique et aux énergies alternatives, France.

⁶ Institut national de recherche en sciences et technologies du numérique, France.

*Corresponding authors. E-mail(s): ronan.dupont@umontpellier.fr;
mathias.dauphin-ssm@unina.it; romain.mottier@outlook.com.

Abstract

The numerical solution of the Mild-slope equation (MSE) is crucial in various fields, including coastal engineering, oceanography, and offshore structure design. In this article, we present a novel approach utilizing the Virtual Element Method (VEM) for the numerical solution of the MSE. The VEM offers significant advantages over traditional finite element methods, particularly in handling complex geometries and irregular meshes. We first look at the implementation and validation of the model in the presence of Robin boundary conditions. We then apply the results to the calculation of eigenmodes for the port of Cherbourg.

Keywords. Mild-slope equation, Helmholtz equation, Virtual Element Method, computational fluid dynamics, validation, Finite Element Methods, Numerical Analysis, Complex Geometries, Irregular Meshes, Robin Boundary Condition, Coastal Engineering.

1 Introduction

Nowadays, coastal modeling has become a major challenge in the face of climate change. Coastal-related topics have become very numerous, including ocean modeling (large-scale), port modeling and numerous other topics such as morphodynamics. In this study, we are particularly interested in port modeling through the equation models developed by Helmholtz (1868) and Berkhoff (1972). These two equation models can be used in coastal modeling to calculate wave agitation inside a harbor. The Helmholtz (1868) equation is a very classical equation, which can be used in various fields such as electromagnetics or acoustics. In our study, it is used for flat sea bottoms, while the Berkhoff (1972) equation, also known as the Mild-Slope equation, is used for variable bottoms with a maximum slope of 1/3 (Booij 1983). In this study, we have chosen to solve these equations using the virtual element method (Beirão da Veiga et al. 2014) MATHIAS, tu peux ajouter qqsls autres refs importantes stp (refs.bib). This method has the advantage of i) being a high-order finite element method, which enables wave phenomena to be accurately captured, where simple finite elements have difficulty capturing them RORO, tu peux ajouter des références pour ce que tu disais, que sur des méthodes FEM simple, même en raffinant, on arrivait pas à capturer tous les phénomènes ondulatoires, ii) handle polyhedral meshes as well as non-conforming meshes, enabling simple refinement in certain areas. Although a few studies have already dealt with the Helmholtz (1868) problem in virtual elements (Perugia et al. 2016; Mascotto et al. 2019), none of them has had any concrete application in the coastal sector. In this study, we will first express the modeling of the problem. Next, we will explain the virtual element strategy for approximating this problem. Finally, we'll look at a particular boundary condition, the Robin condition. After validating our model, we will apply it to the calculation of eigenvalues for the port of Cherbourg.

(Cook et al. 2021)

Vous pouvez étoffer un peu, c'est un draft.

2 Model Problem

2.1 Physical Background

Voilà ce que je pense faire ici:

- introduire le setting de la théorie des ondes pour les vagues en milieu portuaire
- introduire le problème spécifique avec le port + commenter les conditions aux bords

In this section, we consider the wave problem described in figure 1.

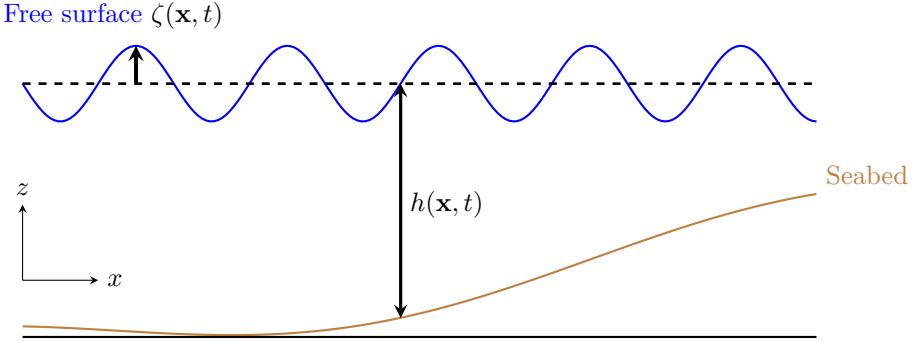


Figure 1: Sketch of a free surface elevation ζ in the (x, z) -plane.

with ζ the free surface defined by $\zeta(\mathbf{x}, t) = \Re \{ \eta(x, y) e^{-i\omega t} \}$, η a complex-valued amplitude of ζ , $\omega = 2\pi/T_0$ the angular frequency, T_0 the wave period and h the depth.

The amplitude η can be split into its incident and reflective part,

$$\eta = \eta_I + \eta_R.$$

We thus have,

$$\eta_I(\mathbf{x}, t) = u_I(\mathbf{x}) e^{-i\omega t} \quad \text{and} \quad \eta_R(\mathbf{x}, t) = u_R(\mathbf{x}) e^{-i\omega t}$$

with the incident wave amplitude defined by,

$$u_I(\mathbf{x}) = u_{\max} e^{-i\mathbf{k}\mathbf{x}} \quad \text{with} \quad \mathbf{k} = k(\cos(\theta), \sin(\theta))^T,$$

with θ the incident wave angle, u_{\max} the maximum wave amplitude.

The amplitude of the reflected wave u_R is obtained by solving the [Helmholtz \(1868\)](#) equation, in the case of a flat bottom,

M.D: Modifications à partir d'ici. Je pense qu'on devrait ranger tout ce qu'il y a avant dans une sous-section "Linear Wave Theory" qui permettrait de définir les variables physiques et leur notations, le setting du problème (port donc domaine borné) avec un commentaire sur les conditions aux bords que l'on impose.

R.D: Je ne pense pas que ça soit nécessaire, on peut le laisser en introduction de la section car ça introduit bien le problème, de plus, il n'y a pas énormément de contenu (moins d'une page). Sinon on peut faire une section "physical background" ou qdc du genre si tu préfères.

2.2 The Helmholtz Model

We first consider a model based on the [Helmholtz \(1868\)](#) equation together with mixed boundary conditions

$$\begin{cases} -\Delta u - k^2 u = 0, & \text{in } \Omega \\ u = g_D, & \text{in } \Gamma_D, \\ \frac{\partial u}{\partial n} + ik u = 0, & \text{in } \Gamma_R. \end{cases} \quad (1)$$

where k is the wave number and $g_D = \alpha u_I \in H^{1/2}(\partial\Omega)$. From a mathematical point of view, no particular assumption is required on k except for establishing estimates. In that case, we assume that k is bounded away from 0, that is, there exists $k_0 > 0$ such that $k > k_0$. From a more practical point of view, the value of k corresponds to the solution of the first order approximation of the dispersion relation from linear theory (Airy 1845),

$$\omega^2 = g k \tanh(kh). \quad (2)$$

Je pense que je vais réécrire toute la partie qui suit avec beaucoup moins de détails. Le problème vient de la condition de Dirichlet qui oblige à écrire la formulation variationnelle avec l'espace $V = \{v \in H^1(\Omega) : v = g_D \text{ sur } \Gamma_D\}$ qui n'est pas un espace vectoriel... Je laisse ça en suspens pour me consacrer à la partie VEM dès lundi !

Before writing the variational formulation for problem (3), we state the following lemma.

Lemma 1 (Lifting). *If $f \in H^{1/2}(\partial\Omega)$, then there exist $\tilde{f} \in H^1(\Omega)$ such that*

$$\tilde{f} = f \quad \text{a.e. on } \partial\Omega$$

R.D: C'est normal le texte en italique ? Yes, tous les environnements de maths type lemme, def, theorem, etc. ont un corps en italique okok, Bah pourtant tes définitions sont pas en italique!

Proof. The result is a direct consequence of the surjectivity of the trace operator $\gamma : H^1(\Omega) \rightarrow H^{1/2}(\partial\Omega)$ [REF Lions-Magenes]. \square

Let us denote by $\widetilde{g_D}$ the lifting of g_D . Letting $v = u - \widetilde{g_D}$, we can rewrite the problem (3) with homogeneous Dirichlet boundary conditions

$$\begin{cases} -\Delta v - k^2 v = F_1, & \text{in } \Omega \\ v = 0, & \text{on } \Gamma_D, \\ \frac{\partial v}{\partial n} + ik v = G_1, & \text{on } \Gamma_R, \end{cases} \quad (3)$$

which weak formulation reads

$$\begin{cases} \text{find } v \in V := H_0^1(\Omega) \text{ such that} \\ a(v, w) = l(w) \quad \forall w \in V, \end{cases} \quad (4)$$

with

$$a(v, w) := \int_{\Omega} \nabla v \cdot \nabla w - k^2 \int_{\Omega} vw + ik \int_{\Gamma_R} vw \quad \forall v, w \in V \quad (5)$$

$$l(w) := \int_{\Omega} F_1 w + \int_{\Gamma_R} G_1 w \quad \forall w \in V \quad (6)$$

For carrying out the analysis of the problem, we equip V with the following weighted norm, such that for all $w \in V$

$$\|w\|_{1,k}^2 = \|\nabla w\|_{0,\Omega}^2 + k^2 \|w\|_{0,\Omega}^2$$

Lemma 2 (Continuity and Garding inequality). *The bilinear form $a : V \times V \rightarrow \mathbb{R}$ satisfies the following properties*

(i) *it is continuous*

(ii) *it satisfies the Garding inequality*

$$\Re(a(w, w)) + 2k^2 \|w\|_{0,\Omega}^2 = \|w\|_{1,k}^2 \quad (7)$$

2.3 The Mild-Slope Model

The amplitude of the reflected wave u_R can also be obtained by solving the Mild-Slope equation (Berkhoff 1972), in the case of a variable bottom,

$$\left\{ \begin{array}{ll} \nabla(C_p C_g \nabla u) + k^2 C_p C_g u = 0, & \text{in } \Omega, \\ u = u_I, & \text{in } \Gamma_{\text{in}}, \\ \frac{\partial u}{\partial n} + ik u = 0, & \text{in } \Gamma_{\text{out}}, \\ u = -\gamma u_I & \text{in } \Gamma_D. \end{array} \right. \quad (8)$$

with

$$C_p = \frac{\omega}{k} \quad \text{and} \quad C_g = \frac{1}{2} C_p \left[1 + kh \frac{1 - \tanh^2(kh)}{\tanh(kh)} \right]. \quad (9)$$

The choice of boundary conditions will be explained in the application section 5.

Remarks:

- In practice, k is obtained simply by using the Guo (2002) approximation.
- Assuming constant depth within the port and $C_g = C_p/2$ (as in shallow water) and noting that $C_p = \omega/k = Cte$, equation (8) can be simplified to yield the Helmholtz (1868) equation.

Formulation varitionnelle à modifier

Now, we consider the Berkhoff (1972) equation (8) and thus the following variational formulation:

$$\begin{cases} \text{find } u \in V = H_0^1(\Omega) \text{ such that} \\ \quad \mathcal{A}(u, v) = 0 \quad \forall v \in V, \end{cases} \quad (10)$$

where,

$$\mathcal{A}(u, v) = \int_{\Omega} \nabla(C_p C_g \nabla u) v + \int_{\Omega} k^2 C_p C_g u v . \quad (11)$$

We build a discrete problem in following form:

$$\begin{cases} \text{find } u_h \in V_h \subset V \text{ such that} \\ \quad \mathcal{A}_h(u_h, v_h) = 0 \quad \forall v_h \in V_h, \end{cases} \quad (12)$$

where $V_h \subset V$ is a finite dimensional space and $\mathcal{A}_h(\cdot, \cdot) : V_h \times V_h \rightarrow \mathbb{R}$ is a discrete bilinear form approximating the continuous form $\mathcal{A}(\cdot, \cdot)$.

We thus have the discrete form:

$$\begin{aligned} \mathcal{A}_h(u_h, v_h) &= \sum_{E \in \Omega_h} \left[\int_E \nabla(C_p C_g \nabla u_h) v_h + \int_E k^2 C_p C_g u_h v_h \right], \\ &\stackrel{1/E \int_E C_p C_g = \mathcal{A}_E}{\approx} \sum_{E \in \Omega_h} \left[\mathcal{A}_E \int_E (\Delta u_h) v_h + \mathcal{B}_E \int_E u_h v_h \right], \\ &\stackrel{1/E \int_E k^2 C_p C_g = \mathcal{B}_E}{=} \sum_{\substack{\text{green} \\ \partial u / \partial n = -iku}}_{E \in \Omega_h} \left[-\mathcal{A}_E \int_E \nabla u_h \nabla v_h + \mathcal{B}_E \int_E u_h v_h - \mathbb{1}_{\Gamma_{\text{out}} \subset E} i \mathcal{A}_E \int_{\Gamma_{\text{out}}} k u_h v_h \right], \\ &= a_h(u_h, v_h) + r_h(u_h, v_h). \end{aligned} \quad (13)$$

with a_h and r_h the discrete forms of a and r : defined in the same way as above, $\mathbb{1}_{\Gamma_{\text{out}} \subset E}$ the indicator function and $u = u + u_D$ and u_D is the lifting of $-\gamma u_I$ or u_I (depending on the border).

3 Solving Equations Using the Virtual Element Method

In this section, we expose the building of the VEM. The method follows the same

3.1 Mesh Decomposition

As in FEM, the first step for building the method is the discretization of the computational domain Ω . But, contrary to the FEM, the VEM can handle

polygonal meshes made out of a wide variety of element shapes. This geometrical flexibility allows us to consider non-conforming elements with hanging nodes as well as non-convex elements.

Definition 3 (Polygonal Mesh). A polygonal mesh $\Omega_h = (\mathcal{T}_h, \mathcal{E}_h, \mathcal{V}_h)$ consists of

- (i) a collection of polygonal elements \mathcal{T}_h ;
- (ii) a collection of edges $\mathcal{E}_h = \mathcal{E}_h^i \cup \mathcal{E}_h^b$, where \mathcal{E}_h^i corresponds to the internal edges and \mathcal{E}_h^b the boundary edges;
- (iii) a collection of vertices \mathcal{V}_h .

3.2 Polynomial Projections

One of the key ingredient for building the VEM are the projection operators onto local polynomial spaces. They are essential to the virtual element discretization in order to

Definition 4 (L^2 -Projection). Let $E \in \mathcal{T}_h$ and $l \geq 0$ an integer. We define the L^2 -projection $\Pi_E^{0,l} : L^2(E) \rightarrow \mathbb{P}^l(E)$ such that, for any $v \in L^2(E)$,

$$\int_E \Pi_E^{0,l} v \ p = \int_E v \ p \quad \forall p \in \mathbb{P}^l(E) \quad (14)$$

3.3 Virtual Space

Introduire un peu de VEM

1. SPACE:

Let Ω_h be a simple polygonal mesh on Ω . This can be any decomposition of Ω in non overlapping polygons E with straight faces. The space V_h will be defined element-wise, by introducing

- local spaces $V_{h|E}$;
- the associated local degrees of freedom.

For all $E \in \Omega_h$:

$$V_{h|E} = \left\{ v_h \in H_0^1(\Omega) \cap C^0(\partial E) : v_h|_{\partial E} \in \mathbb{P}_k(E), \Delta v_h \in \mathbb{P}_k(E) \right. \\ \left. (\Pi_k^\nabla v_h - v_h, p)_{0,E} = 0, \forall p \in \mathbb{P}_k(E)/\mathbb{P}_{k-2}(E) \right\} \quad (15)$$

where $\mathbb{P}_k(E)/\mathbb{P}_{k-2}(E)$ is the subspace of $\mathbb{P}_k(E)$ of the polynomials that are orthogonal in the sense of L^2 to $\mathbb{P}_{k-2}(E)$, or, alternatively, the polynomials of degree $k-1$ and k .

- the functions $V_{h|E}$ are continuous (and known) on ∂E ;

- the functions $V_{h|E}$ are unknown on E !

1. Mesh Decomposition: We consider a polytopal decomposition $\{T_h\}_h$ of the domain Ω which is regular—that is, there exists $\rho \in (0, 1)$, independent of h , such that every element $E \in T_h$ is star-shaped with respect to a ball of radius $\geq \rho h_E$, with h_E the diameter of E .

2. Local Projections: We denote by $\Pi_k^{\nabla,E} : H^1(E) \rightarrow \mathbb{P}_k(E)$ and $\Pi_k^{0,E} : L^2(E) \rightarrow \mathbb{P}_k(E)$ the usual lo elliptic projection and local L^2 -projection respectively onto the space of polynomials of degree a most k .

3. Virtual Space: We define the local virtual space by

$$V_h^E = \left\{ v_h \in H_0^1(\Omega) \cap C^0(\partial E) : v_h|_{\partial E} \in \mathbb{P}_k(E), \Delta v_h \in \mathbb{P}_k(E) \right. \\ \left. (\Pi_k^{\nabla} v_h - v_h, p)_{0,E} = 0, \forall p \in \mathbb{P}_k(E)/\mathbb{P}_{k-2}(E) \right\}$$

where $\mathbb{P}_k(E)/\mathbb{P}_{k-2}(E)$ is the subspace of $\mathbb{P}_k(E)$ of the polynomials that are orthogonal in the sense of L^2 to $\mathbb{P}_{k-2}(E)$, or, alternatively, the polynomials of degree $k-1$ and k .

3. DDL:

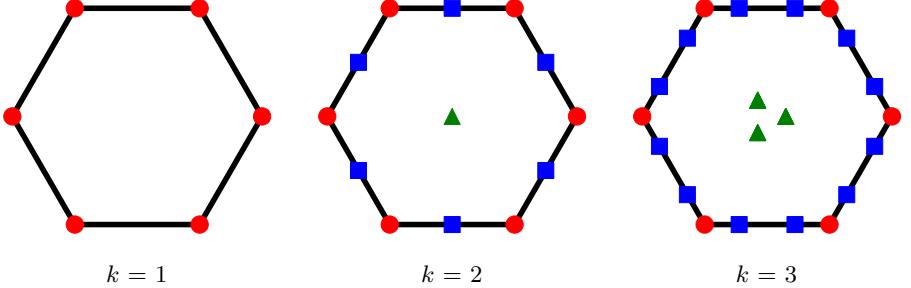


Figure 2: 2D element with ● : Summits dofs, ■ : Edges dofs, ▲ : Inner dofs.

4. Base:

$$m_{\alpha_1, \alpha_2} = \left(\frac{x - x_D}{h_D} \right)^{\alpha_1} \cdot \left(\frac{y - y_D}{h_D} \right)^{\alpha_2}$$

3.4 Discrete Problem

We decompose this subsection into two parts, one expressing the variational formulation for the Helmholtz (1868) equation (3) and another for that of the Mild-Slope equation (8).

3.4.1 Discrete Helmholtz Problem

3.4.2 Discrete Mild-Slope Problem

Ajouter la partie FV avec les projecteurs etc

3.5 Analysis

M.D: éventuellement si j'arrive à dériver les estimateurs

4 Guidelines for implementing and validating the virtual element method with a Robin boundary condition

In this section, we outline the implementation of the virtual element method with a Robin's boundary condition and its validation.

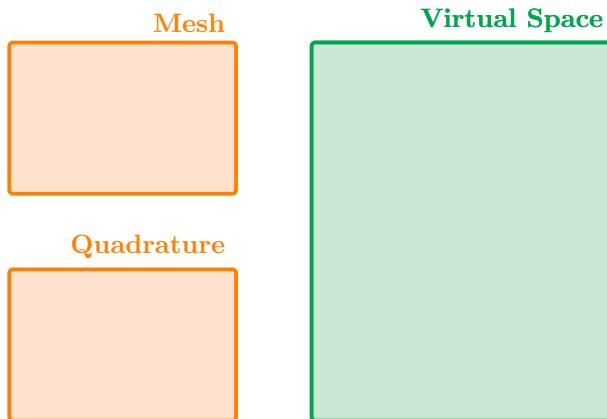


Figure 3: A guideline for implementing the Virtual Element Method

4.1 Virtual Element Method Implementation

In this section, we take a quick look at the implementation of the virtual element method. This can be completed using the ([Beirão da Veiga et al. 2014](#); [Sutton 2017](#); [Herrera et al. 2023](#)) articles.

4.1.1 Some Notations

In order to introduce a few definitions for implementation purposes, we define the following notations: l the maximum polynomial order, $n_l := \dim \mathcal{P}_l = \frac{(l+1)(l+2)}{2}$, the dimension of the polynomial space; N^{dof} , the number of degrees

of freedom on the discretized problem; $\varphi_{i=1,\dots,N^{\text{dof}}}$, the basis functions, $\varphi_i \in V_l(E)$; $m_{i=1,\dots,n_l}$, the scaled monomial of degree defined in **REF TO PARTIE MATHIAS**; P_0 , a projection operator onto constant; $\text{dof}_i(\varphi_j) = \delta_{ij}$; $(\cdot)_{0,E}$ the integral on the E element.

4.1.2 Calculating the stiffness matrix

Here, we recall the definition of \mathbf{B} , \mathbf{D} , \mathbf{G} , $\boldsymbol{\Pi}_*^\nabla$, $\boldsymbol{\Pi}^\nabla$ matrices, enabling us to calculate the stiffness matrix locally on an element E . Details of the calculations and implementation can be found in the 3 articles cited above (section 4.1).

Definition of \mathbf{B}

We denote \mathbf{B} the $n_l \times N^{\text{dof}}$ matrix given by:

$$\mathbf{B} := \begin{bmatrix} P_0\varphi_1 & \cdots & P_0\varphi_{N^{\text{dof}}} \\ (\nabla m_2, \nabla \varphi_1)_{0,E} & \cdots & (\nabla m_2, \nabla \varphi_{N^{\text{dof}}})_{0,E} \\ \vdots & \ddots & \vdots \\ (\nabla m_{n_l}, \nabla \varphi_1)_{0,E} & \cdots & (\nabla m_{n_l}, \nabla \varphi_{N^{\text{dof}}})_{0,E} \end{bmatrix}. \quad (16)$$

Definition of \mathbf{D}

We denote \mathbf{D} the $N^{\text{dof}} \times n_l$ matrix given by:

$$\mathbf{D}_{i\alpha} := \text{dof}_i(m_\alpha), \quad i = 1, \dots, N^{\text{dof}}, \quad \alpha = 1, \dots, n_l,$$

that is,

$$\mathbf{D} = \begin{bmatrix} \text{dof}_1(m_1) & \text{dof}_1(m_2) & \cdots & \text{dof}_1(m_{n_l}) \\ \text{dof}_2(m_1) & \text{dof}_2(m_2) & \cdots & \text{dof}_2(m_{n_l}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{dof}_{N^{\text{dof}}}(m_1) & \text{dof}_{N^{\text{dof}}}(m_2) & \cdots & \text{dof}_{N^{\text{dof}}}(m_{n_l}) \end{bmatrix}. \quad (17)$$

Definition of \mathbf{G}

We denote \mathbf{G} the $n_l \times n_l$ matrix given by:

$$\mathbf{G} := \begin{bmatrix} P_0 m_1 & P_0 m_2 & \cdots & P_0 m_{n_l} \\ 0 & (\nabla m_2, \nabla m_2)_{0,E} & \cdots & (\nabla m_2, \nabla m_{n_l})_{0,E} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (\nabla m_{n_l}, \nabla m_2)_{0,E} & \cdots & (\nabla m_{n_l}, \nabla m_{n_l})_{0,E} \end{bmatrix}. \quad (18)$$

Definition of Π_*^∇

The $n_l \times N^{\text{dof}}$ matrix representation Π_*^∇ of the operator Π^∇ acting from $V_l(E)$ to $P_l(E)$ in the basis $\mathcal{M}_l(E)$ is given by:

$$\Pi_*^\nabla = \mathbf{G}^{-1} \mathbf{B}. \quad (19)$$

Definition of Π^∇

The $N^{\text{dof}} \times N^{\text{dof}}$ matrix representation Π^∇ of the operator $\Pi^\nabla : V_l(E) \rightarrow V_l(E)$ in the canonical basis is given by:

$$\Pi^\nabla = \mathbf{D} \mathbf{G}^{-1} \mathbf{B} = \mathbf{D} \Pi_*^\nabla. \quad (20)$$

Definition of \mathbf{K}_E^h

The $N^{\text{dof}} \times N^{\text{dof}}$ local stiffness matrix is given by:

$$\mathbf{K}_E^h = (\Pi_*^\nabla)^T \tilde{\mathbf{G}} (\Pi_*^\nabla) + (\mathbf{I} - \Pi^\nabla)^T (\mathbf{I} - \Pi^\nabla), \quad (21)$$

with $\tilde{\mathbf{G}}$ is the matrix that coincides with \mathbf{G} (see equation (18)) except for the first row which is set to zero.

4.1.3 Calculating the mass matrix

Here, we recall the definition of \mathbf{H} , \mathbf{C} , Π^0 matrices, enabling us to calculate the mass matrix locally on an element E . Details of the calculations and implementation can be found in the 3 articles cited above (section 4.1).

Definition of \mathbf{H}

We denote \mathbf{H} the $n_l \times n_l$ matrix given by:

$$\mathbf{H} := (m_\alpha, m_\beta)_{0,E} = \begin{bmatrix} (m_1, m_1)_{0,E} & \cdots & (m_1, m_{n_l})_{0,E} \\ \vdots & \ddots & \vdots \\ (m_{n_l}, m_2)_{0,E} & \cdots & (m_{n_l}, m_{n_l})_{0,E} \end{bmatrix}, \quad (22)$$

for $\alpha = 1, \dots, n_l$ and $\beta = 1, \dots, n_l$.

Definition of \mathbf{C}

We denote \mathbf{C} the $n_k \times N^{\text{dof}}$ matrix given by:

$$\mathbf{C}_{\alpha i} := (m_\alpha, \varphi_i)_{0,E} = \begin{cases} (m_\alpha, \varphi_i)_{0,E} & \text{if } 1 \leq \alpha \leq n_{l-2} \\ (m_\alpha, \Pi^\nabla \varphi_i)_{0,E} & \text{if } n_{l-2} + 1 \leq \alpha \leq n_l \end{cases} \quad (23)$$

for $i = 1, \dots, N^{\text{dof}}$ and $\alpha = 1, \dots, n_l$.

Definition of Π^0

We denote Π^0 the $N^{\text{dof}} \times N^{\text{dof}}$ matrix given by:

$$\Pi^0 = \mathbf{D}\mathbf{H}^{-1}\mathbf{G}. \quad (24)$$

Definition of \mathbf{M}_E^h

The $N^{\text{dof}} \times N^{\text{dof}}$ local mass matrix is given by:

$$\mathbf{M}_E^h = \mathbf{C}^T \mathbf{H}^{-1} \mathbf{C} + |E| (\mathbf{I} - \Pi^0)^T (\mathbf{I} - \Pi^0). \quad (25)$$

4.1.4 Example of implementation

Below in algorithm 1, we propose a pseudo-code implementation of virtual element method taking into account Dirichlet and Robin boundary conditions.

Algorithm 1 Pseudocode of the Virtual Element Implementation

```

1: for  $E \in \mathcal{T}_h$  do
2:   Compute  $B, D, G, C, H$ 
3:   Compute  $\Pi_*^\nabla, \Pi_*^\nabla, \Pi_*^0, \Pi^0$ 
4:   Compute the local stiffness matrix:  $\mathbf{K}_E^h = (\Pi_*^\nabla)^T \tilde{\mathbf{G}} (\Pi_*^\nabla) +$ 
    $(\mathbf{I} - \Pi^\nabla)^T (\mathbf{I} - \Pi^\nabla)$ 
5:   Compute the local mass matrix:  $\mathbf{M}_E^h = \mathbf{C}^T \mathbf{H}^{-1} \mathbf{C} +$ 
    $|E| (\mathbf{I} - \Pi^0)^T (\mathbf{I} - \Pi^0)$ 
6:    $A_h \leftarrow A_h + (-\mathbf{K}_E^h + k^2 \mathbf{M}_E^h)$             $\triangleright$  Adds global contributions
7: end for
8: for  $e \in \mathcal{E}_h^b$  do                                 $\triangleright$  Browse the DDL on Boundary
9:   if  $e \in \mathcal{E}_h^{b,r}$  then                       $\triangleright$  See if it's about Robin's condition
10:    Compute  $\mathbf{R}_e^h$                              $\triangleright$  See section 4.2
11:     $A_h \leftarrow A_h + \mathbf{R}_e^h$                    $\triangleright$  Adds global contributions
12:   else if  $e \in \mathcal{E}_h^{b,d}$  then
13:      $A_h \leftarrow 1$   $\triangleright$  Set the line to 0 and add a 1 to the correct DDL position
14:      $b_h \leftarrow$  value of DBC                 $\triangleright$  Set the value of the DC on this DDL
15:   end if
16: end for
17: Solve  $A_h u_h = b_h$ 

```

4.2 Robin Boundary Condition

In this section, we focus on calculating Robin's term $r_h(u_h, v_h)$ in our variational formulation. Unlike the mass matrix \mathbf{M} and the stiffness matrix \mathbf{K} , which are calculated using the virtual element formalism, the robin matrix \mathbf{R}

is calculated in a manner analogous to conventional high-order finite elements, with the difference that the degrees of freedom are not placed in the same locations.

We can express the global matrix \mathbf{R} associated to the formulation r_h in a basis of classical shape function, with thus $\mathbf{R} = \left(\int_{\Gamma_{\text{out}}} \alpha(x, y) \Phi_j(x, y), \Phi_i(x, y) \right)_{i,j}$ with α any function of \mathbb{R}^2 .

The Γ_{out} boundary can be decomposed into a sum of 1D elements that can be characterized by the segment $[\xi_0, \xi_0 + \lambda]$ between two points V_0 and V_1 . These elements are segments joining 2 consecutive points of the edge. The Φ_i basis function attached to the i vertex of the edge, restricted to the edge element, is a \mathbb{P}_k function of the edge. These characteristic 1D elements are shown in the figure 4 with the degrees of freedom corresponding to the Gauss-Lobatto quadrature points on $[0, \lambda]$. Consequently, the higher the order, the more points there will be on the segment.

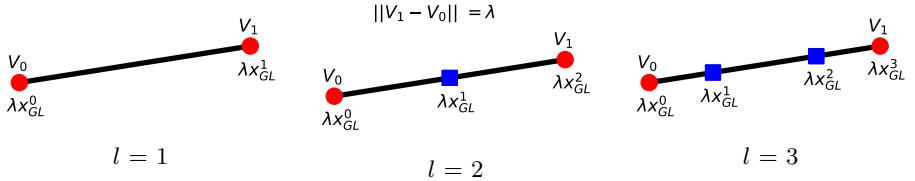


Figure 4: 1D element $[\xi_0, \xi_0 + \lambda]$ representation for different orders l , with ● : Summits dofs, ■ : Edges dofs.

In this way, we can express the segment's local matrix:

$$\mathbf{R}_e^h = \left(\int_0^\lambda \alpha_*(\xi_0 + \xi) \varphi_i(\xi) \varphi_j(\xi) d\xi \right)_{0 \leq i, j \leq l} \quad (26)$$

with φ_i, φ_j polynomial test functions of order l , $\alpha_*(\xi_0 + \xi) = \alpha(V_0 + \xi \vec{t})$ the 1D restriction of α on Γ_{out} and \vec{t} the tangential unit vector (from V_0 to V_1).

We can easily deduce the explicit form of $\varphi_{i=0, \dots, l}$ because we have $\forall i, j \in \llbracket 0, l \rrbracket$,

$$\varphi_i(\lambda x_{GL}^j) = \delta_j^i, \quad (27)$$

with x_{GL}^j the $j - th$ Gauss-Lobatto quadrature point on $[0, 1]$ (see figure 4). We can therefore deduce from the Lagrange polynomials:

$$\begin{aligned} \varphi_i(\xi) &= \sum_{j=0}^l \delta_j^i \left(\prod_{m=0, m \neq j}^l \frac{\xi - \lambda x_{GL}^m}{\lambda x_{GL}^j - \lambda x_{GL}^m} \right), \\ &= \frac{1}{\lambda^l} \prod_{m=0, m \neq i}^l \frac{\xi - \lambda x_{GL}^m}{x_{GL}^i - x_{GL}^m}. \end{aligned}$$

For example, for $l = 1$, $\varphi_0(\xi) = \frac{\lambda - \xi}{\lambda}$ and $\varphi_1(\xi) = \frac{\xi}{\lambda}$.

Now we can compute the local matrix \mathbf{R}_e^h of size $(l+1, l+1)$ equation (26). All we need to do is calculate the integrations using a quadrature method, such as Gauss-Lobatto's, which was introduced in the section on virtual elements, section 3.

Next, \mathbf{R} is assembled in the same way as conventional finite element assemblies. Except that here, instead of traversing all the dofs of all the elements in the mesh, we only traverse the dofs of the segments on the Γ_{out} boundary.

The procedure is summarized in the algorithm 1.

Remarks:

- In the case where α is a constant function (which is the case for the Helmholtz equation), we can simplify the equation (26) using a change of variable to obtain the following local matrix equation (28) (see calculation details in appendix C).

$$\begin{aligned} \mathbf{R}_e^h &= \left(\int_0^\lambda \alpha_*(\xi_0 + \xi) \varphi_j(\xi) \varphi_i(\xi) d\xi \right)_{0 \leq i, j \leq l}, \\ &= \alpha \lambda \left(\int_0^1 \tilde{\varphi}_j(\xi) \tilde{\varphi}_i(\xi) d\xi \right)_{0 \leq i, j \leq l}, \end{aligned} \quad (28)$$

with $\tilde{\varphi}_i$ the polynomials for a unit element $[\xi_0, \xi_0 + 1]$.

So all we need to do is evaluate the integral equation (28) once to obtain all the local matrices of the boundary segments. Moreover, this integral deals with a polynomial of degree $2l$, which can be evaluated exactly using a quadrature method with $l + 2$ Gauss-Lobatto points.

- In the case where there is a second member in Robin's term, the approach is similar. We express the second member matrix analogously to the 26 matrix.

$$\mathbf{L} = \int_{\partial\Omega} \beta(x, y) \Phi_i(x, y). \quad (29)$$

with β any function of \mathbb{R}^2 . Then we express the local matrix always on the segments characterized by $[\xi_0, \xi_0 + \delta]$,

$$\mathbf{L}_e^h = \left(\int_0^\lambda \beta_*(\xi_0 + \xi) \varphi_i(\xi) d\xi \right)_{1 \leq i \leq l+1}, \quad (30)$$

with φ_i a polynomial test function of order l , $\beta_*(\xi_0 + \xi) = \beta(V_0 + \xi \vec{t})$ the 1D restriction of β on Γ_{out} and \vec{t} the tangential unit vector (from V_0 to V_1).

Here, we calculate and assemble this matrix as before. Moreover, we can always simplify the calculation of this matrix if β is a constant function, then:

$$\mathbf{L}_e^h = \lambda \beta \left(\int_0^1 \tilde{\varphi}_i(\xi) d\xi \right)_{1 \leq i \leq l+1}. \quad (31)$$

4.3 Numerical Validation

In this section, we check the validity of our model. Thus, we compare our model with a manufactured analytical solution (see appendix A). We therefore consider the [Helmholtz \(1868\)](#) equation with mixed Dirichlet and Robin boundary conditions, below equation (32).

$$\begin{cases} \Delta u + k^2 u = f(x, y) & , \quad \text{in } \Omega, \\ u = u_{\text{exact}} & , \quad \text{on } \Gamma_2 \cup \Gamma_3 \cup \Gamma_4, \\ \frac{\partial u}{\partial n} + i k u = g(x, y) & , \quad \text{on } \Gamma_1, \end{cases} \quad \begin{array}{c} \Gamma_3 \\ \square \\ \Omega \\ \Gamma_4 \\ \Gamma_2 \\ \Gamma_1 \end{array} \quad (32)$$

For this analytical case, we take the geometry of a unit square and link it with regular triangles, irregular triangles, irregular quadrilaterals and polygons. To generate these meshes, we use the Gmsh ([Geuzaine et al. 2009](#)) and PyPoly-Mesher ([Abedi-Shahri 2024; Talischi et al. 2012](#)). We perform calculations from order 1 to order 5 on maximum cell diameters h from 0.05 to 0.7 m. We then compute the L^2 error for each calculation. The results are shown in figure 5.

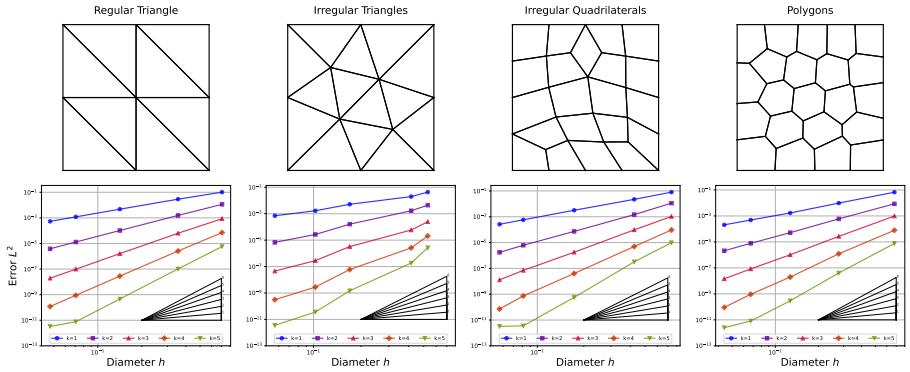


Figure 5: Convergence of order $\mathcal{O}(h^{k+1})$.

We find the expected superconvergence of order $\mathcal{O}(h^{k+1})$.

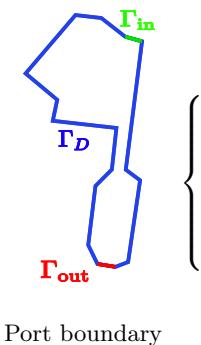
Remarks: For a validation with the Mild-Slope equation, the order would have been less good, given the approximation we have made per cell.

5 Application and Discussion

In this section, we apply the solution of the Helmholtz and Mild-Slope equations to a coastal engineering problem. We take the case of the port of Cherbourg in France and calculate the associated wave fields under certain conditions. First, we select our study site, as shown in figure 6 (left). Next, we



Port location



Port boundary

The Helmholtz or Mild-Slope equation:

$$\left\{ \begin{array}{ll} \nabla(C_p C_g \nabla u) + k^2 C_p C_g u = 0, & \text{in } \Omega, \\ u = u_I, & \text{in } \Gamma_{\text{in}}, \\ \frac{\partial u}{\partial n} + ik u = 0, & \text{in } \Gamma_{\text{out}}, \\ u = -\gamma u_I, & \text{in } \Gamma_D. \end{array} \right.$$

Figure 6: Configuration of our study of the port of Cherbourg

break down the contour into 3 different boundaries (figure 6 (center)): Γ_{in} the harbour entrance, Γ_{out} the harbour exit and Γ_D the port walls. Finally, we assign the correct boundary condition to these edges (figure 6 (right)).

The Γ_{in} boundary condition is modeled by an inhomogeneous Dirichlet condition taking the incident field as argument. The Γ_{out} boundary condition is modeled by a Robin condition allowing the wave to exit without disturbing other wave fields. More information on this condition in Appendix B. The Γ_D boundary condition is modeled by an inhomogeneous Dirichlet condition with a reflection coefficient γ . First, we'll look at the importance of this reflection coefficient in the section 5.1. Then, we will compare the results obtained using the Helmholtz equation and the Mild-Slope equation, in section 5.2. Finally, we will compare the results with different orders of the virtual element method, in section 5.3.

5.1 Sensitivity of the γ reflection coefficient

In this section, we look at the influence of the harbor wall reflection coefficient γ on wave fields. We compare reflected and total wave fields for two different reflection coefficients, $\gamma = 1$ (figure 7 (top)) and $\gamma = 0.5$ (figure 7 (bottom)). For this study, we generate an incident wave field entering the harbour at 280° with a maximum amplitude $u_{\max} = 1$ m and a wave period $T_0 = 8$ s. This incident field can be seen in figure 7 (left). The results of this study are shown in figure 7 with i) on the left, the incident field ii) in the middle, the reflected field (solution of the Helmholtz equation) iii) on the right, the total field.

The results in figure 7 show that by halving the reflection coefficient γ , the reflected wave field is also halved. This results in a totally different total wave field (incident + reflected), so that the harbour's eigenmodes are no longer located in exactly the same places for the two configurations. It is therefore very important to calibrate this reflection condition correctly.

Solving the Helmholtz problem with $\gamma = 1$

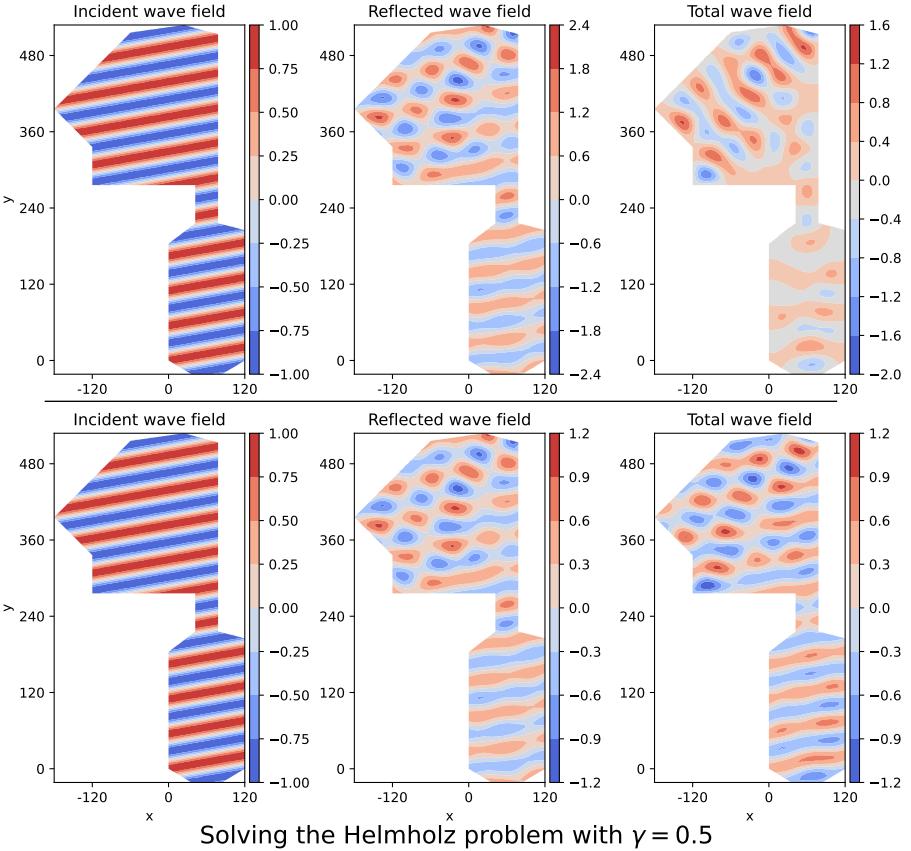


Figure 7: Comparison of wave fields for the two reflection coefficients $\gamma = 1$ (top) and $\gamma = 0.5$ (bottom). Problem condition: $\alpha = 280^\circ$, $u_{\max} = 1$ m and $T_0 = 8$ s.

5.2 Bottom sensitivity between Helmholtz and Mild-Slope

In this section, we look at the influence of the sea bottom on wave fields. We compare a simulation with a flat bottom at a depth of 5 m (figure 8 top left) using the Helmholtz model, with a linear bottom (figure 8 bottom left) using the Mild-Slope model. For this study, we generate an incident wave field entering the harbour at 280° with a maximum amplitude $u_{\max} = 2$ m and a wave period $T_0 = 8$ s. To make the modelling more realistic, Munk (1949) breaking wave criterion is added. This decreases wave amplitude linearly with depth. This incident field can be seen in figure 8. The results of this study are shown in figure 8 from left to right: i) the depth ii) the incident field iii) the reflected field iv) the total field.

The results in figure 8 show that the lack of depth limits the formation of eigenmodes. In fact, in the flat-bottom simulation (top), eigenmodes are

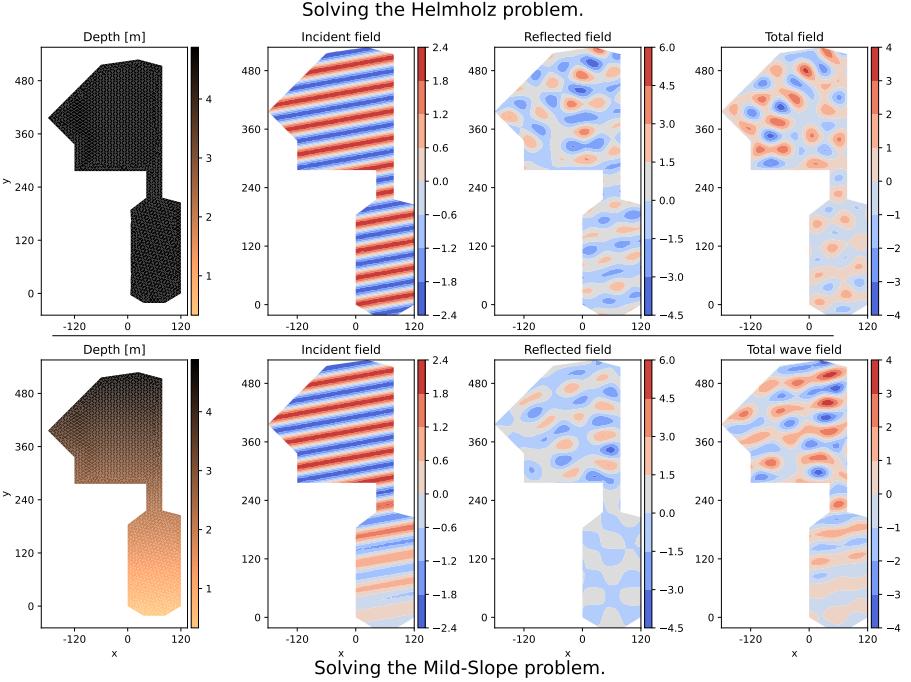


Figure 8: Comparison of wave fields for two different sea bottom: a flat bottom (top) and a linear bottom (bottom). Problem condition: $\alpha = 280^\circ$, $u_{\max} = 2$ m and $T_0 = 8$ s.

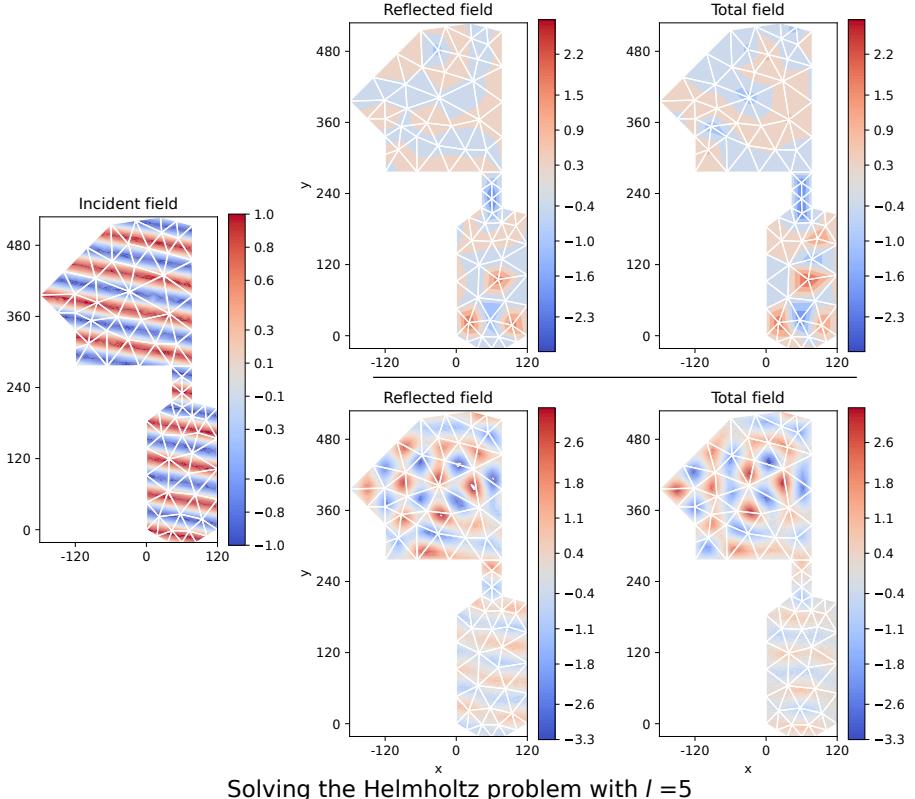
formed in the upper and lower parts of the harbor; whereas in the linear-bottom simulation (bottom), eigenmodes are no longer formed where there is almost no water: in the lower part of the harbor.

5.3 Sensitivity to order of resolution

In this section, we look at the influence of the order of solution of the virtual element method on the solution of the Helmholtz problem. We compare the reflected and total wave fields for two orders of resolution with fairly coarse mesh (figure 9), order 1 (figure 9 top) with 81 degrees of freedom and order 5 (figure 9 bottom) with 881 degrees of freedom. For this study, we generate an incident wave field entering the harbour at 250° with a maximum amplitude $u_{\max} = 1$ m and a wave period $T_0 = 8$ s. The results of this study are shown in figure 9 with i) on the left, the incident field ii) in the middle, the reflected field (solution of the Helmholtz equation) iii) on the right, the total field.

The results in figure 9 show the importance of a high-order solution method for this kind of problem. Indeed, we note that with order 1, it's very difficult to capture the port's eigenmodes, whereas with order 5, the port's eigenmodes are distinguishable.

Solving the Helmholtz problem with $l = 1$



Solving the Helmholtz problem with $l = 5$

Figure 9: Comparison of wave fields for the two order of resolution $l = 1$ (top) and $l = 5$ (bottom). Problem condition: $\alpha = 250^\circ$, $u_{\max} = 1$ m and $T_0 = 8$ s.

6 Conclusion

In this article, we have addressed a coastal engineering problem using the virtual element formalism. This study has highlighted a formalism that is more precise than traditional formalisms, and above all, can handle complex meshes, enabling eigenmodes to be targeted on a given geometry. Thanks to a description of the implementation of the virtual elements, we have been able to provide a guide line for dealing with these wave problems. In addition, the implementation of a high-order Robin condition is something that is rarely mentioned in the literature. The results of the application on the port of Cherbourg showed that it is essential to choose the right parameters, such as the reflection coefficient of the walls and the order of calculation of the virtual element method.

7 Declarations

7.1 Availability of data and material

All data, models, and code generated or used during the study are available on request.

7.2 Conflict of interest

The authors declare that they have no conflict of interest.

7.3 Acknowledgements

This work was conducted as part as M. Dupont's PhD studies which is funded by the CNRS with the MITI grant. We gratefully acknowledge funding from CNRS, OPTIBEACH projects and FEDER Europe. We would also like to thank Pr. Bouchette Frédéric for his ideas.

Appendix

A Manufactured analytical solution

The solution to the equation 32 problem has been produced using the following functions:

$$\begin{aligned} u_{\text{exact}}(x, y) &= (x + y) \cdot (1 + i) + \exp(x^2 + i y^2), \\ f(x, y) &= -((2x)^2 + (2i y)^2 + 2(1 + i)) \cdot \exp(x^2 + i y^2) + k^2 \cdot u_{\text{exact}}(x, y), \\ g(x, y) &= (1 + i) + (2i y) \cdot \exp(x^2 + i y^2) + i k \cdot u_{\text{exact}}(x, y). \end{aligned} \tag{A1}$$

This produces the complex analytical solution that can be seen below in figure A1.

B Interest of a Robin condition

To show the interest of a Robin boundary condition in our problem, we look at the problem represented by figure B1. In this problem, we want to calculate the amplitude of reflected waves around an island. Thus, we have an incident wave arriving at 0° with an amplitude of 2 m and a period of 20 s. This wave is reflected on the boundary island Γ_D . On the other hand, the wave must be able to leave the domain freely via the boundary Γ_{inf} .

The reflected wave is calculated by the following Helmholtz (1868) equation and the wave leaving condition at infinity Γ_{inf} will be studied for a Robin (left) and zero Neumann (right) condition.

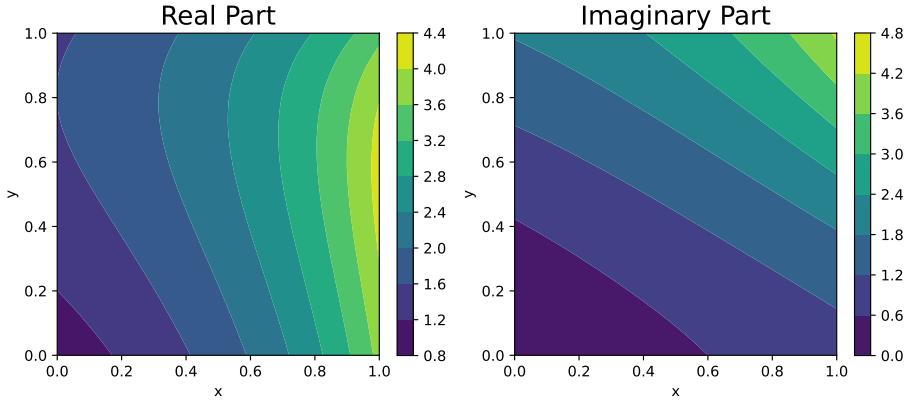


Figure A1: Real and Imaginary part of u_{exact} .

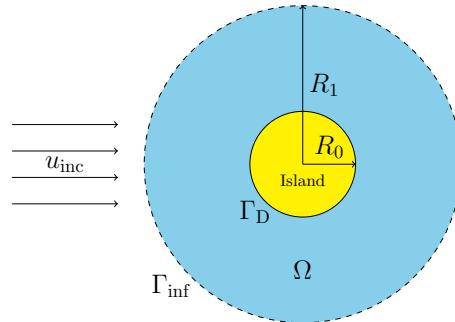
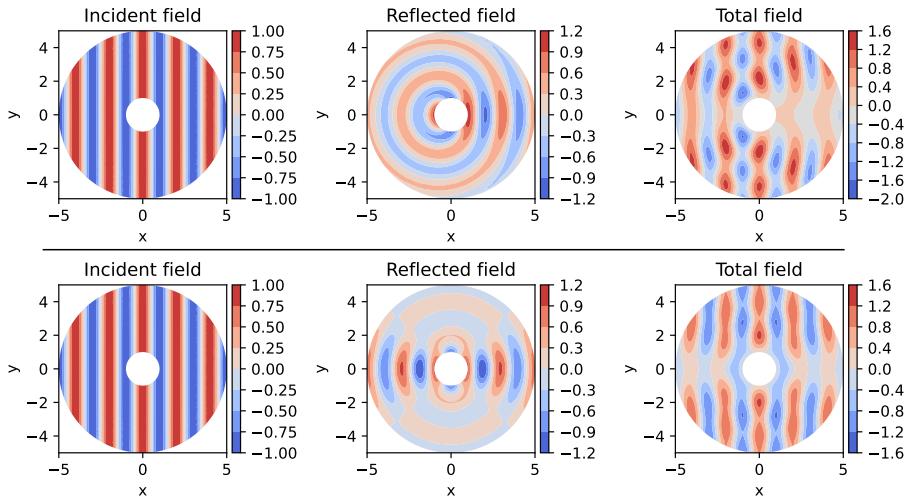


Figure B1: Sketch of the island reflection problem.

$$\left\{ \begin{array}{l} \Delta u + k^2 u = 0 \quad , \quad \text{in } \Omega , \\ u = -u_{\text{inc}} \quad , \quad \text{on } \Gamma_D , \\ \frac{\partial u}{\partial n} + i k u = 0 \quad , \quad \text{on } \Gamma_{\text{Inf}} . \end{array} \right. \quad \text{or} \quad \left\{ \begin{array}{l} \Delta u + k^2 u = 0 \quad , \quad \text{in } \Omega , \\ u = -u_{\text{inc}} \quad , \quad \text{on } \Gamma_D , \\ \frac{\partial u}{\partial n} = 0 \quad , \quad \text{on } \Gamma_{\text{Inf}} . \end{array} \right.$$

The results of this study are shown in figure B2. In this figure, it's clear that the wave reflected under Robin's condition (top) can leave freely, while the wave reflected under Neumann's condition (bottom) seems to be disturbed under this condition.

Solving the Helmholtz problem with a Robin condition on Γ_{inf}



Solving the Helmholtz problem with a Neuman condition on Γ_{inf}

Figure B2: Comparison of the results obtained on the island problem by solving the Helmholtz equation with a Robin condition (top) and a zero Neumann condition (bottom).

C Calculation details of equation (28)

$$\begin{aligned}
\mathbf{R}_e^h &= \left(\int_0^\lambda \alpha_*(\xi_0 + \xi) \varphi_i(\xi) \varphi_j(\xi) d\xi \right)_{0 \leq i, j \leq l}, \\
&\stackrel{\alpha_* = cte}{=} \left(\frac{\alpha}{\lambda^{2k}} \int_0^\lambda \left[\prod_{m=0, m \neq j}^l \frac{\xi - \lambda x_{\text{GL}}^m}{x_{\text{GL}}^i - x_{\text{GL}}^m} \prod_{m=0, m \neq j}^l \frac{\xi - \lambda x_{\text{GL}}^m}{x_{\text{GL}}^j - x_{\text{GL}}^m} \right] d\xi \right)_{0 \leq i, j \leq l}, \\
&\stackrel{\xi = \lambda \xi' \quad d\xi = \lambda d\xi'}{=} \left(\frac{\alpha}{\lambda^{2k}} \int_0^1 \left[\prod_{m=0, m \neq j}^l \frac{\lambda \xi' - \lambda x_{\text{GL}}^m}{x_{\text{GL}}^i - x_{\text{GL}}^m} \prod_{m=0, m \neq j}^l \frac{\lambda \xi' - \lambda x_{\text{GL}}^m}{x_{\text{GL}}^j - x_{\text{GL}}^m} \right] \lambda d\xi' \right)_{0 \leq i, j \leq l}, \\
&= \left(\frac{\alpha}{\lambda^{2k}} \int_0^1 \left[\prod_{m=0, m \neq j}^l \lambda \frac{\xi' - x_{\text{GL}}^m}{x_{\text{GL}}^i - x_{\text{GL}}^m} \prod_{m=0, m \neq j}^l \lambda \frac{\xi' - x_{\text{GL}}^m}{x_{\text{GL}}^j - x_{\text{GL}}^m} \right] \lambda d\xi' \right)_{0 \leq i, j \leq l}, \\
&= \left(\alpha \lambda \int_0^1 \left[\prod_{m=0, m \neq j}^l \frac{\xi' - x_{\text{GL}}^m}{x_{\text{GL}}^i - x_{\text{GL}}^m} \prod_{m=0, m \neq j}^l \frac{\xi' - x_{\text{GL}}^m}{x_{\text{GL}}^j - x_{\text{GL}}^m} \right] d\xi \right)_{0 \leq i, j \leq l}, \\
&= \alpha \lambda \left(\int_0^1 \tilde{\varphi}_j(\xi) \tilde{\varphi}_i(\xi) d\xi \right)_{0 \leq i, j \leq l}.
\end{aligned} \tag{C1}$$

References

- Abedi-Shahri, Seyed Sadjad (2024). *pyPolyMesher: Generation of Polygonal Mesh*. DOI: [10.5281/ZENODO.12794558](https://doi.org/10.5281/ZENODO.12794558).
- Airy, George Biddell (1845). *Tides and waves*. B. Fellowes.
- Beirão da Veiga, L., F. Brezzi, L. D. Marini, and A. Russo (2014). “The Hitch-hiker’s Guide to the Virtual Element Method”. In: *Mathematical Models and Methods in Applied Sciences* 24.08, pp. 1541–1573. DOI: [10.1142/S021820251440003X](https://doi.org/10.1142/S021820251440003X).
- Berkhoff, J.C.W. (1972). “COMPUTATION OF COMBINED REFRACTION - DIFFRACTION”. In: *Coastal Engineering Proceedings* 1.13, p. 23. DOI: [10.9753/icce.v13.23](https://doi.org/10.9753/icce.v13.23).
- Booij, N. (1983). “A note on the accuracy of the mild-slope equation”. In: *Coastal Engineering* 7.3, pp. 191–203. DOI: [https://doi.org/10.1016/0378-3839\(83\)90017-0](https://doi.org/10.1016/0378-3839(83)90017-0).
- Cook, Megan, Frédéric Bouchette, Bijan Mohammadi, Léa Sprunck, and Nicolas Fraysse (2021). “Optimal Port Design Minimizing Standing Waves with A Posteriori Long Term Shoreline Sustainability Analysis”. en. In: *China Ocean Engineering* 35.6, pp. 802–813. DOI: [10.1007/s13344-021-0071-7](https://doi.org/10.1007/s13344-021-0071-7).

- Geuzaine, Christophe and Jean-François Remacle (2009). “Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities”. In: *International journal for numerical methods in engineering* 79.11, pp. 1309–1331.
- Guo, Junke (2002). “Simple and explicit solution of wave dispersion equation”. In: *Coastal Engineering* 45.2, pp. 71–74. DOI: [https://doi.org/10.1016/S0378-3839\(02\)00039-X](https://doi.org/10.1016/S0378-3839(02)00039-X).
- Helmholtz, Professor (1868). “XLIII. On discontinuous movements of fluids”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 36.244, pp. 337–346.
- Herrera, César, Ricardo Corrales-Barquero, Jorge Arroyo-Esquivel, and Juan G Calvo (2023). “A numerical implementation for the high-order 2D virtual element method in MATLAB”. In: *Numerical Algorithms* 92.3, pp. 1707–1721.
- Mascotto, Lorenzo, Ilaria Perugia, and Alexander Pichler (2019). “A nonconforming Trefftz virtual element method for the Helmholtz problem”. In: *Mathematical Models and Methods in Applied Sciences* 29.09, pp. 1619–1656. DOI: [10.1142/S0218202519500301](https://doi.org/10.1142/S0218202519500301).
- Munk, Walter (1949). “The solitary wave theory and its application to surf problems”. In: *Annals of the New York Academy of Sciences* 51, pp. 376–424. DOI: [10.1111/j.1749-6632.1949.tb27281.x](https://doi.org/10.1111/j.1749-6632.1949.tb27281.x).
- Perugia, Ilaria, Paola Pietra, and Alessandro Russo (2016). “A plane wave virtual element method for the Helmholtz problem”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 50.3, pp. 783–808. DOI: [10.1051/m2an/2015066](https://doi.org/10.1051/m2an/2015066).
- Sutton, Oliver J (2017). “The virtual element method in 50 lines of MATLAB”. In: *Numerical Algorithms* 75, pp. 1141–1159.
- Talischi, Cameron, Glauco H Paulino, Anderson Pereira, and Ivan FM Menezes (2012). “PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab”. In: *Structural and Multidisciplinary Optimization* 45, pp. 309–328.

Remarques à traiter:

- **Parler de Problèmes de spiriusmodes ?**
- **Ajouter dans la partie discussion les trucs de maillage sur la discussion: raffinement sur les modes etc, glue etc.**
Eventuellement ajouter port maillé polyédriquement ?