

Internship Report:

**Simulation of acoustic waves in a fluid contained
in a nuclear tank - Fluid-Structure Coupling**

Hanna CHETOUANE

August 21, 2025

Avnir Energy

Contents

1	Introduction	3
1.1	Company Presentation	3
1.2	Context	3
1.3	Company technique and time reversal	4
2	Internship Objectives	5
3	Tools used	5
3.1	Mathematical and numerical tools	5
3.2	Choice of implementation in C++	6
4	Equations	6
4.1	Equation of elasticity	6
4.2	Wave equation	7
4.3	Fluid-structure coupling strategy	8
4.4	Initial conditions and boundary conditions	8
5	Problem to be solved	9
5.1	Complete solution system	9
6	Theory	10
6.1	Mathematical formulation of the elasticity equation	10
6.1.1	Variational formulation	10
6.1.2	Weak formulation	10
6.1.3	Temporal discretization	11
6.1.4	Spatial discretization and matrix formulation	12
6.2	Mathematical formulation of the wave equation	12
6.2.1	Variational formulation	12
6.2.2	Weak formulation	13
6.2.3	Temporal discretization	13
6.2.4	Spatial discretization and matrix formulation	14

7	Implementation	14
7.1	Monolithic resolution	14
7.1.1	Structure and resolution of equations	14
7.1.2	Current limitations	16
7.2	Solving coupled equations	16
8	Simulation	17
8.1	Choice of parameters	17
8.1.1	Fluid parameters	17
8.1.2	Solid parameters	17
8.1.3	Source function used	18
8.1.4	Initial and boundary conditions	18
8.1.5	Stability conditions	18
8.2	Geometries used	19
8.3	Results	19
9	Analysis of computation times	21
10	Conclusion	22
11	Perspectives	23
12	Annex	24
12.1	Wave equation - With ALE method	24
12.1.1	Equation	24
12.1.2	ALE resolution	24
12.1.3	Variational and matrix formulation of ALE	25
12.1.4	Variational formulation of the wave equation with ALE	25
12.1.5	Temporal discretization	26
12.1.6	Spatial discretization and matrix formulation	26

1 Introduction

This project was part of my first-year internship for the CSMI (Scientific Computing and Information Mathematics) Master's program at the University of Strasbourg. It was carried out in collaboration with Rayen TLILI, a second-year student in the same Master's program, and lasted two months.

My internship took place at Avnir Energy, a company located in Villeurbanne that specializes in vibration analysis. The collaboration between the CSMI master's program and Avnir Energy has been ongoing for two years, notably with the work of Sasha ALIDADI HERAN, Rayen TLILI, and Marie SENGLER (former CSMI master's students).

This project is a continuation of this work and aims to develop a digital tool for modeling the propagation of acoustic waves generated by an impact on a solid structure. More generally, it aims to contribute to the integrity and ongoing maintenance of nuclear power plant tanks, where these fluid-structure coupling phenomena play a crucial role.

1.1 Company Presentation



Avnir Energy is an engineering company founded in 2015, specializing in several fields, including nuclear, oil and gas, hydraulics, thermal, renewable energies, and civil engineering. The company offers vibration monitoring services (vibrations, acoustics, ultrasound, acoustic emissions, ionizing radiation) and seismic detection to ensure the safety and maintenance of infrastructure.

The company currently has around 30 employees and is structured around three main divisions: R&D (developing advanced simulation methods), operations (carrying out projects for industrial clients), and sales (directly linked to market needs and seeking new partnerships).

Since its creation, Avnir Energy has worked with more than 200 clients and numerous partners. Among them are EDF (nuclear and hydraulic energy), Sonorhc Technologies (acoustics and non-destructive testing), the CEA (applied research in nuclear engineering), and academic laboratories such as INSA Lyon and CEMOSIS at the University of Strasbourg.

1.2 Context

The fluids present in nuclear power plant tanks are most often water, liquid sodium, or molten lead. For example, when a tank contains liquid lead (whose melting point is close to 330°C), it is impossible to conduct experimental tests involving striking the wall directly from inside the tank to study its resistance.



In this context, numerical simulation is an essential tool for analyzing the behavior of these structures and enabling predictive and safe maintenance.

This issue involves two main fields:

- fluid mechanics, which describes the evolution of pressure (scalar), velocity (vector), and density fields in the fluid, as well as how acoustic waves propagate within it
- solid mechanics, which studies the displacement field (vector), deformations, and stresses in the structure when it is subjected to external stresses

The interaction between these two fields, i.e., fluid-structure coupling, is at the heart of this project. It allows us to model the propagation of waves generated by an impact in the solid and transmitted to the fluid inside it. These approaches contribute directly to improving the safety and ongoing maintenance of nuclear tanks through the development of reliable, robust, and predictive digital tools.

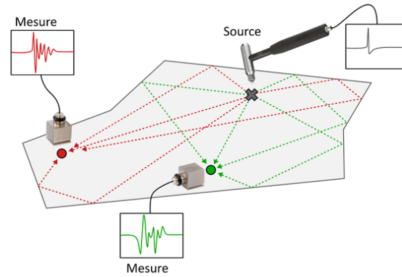
1.3 Company technique and time reversal

One of the main techniques used by Avnir Energy for vibration and acoustic analysis is time reversal.

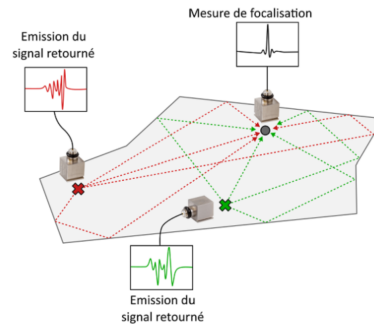
This method is a physical technique initially developed in the electromagnetic field (visible, microwaves) and subsequently applied to ultrasound by Mathias Fink, starting in 1987.

This principle is based on the fact that a signal recorded in a medium can be reversed in time and then reintroduced into the same medium. In the case of a dissipative wave, the wave thus regenerated is precisely refocused at its point of origin.

In the context of structural monitoring, time reversal takes place in two phases. The first is a learning phase carried out on the structure in its initial state. This can be either experimental, by striking the structure directly with a hammer when the areas to be monitored are accessible, or digital, by simulating it. Avnir Energy is working on a tool that allows this to be done digitally.



The second phase consists of injecting the signals back in time, which allows the evolution of the structure's responses over time to be monitored and analyzed.



This technique offers several advantages for monitoring and maintaining structures. It allows the source of impact to be precisely located in any area, defects to be detected early and accurately, and their development to be anticipated.

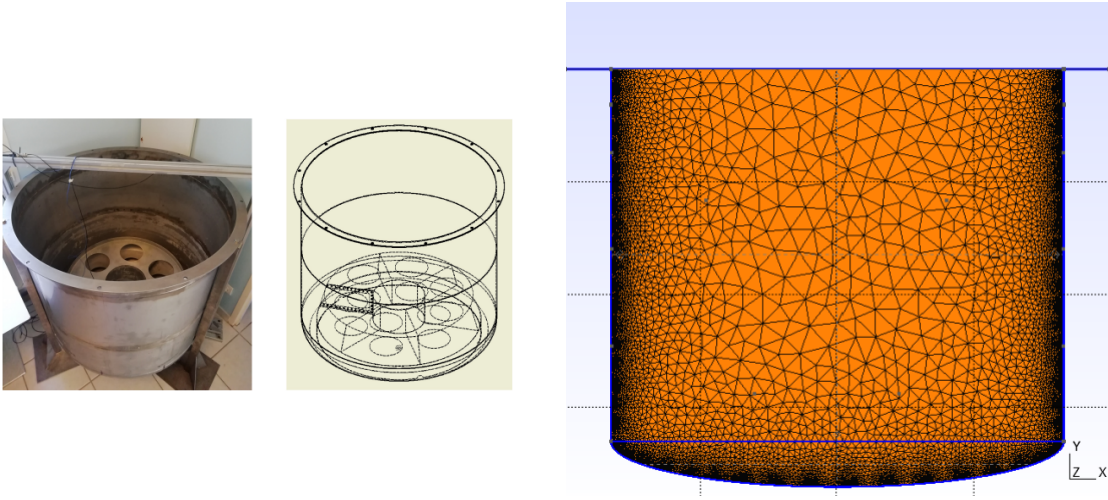
In this context, applied to the monitoring of a nuclear reactor tank (Sinudyn' project by Avnir Energy and Sonorch Technologies in collaboration with), my objective is to develop a numerical model to simulate the propagation of waves in a fluid at rest within the tank following an impact on the outer wall, in order to exploit these results for diagnostic and predictive maintenance applications, in particular by providing acceleration data for the learning phase.

2 Internship Objectives

The main objective of the internship is to develop a digital tool for simulating wave propagation in a fluid at rest following the application of a force on a solid structure, dynamically and on a simplified geometry of a nuclear power plant tank in 2D and 3D. This tool will enable Avnir Energy to exploit the principle of time reversal for diagnostic and predictive maintenance applications.

More specifically, my objective is to implement fluid-structure coupling in order to model the effect of a shock applied to the solid and transmitted to the fluid. To do this, I will first need to monolithically implement the elasticity equation to model the shock on the tank wall and the wave equation to simulate the propagation of waves in the fluid following this shock. Then, in a second step, fluid-structure coupling will need to be implemented by injecting the fluid pressure into the elasticity equations and the solid velocity into the wave equations.

The simulation will be performed on a simplified geometry of a nuclear power plant tank approximately 94 cm wide, 78 cm high, and 3 mm thick, in 2D and 3D.



Development will be in C++ using the Feel++ library, allowing for simpler and more efficient coding of partial differential equations.

3 Tools used

3.1 Mathematical and numerical tools

Finite element method (FEM): The main method used for this project, and more generally for numerically solving partial differential equations (PDE), is the finite element method (FEM). This method is used to discretize space and transform a continuous problem into a discrete problem, which can then be solved numerically.

Spatial discretization is performed here using the continuous Galerkin finite element method with Lagrange elements. It consists of searching for an approximate solution in a subspace of test functions by imposing that the residue of the equation be zero. Lagrange elements are polynomials defined locally on each element of the mesh and are used to approximate the solution within the element. Each node of the mesh has an associated function that is 1 at the node and 0 at the others, which facilitates the assembly of the overall system.

Feel++: Feel++ is a C++ library dedicated to the numerical solution of partial differential equations using Galerkin methods, in particular finite elements and spectral elements (SEM), which use higher-degree polynomials to achieve greater accuracy. Feel++ allows you to manage weak formulation, matrix assembly, geometry meshing, parallel computing with mpirun, and visualization

on Paraview.

Feel++ also provides specialized toolboxes, such as the FSI (Fluid-Structure Interaction) toolbox, which allows efficient simulation of fluid-structure coupling from a .json file and a .cfg configuration file, and the CFPDE (Coefficient Form Partial Differential Equations) toolbox for the general solution of partial differential equations.

GMSH: GMSH is used to generate 2D and 3D geometries and meshes for our simulations, which can be used in Feel++.

Paraview: Paraview allows us to visualize the results of numerical simulations, and thus analyze the deformation of the solid following the application of force and the resulting wave propagation.

Gaya: Gaya is a computing machine used for our simulations, allowing several computing cores to be used simultaneously in parallel, enabling more accurate and longer simulations while significantly reducing simulation time.

Slurm: Slurm is a resource manager for Gaya. It allows us to organize and launch simulation jobs, manage parallel computing, and optimize the use of available resources.

GitHub: Github is used for collaborative code management, to keep a history of work and facilitate teamwork, particularly with Rayen TLILI. The project is located in a repository shared with Avnir Energy.

3.2 Choice of implementation in C++

Before this internship, I had carried out a project, also in collaboration with Avnir Energy, on the simulation of dynamic waves in a fluid at rest in a nuclear power plant tank using the Feel++ CFPDE toolbox. At the beginning of my internship, I tested the use of this toolbox to implement the elasticity equation in addition to the wave equation. This step went smoothly, but when I tried to set up the fluid-structure coupling via fluid pressure and solid velocity, I encountered some limitations. In fact, the coupling only took place through the boundary conditions, and certain numerical errors appeared during the execution of the simulations.

I finally decided to implement it directly in C++ using the Feel++ library, which allowed me to understand in detail how the equations and the coupling between the two equations were implemented and to optimize performance thanks to full access to the solutions. In fact, we opted for explicit rather than implicit time schemes (used in the toolbox). One of the main advantages of the explicit method is its speed in terms of calculation time. Each time step is obtained directly from the previous solution, without the need to invert a large matrix associated with the mesh. The solution is updated by direct, local calculation. However, its main disadvantage is its CFL (Courant-Friedrichs-Lewy) stability condition. For the simulation to remain stable, the time step must be small enough so that the wave does not skip elements and produce incorrect results.

The Feel++ FSI toolbox also allows fluid-structure coupling, but remains limited in terms of flexibility. The methods are coded implicitly, making it impossible for us to change the coupling strategy. By coding directly in C++, we were able to build a modular base, experiment with explicit or implicit schemes, and add future extensions, particularly regarding the integration of the ALE method and the development of a tool tailored to Avnir Energy's needs.

4 Equations

4.1 Equation of elasticity

The equation of elasticity is at the heart of solid mechanics and describes how a structure reacts when subjected to external forces. It links the displacement of the solid to the internal stress

generated by its deformation, taking into account the mechanical properties of the material. In the context of this project, it is used to model the propagation of a shock applied to the tank wall, tracking the deformation of the solid at each moment.

In the case where small deformations of the solid are assumed (linear case), two formulations exist. The static equation, where we assume that the system is in equilibrium (without time dependence), and the dynamic equation, which includes time dependence and allows us to describe transient phenomena (vibrations, waves). Since our objective is to study the propagation of a shock over time and its effect on the fluid, we use the dynamic equation.

The solution is the displacement field of the solid, denoted u_s , which is vectorial ($u_s = (u_x, u_y)$ in 2D and $u_s = (u_x, u_y, u_z)$ in 3D).

The dynamic elasticity equation is written as:

$$\rho_s \frac{\partial^2 u_s}{\partial t^2} - \nabla \cdot \sigma(u_s) = f_s \quad \text{in } \Omega_s \times [0, T]$$

where σ is the stress tensor, given by:

$$\sigma(u_s) = \lambda(\nabla \cdot u_s)I + 2\mu\epsilon(u_s)$$

with I the identity matrix ϵ the deformation tensor given by:

$$\epsilon(u_s) = \frac{1}{2}(\nabla u_s + \nabla u_s^T)$$

and $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ et $\mu = \frac{E}{2(1+\nu)}$ the Lamé coefficients related to E the Young's modulus and ν the Poisson's ratio, which describe the properties of the material.

The different quantities are:

- u_s : displacement of the solid (vector field, m)
- ρ_s : density of the solid (kg/m^3)
- σ : stress tensor (Pa)
- ϵ : strain tensor (unitless)
- E : Young's modulus (Pa)
- ν : Poisson's ratio (Pa)
- f_s : source term (N/m^3)
- Ω_s : spatial domain of the solid (2D or 3D, m^2 or m^3)
- $[0, T]$: time domain (s)

To complete the system, initial and boundary conditions will be specified later.

4.2 Wave equation

The wave equation is fundamental in fluid mechanics and acoustics. It describes the propagation of small pressure disturbances in a fluid at rest (linear case). In our case, it allows us to model the response of the fluid following the impact on the tank wall and to study how acoustic waves propagate and interact with the structure.

The solution studied is the acoustic pressure, which depends on time and is denoted by p . It is a scalar field, unlike the displacement of the solid, which is vectorial.

The wave equation in the dynamic and linear case is written as:

$$\frac{\partial^2 p}{\partial t^2} - c^2 \Delta p = f_f \quad \text{in } \Omega_f \times [0, T]$$

where the term $\frac{\partial^2 p}{\partial t^2}$ represents the inertia term and reflects the temporal evolution of pressure, Δp is the spatial diffusion term, which reflects the propagation of the wave in the fluid, and c is

the speed of sound in the fluid, which represents the speed at which acoustic waves propagate and depends on the thermodynamic properties of the fluid.

In general, the wave equation is only valid for compressible fluids (fluids whose density varies slightly under the effect of a disturbance) since the propagation of an acoustic wave is based on small variations in density. The speed is then related to the properties of the fluid: $c = \sqrt{\frac{K}{\rho_0}}$ where K is the compressibility modulus (or bulk modulus) and ρ_0 is the density of the fluid at rest. In the case of an incompressible fluid (such as water), this equation no longer makes sense, since the density is constant and no acoustic waves can propagate. However, in many cases, a fluid is approximated as being weakly compressible, which allows us to retain a wave equation for pressure (as here with water, where we consider c to be finite). We will use this approach in this project.

The different quantities are:

- p : pressure field (scalar, in Pa)
- c : speed of sound in the fluid (m/s)
- f_f : source term (Ricker wavelet, Pa/s²)
- Ω_f : spatial domain of the fluid (2D or 3D, m^2 or m^3)
- $[0, T]$: time domain (s)

As with the elasticity equation, this system is not yet complete; it is necessary to add the initial and boundary conditions, which we will detail below.

Note: The source term f_f is added only when one wants to solve the wave equations in isolation, in order to avoid a zero solution. In the case of coupling between the fluid and the structure, this source term will no longer be meaningful. The excitation of the fluid comes directly from the interaction with the solid wall.

4.3 Fluid-structure coupling strategy

Fluid-structure coupling occurs at the interface between the two environment, Ω_s and Ω_f , denoted Γ_{fsi} . It is performed using Neumann boundary conditions applied to the solid and fluid equations.

For the elasticity equation, the fluid pressure acts as a dynamic load applied to the solid wall:

$$\sigma(u_s) \cdot n_s = -pn_s \quad \text{on } \Gamma_{fsi},$$

where n_s is the normal vector to the surface of the solid, $\sigma(u_s)$ is the stress tensor, and p is the fluid pressure.

For the wave equation, the displacement of the solid causes a pressure variation according to the normal velocity of the solid:

$$\frac{\partial p}{\partial n_f} = \rho_f \frac{\partial u_s}{\partial t} \cdot n_s \quad \text{on } \Gamma_{fsi},$$

with n_f the normal vector pointing toward the fluid and ρ_f the density of the fluid.

So, the movement of the solid generates pressure variations in the fluid. The pressure exerted by the fluid acts back on the wall of the solid, ensuring a coherent dynamic interaction and conservation of energy.

4.4 Initial conditions and boundary conditions

We consider the system to be initially at rest, i.e., at the initial time $t = 0$, the displacement of the solid and the pressure of the fluid are zero:

$$u_s(t = 0) = 0 \quad ; \quad p(t = 0) = 0$$

Physically, this means that the structure is not yet deformed and that no stress is exerted on the fluid side.

The boundary conditions reflect the physical conditions imposed on the domain. On the fluid side, zero pressure is imposed on the top of the tank Γ_{top} , which corresponds to an open surface.

On the solid side, zero displacement is imposed on the tank flange Γ_{fixed} , which reflects the fact that this part of the structure is fixed and cannot move.

These conditions are Dirichlet conditions, as they directly impose the value of the unknowns on the boundaries of the domain.

On the fluid-structure interface Γ_{fsi} , the coupling conditions are Neumann type, as they impose the force exchanged between the fluid and the solid, expressed via the derivatives with respect to the normal (stress tensor applied to the normal and normal derivative of the pressure).

These initial and boundary conditions make it possible to reflect the physical constraints of the system and also ensure the uniqueness of the solution. By setting certain values (Dirichlet) or fluxes (Neumann) on the boundaries and interface, they eliminate indeterminate degrees of freedom and allow a single solution to be obtained that is consistent with the desired physics.

5 Problem to be solved

5.1 Complete solution system

After setting up the equations and specifying the initial and boundary conditions, we can write the complete system to be solved:

$$\left\{ \begin{array}{l} \rho_s \frac{\partial^2 u_s}{\partial t^2} - \nabla \cdot \sigma(u_s) = f_s \quad \text{in } \Omega_s \times [0, T] \\ \frac{\partial^2 p}{\partial t^2} - c^2 \Delta p = 0 \quad \text{in } \Omega_f \times [0, T] \\ u_s(t=0) = 0 \quad \text{in } \Omega_s \times \{0\} \\ p(t=0) = 0 \quad \text{in } \Omega_f \times \{0\} \\ u_s(x, y) = 0 \quad \text{in } \Gamma_{fixed} \times [0, T] \\ p(x, y) = 0 \quad \text{in } \Gamma_{top} \times [0, T] \\ \sigma(u_s) \cdot n_s = -pn_s \quad \text{in } \Gamma_{fsi} \times [0, T] \\ \frac{\partial p}{\partial n_f} = -\rho_f \left(\frac{\partial u_s}{\partial t} \cdot n_s \right) \quad \text{in } \Gamma_{fsi} \times [0, T] \end{array} \right.$$

With:

- u_s : displacement of the solid (vector field, in m)
- p : pressure field (scalar, in Pa)
- $\sigma(u_s) = \lambda(\nabla \cdot u_s)I + 2\mu\epsilon(u_s)$: stress tensor (Pa)
- $\epsilon(u_s) = \frac{1}{2}(\nabla u_s + \nabla u_s^T)$: strain tensor (unitless)
- $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$ where E is Young's modulus (Pa) and ν is Poisson's ratio (Pa)
- ρ_s : density of the solid (in kg/m^3)
- ρ_f : density of the fluid (in kg/m^3)
- f_s : source term applied to the solid (in N/m^3)
- c : speed of sound in the fluid (in m/s)
- Ω_s : spatial domain of the solid (2D or 3D, in m^2 or m^3) with boundary Γ_{fsi} , Γ_{fixed}

- Ω_f : spatial domain of the fluid (2D or 3D, in m^2 or m^3) with boundary Γ_{fsi} , Γ_{top}
- $[0, T]$: time domain (in s)

6 Theory

To solve our system of two coupled equations numerically, we must proceed in several steps. We will begin by writing the variational formulation of the equations, move on to the weak formulation, and then discretize it temporally and spatially to obtain the final matrix formulations.

We will start with the mathematical formulation of the elasticity equation, then detail it for the wave equation.

6.1 Mathematical formulation of the elasticity equation

6.1.1 Variational formulation

Let $u_s \in [H^1(\Omega_s)]^d$ be the displacement field of the solid and $v \in [H^1(\Omega_s)]^d$ ($d = 2$ or 3) be a test function with the same boundary conditions as u_s .

We multiply the elasticity equation by the test function v and integrate over the domain Ω_s :

$$\int_{\Omega_s} \rho_s \frac{\partial^2 u_s}{\partial t^2} \cdot v \, d\Omega_s - \int_{\Omega_s} (\nabla \cdot \sigma(u_s)) \cdot v \, d\Omega_s = \int_{\Omega_s} f_s \cdot v \, d\Omega_s$$

We apply Green's formula to the second term:

$$\int_{\Omega_s} \rho_s \frac{\partial^2 u_s}{\partial t^2} \cdot v \, d\Omega_s + \int_{\Omega_s} \sigma(u_s) : \nabla v \, d\Omega_s = \int_{\Omega_s} f_s \cdot v \, d\Omega_s + \int_{\partial\Omega_s} (\sigma(u_s) \cdot n_s) \cdot v \, d\partial\Omega_s$$

Considering the coupling boundary conditions $\sigma(u_s) \cdot n_s = -pn_s$ on Γ_{fsi} (p being the acoustic pressure) and Dirichlet conditions $u_s = 0$ everywhere else, this gives us:

$$\int_{\Omega_s} \rho_s \frac{\partial^2 u_s}{\partial t^2} \cdot v \, d\Omega_s + \int_{\Omega_s} \sigma(u_s) : \nabla v \, d\Omega_s = \int_{\Omega_s} f_s \cdot v \, d\Omega_s - \int_{\Gamma_{fsi}} (pn_s) \cdot v \, d\Gamma_{fsi}$$

Note: Neumann boundary conditions can also be incorporated. They would appear naturally in the variational formulation, via Green's formula, in the form of terms added to the second member.

6.1.2 Weak formulation

We identify the bilinear term $a(u_s, v)$ and the linear term $l(v)$ associated with the variational formulation:

- $a(u_s, v) = \int_{\Omega_s} \rho_s \frac{\partial^2 u_s}{\partial t^2} \cdot v \, d\Omega_s + \int_{\Omega_s} \sigma(u_s) : \nabla v \, d\Omega_s$
- $l(v) = \int_{\Omega_s} f_s \cdot v \, d\Omega_s - \int_{\Gamma_{fsi}} (pn_s) \cdot v \, d\Gamma_{fsi}$

By Korn's lemma, since $a(u_s, v)$ is bilinear, coercive, and continuous, and $l(v)$ is linear and continuous, there exists a unique weak solution to the associated problem.

6.1.3 Temporal discretization

To handle the acceleration term $\frac{\partial^2 u_s}{\partial t^2}$, we consider the Newmark scheme:

$$\frac{\partial^2 u_s}{\partial t^2} = a_s^{n+1} \approx \frac{u_s^{n+1} - u_s^n - \Delta t v_s^n - \frac{\Delta t^2}{2}(1-2\beta)a_s^n}{\beta \Delta t^2}$$

where:

- u_s^{n+1} : displacement of the solid at step $n+1$ (solution to solve)
- u_s^n : displacement of the solid at step n , already calculated at the previous time (variable to be stored)
- $v_s^n = \dot{u}_s^n$: velocity of the solid at step n calculated at the previous time (variable to be stored)
- $a_s^{n+1} = \ddot{u}_s^n$: acceleration of the solid at step $n+1$.

Note: Feel++ already implements the Newmark time scheme with fixed parameters: $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$, which corresponds to an unconditionally stable scheme.

Thus, the time-discretized variational formulation becomes:

$$\int_{\Omega_s} \rho_s \frac{u_s^{n+1} - u_s^n - \Delta t v_s^n - \frac{\Delta t^2}{2}(1-2\beta)a_s^n}{\beta \Delta t^2} \cdot v \, d\Omega_s + \int_{\Omega_s} \sigma(u_s^n) : \nabla v \, d\Omega_s = \int_{\Omega_s} f_s \cdot v \, d\Omega_s - \int_{\Gamma_{fsi}} (p^n n_s) \cdot v \, d\Gamma_{fsi}$$

By replacing a_s^{n+1} with its expression in Newmark's scheme and isolating the term u_s^{n+1} , this brings us back to the following explicit system:

$$\begin{aligned} \int_{\Omega_s} \frac{\rho_s}{\beta \Delta t^2} u_s^{n+1} \cdot v \, d\Omega_s &= \int_{\Omega_s} \rho_s \frac{u_s^n + \Delta t v_s^n + \frac{\Delta t^2}{2}(1-2\beta)a_s^n}{\beta \Delta t^2} \cdot v \, d\Omega_s \\ &+ \int_{\Omega_s} [-\sigma(u_s^n) : \nabla v + f_s \cdot v] \, d\Omega_s - \int_{\Gamma_{fsi}} (p^n n_s) \cdot v \, d\Gamma_{fsi} \end{aligned}$$

Note: Here, the term $\frac{u_s^n + \Delta t v_s^n + \frac{\Delta t^2}{2}(1-2\beta)a_s^n}{\beta \Delta t^2}$ is explained, but in practice it is directly stored and usable by Feel++ with the `polyDeriv()` function of the Newmark scheme.

This expression can be rewritten as:

$$a_{mass}(u^{n+1}, v) = a_{stiff}(u^n, v) + l(v) \quad (1)$$

with:

$$\begin{aligned} a_{mass}(u, v) &= \int_{\Omega_s} \frac{\rho_s}{\beta \Delta t^2} u \cdot v \, dx \\ a_{stiff}(u, v) &= - \int_{\Omega_s} \sigma(u) : \nabla v \, dx \\ l(v) &= \int_{\Omega_s} \rho_s \text{polyDeriv}() \cdot v \, d\Omega_s - \int_{\Gamma_{sf}} p^n n_s \cdot v \, d\Gamma_{fsi} + \int_{\Omega_s} f_s^n \cdot v \, dx \end{aligned}$$

6.1.4 Spatial discretization and matrix formulation

Let $\{\phi_j\}_j$ be a basis of $[H^1(\Omega_s)]^d$ ($d = 2$ or 3). We take the test function $\mathbf{v}_s = \phi_i$. By approximating the displacement using finite elements, we obtain the spatial discretization:

$$\mathbf{u}_s(x, t) \approx \sum_j U_j(t) \phi_j(x)$$

This gives us the following matrix formulation:

$$\frac{\rho_s}{\beta \Delta t^2} \mathbf{M} u_s^{n+1} = \rho_s \mathbf{M}^{\text{polyDeriv}} u_s^n - \mathbf{K} \mathbf{u}_s^n + \mathbf{f}_s^n - \mathbf{p}^n$$

- $\mathbf{M} = \int_{\Omega_s} \phi_i \cdot \phi_j dx$: mass matrix
- $\mathbf{M}^{\text{polyDeriv}} = \int_{\Omega_s} \text{polyDeriv}() \cdot \phi_i dx$: mass matrix with terms from the right-hand side of Newmark's time discretization, from a_s^n
- $\mathbf{K} = \int_{\Omega_s} \sigma(\phi_j) : \nabla \phi_i dx$: stiffness matrix
- $\mathbf{p}_{\Gamma_{sf}}^n = \int_{\Gamma_{sf}} p^n n_s \cdot \phi_i ds$: vector with the coupling condition with the fluid
- $\mathbf{f}_{\Omega_s}^n = \int_{\Omega_s} f_s^n \cdot \phi_i dx$: vector with the source term
- $\mathbf{u}_s^{n+1}$: displacement of the solid (unknown to be solved at time $n + 1$)

6.2 Mathematical formulation of the wave equation

6.2.1 Variational formulation

Note: Here we have left the source term f_f , but in practice $f_f = 0$ if we want to solve the wave equation coupled to the solid.

Let $p \in H^1(\Omega_f)$ be the fluid pressure field and $q \in H^1(\Omega_f)$ a test function with the same boundary conditions as p .

We multiply the wave equation by the test function q and integrate over the domain Ω_f :

$$\int_{\Omega_f} \frac{\partial^2 p}{\partial t^2} q d\Omega_f - \int_{\Omega_f} c^2 (\Delta p) q d\Omega_f = \int_{\Omega_f} f_f q d\Omega_f$$

We apply Green's formula:

$$\int_{\Omega_f} \frac{\partial^2 p}{\partial t^2} q d\Omega_f + \int_{\Omega_f} c^2 \nabla p \cdot \nabla q d\Omega_f = \int_{\Omega_f} f_f q d\Omega_f + \int_{\partial\Omega_f} c^2 \frac{\partial p}{\partial n_f} q d\partial\Omega_f$$

Considering coupling boundary conditions: $\frac{\partial p}{\partial n_f} = -\rho_f \frac{\partial u_s}{\partial t} \cdot n_s$ and $d_f = u_s$ on Γ_{fsi} and Dirichlet conditions $p = 0$ everywhere else, this gives us:

$$\int_{\Omega_f} \frac{\partial^2 p}{\partial t^2} q d\Omega_f + \int_{\Omega_f} c^2 \nabla p \cdot \nabla q d\Omega_f = \int_{\Omega_f} f_f q d\Omega_f - \int_{\Gamma_{fsi}} c^2 \rho_f \left(\frac{\partial u_s}{\partial t} \cdot n_s \right) q d\Gamma_{fsi}$$

6.2.2 Weak formulation

We identify the bilinear term $a(p, q)$ and the linear term $l(q)$ associated with the variational formulation:

- $a(p, q) = \int_{\Omega_f} \frac{\partial^2 p}{\partial t^2} q \, d\Omega_f + \int_{\Omega_f} c^2 \nabla p \cdot \nabla q \, d\Omega_f$
- $l(q) = \int_{\Omega_f} f_f q \, d\Omega_f - \int_{\Gamma_{fsi}} c^2 \rho_f \left(\frac{\partial u_s}{\partial t} \cdot n_s \right) q \, d\Gamma_{fsi}$

By the Lax-Milgram theorem, since $a(p, q)$ is bilinear, coercive, and continuous, and $l(q)$ is linear and continuous, there exists a unique weak solution to the associated problem.

6.2.3 Temporal discretization

We consider a centered BDF scheme of order 2 to discretize the temporal term of the wave equation in time:

$$\frac{\partial^2 p^{n+1}}{\partial t^2} \approx \frac{p^{n+1} - 2p^n + p^{n-1}}{\Delta t^2}$$

where:

- p^{n+1} : fluid pressure at time step $n + 1$ (solution to be calculated)
- p^n : fluid pressure at time step n (value known at the previous time)

For the boundary condition term, the velocity of the solid is approximated using a Newmark scheme:

$$\frac{\partial u_s^{n+1}}{\partial t} = v_s^{n+1} \approx v_s^n + \Delta t[(1 - \gamma)a_s^n + \gamma a_s^{n+1}]$$

where:

- v_s^n : velocity of the solid at step n (known value)
- a_s^{n+1} : acceleration of the solid at step $n + 1$ (value calculated just before when solving the elasticity equation)

Thus, the time-discretized variational formulation becomes:

$$\int_{\Omega_f} \frac{p^{n+1} - 2p^n + p^{n-1}}{\Delta t^2} q \, d\Omega_f + \int_{\Omega_f} c^2 \nabla p^n \cdot \nabla q \, d\Omega_f = \int_{\Omega_f} f_f^n q \, d\Omega_f - \int_{\Gamma_{fsi}} c^2 \rho_f (v_s^{n+1} \cdot n_s) q \, d\Gamma_{fsi}$$

And isolating the term p^{n+1} :

$$\int_{\Omega_f} \frac{p^{n+1}}{\Delta t^2} q \, d\Omega_f = \int_{\Omega_f} \frac{2p^n - p^{n-1}}{\Delta t^2} q - c^2 \int_{\Omega_f} \nabla p^n \cdot \nabla q \, d\Omega_f + \int_{\Omega_f} f_f^n q \, d\Omega_f - \int_{\Gamma_{fsi}} c^2 \rho_f (v_s^{n+1} \cdot n_s) q \, d\Gamma_{fsi}$$

Note: The term v_s^{n+1} can be used directly by the Newmark time scheme in Feel++ (with `CurrentVelocity()`). In addition, $\frac{v_s^{n-1}}{\Delta t^2}$ can also be used with `polySecondDeriv()` from the Feel++ second-order bdf time scheme.

6.2.4 Spatial discretization and matrix formulation

Let $\{\psi_j\}_i$ be a basis for $H^1(\Omega_f)$. We take the test function $q = \psi_i$.

By approximating the pressure using finite elements, we obtain the spatial discretization. The spatial discretization is written as:

$$p(x, t) \approx \sum_j P_j(t) \psi_j(x)$$

This gives us the following matrix formulation:

$$\frac{1}{\Delta t^2} \mathbf{M} p^{n+1} = \frac{2}{\Delta t^2} \mathbf{M} p^n - \frac{1}{\Delta t^2} \mathbf{M} p^{n-1} - c^2 \mathbf{K} p^n + \mathbf{f}^n - \mathbf{v}_{bc}^{n+1}$$

With:

- $\mathbf{M}_{ij} = \int_{\Omega_f} \psi_i \psi_j dx$: mass matrix
- $\mathbf{K}_{ij} = \int_{\Omega_f} \nabla \psi_j \cdot \nabla \psi_i dx$: stiffness matrix
- $\mathbf{f}_i^n = \int_{\Omega_f} f_f^n \psi_i dx$: source term vector for the fluid at time n
- $\mathbf{v}_{bc,i}^{n+1} = \int_{\Gamma_{fsi}} c^2 \rho_f (v_s^{n+1} \cdot n_s) \psi_i ds$: Neumann boundary condition vector on Γ_{fsi}

The variational and matrix formulations presented above form the theoretical basis of our approach. In the next section, we will see how we can use them to numerically solve our equations with Feel++.

7 Implementation

The code is organized in C++ in the form of classes. Each equation (waves and elasticity) has its own class, and an additional class manages the coupling of the two equations. The meshes and the export of results are shared between the equations, allowing for common visualization with Paraview.

The simulation parameters are read from two files:

- a .json file containing the physical and numerical parameters (sources, temporal parameters, initial and boundary conditions, etc.)
- a .cfg file containing the general resolution parameters (dimension, mesh, solver parameters, Picard loop parameters)

7.1 Monolithic resolution

7.1.1 Structure and resolution of equations

The Wave and Elastic classes are structured similarly, with several common functions to facilitate reading. Object-oriented programming allows these objects to be constructed using a constructor and these equations to be solved using dedicated methods.

Constructor: The constructor calls two main functions:

- **readJson():** reads the .json file (whose path is specified in the .cfg file or on the command line) and retrieves the physical, temporal, and numerical parameters defined therein (source function, initial and boundary conditions, simulation time, etc.). Feel++ provides functions to read these files directly.
- **Initialize():** creates the objects needed for the simulation. It creates scalar (Pch for waves) and vector (Pchv for elasticity) function spaces, choosing the order of the Lagrange polynomial used to approximate the solution on each element. It also initializes the time scheme (bdf for the wave equation and Newmark for the elasticity equation) from the time parameters (time step, initial time, final time) and defines the initial solution. Then, it constructs the structures of the variational formulations (bilinear form and linear form). To optimize the resolution and since we are using an explicit method, the main term of the variational formulation, which corresponds to the solution at time $n + 1$, is constant over time. It is therefore calculated only once outside the loop, during initialization.

Solve(): assembles the variational formulation and solves the associated system.

- the bilinear and linear terms from the theoretical formulation are added
- **Source():** adds the source term that was read from the .json file
- adds the coupling term, if coupling is requested (for the wave equation, either the coupling term or the source term is added, but never both)
- **BoundaryConditions():** adds Neumann boundary conditions (the source can be considered a Neumann condition) in linear form, and Dirichlet boundary conditions are imposed after the linear system is assembled to directly set the solution values at the relevant boundaries
- solves the resulting matrix system with the Feel++ linear solver

run(): This is the main time loop. It launches the simulation in a monolithic manner. At each time step:

- calls the Solve() function with a null element of the same type as the coupling term as a parameter
- calls the Export() function of the class, which allows the solution to be exported for visualization in Paraview. For elasticity, velocity and acceleration are also exported. For waves, an export of velocity is available (commented in the code).
- saves the export to avoid conflicts during coupling (saves twice on the same time step)
- calls the nextStep() function, which increments the time and prepares the data for the next step.
- calculates the simulation time of the time step using the tic() and toc() functions of Feel++

Accessors and small auxiliary functions are also defined, making it easier to couple the two equations.

Launching the simulation: The code is separated into .hpp and .cpp files. The .cpp file, which contains the main(), reads the .cfg file and retrieves the path to the .json file, the dimension, and the mesh (.geo), then creates the export and constructs the Wave or Elastic object. Finally, it launches the simulation by simply calling the run() function of the associated object.

7.1.2 Current limitations

The resolution of monolithic equations is not yet fully complete. The code works overall, but several points need to be improved or corrected:

- Second-order BDF time scheme: it is necessary to be in the branch associated with output 2351 to use it. However, in practice, this generates a bug. I therefore preferred to code this term manually.
- Reading spatial parameters: currently, the flexibility of reading the source function and boundary conditions is limited. Only A (the amplitude of the source), f_c (the frequency of the source), and t (the time parameter) are supported. The use of the spatial parameters x , y , and z causes inconsistent results.
- Boundary conditions: it is possible to apply the same expression to several boundaries, but the combination of several conditions of the same type (Neumann or Dirichlet) is not yet supported. The addition of Robin conditions also remains to be developed.
- Explicit elasticity: the tests did not yield a stable solution despite a very small time step ($\Delta t = 10^{-9}$). **Note:** Implicitly, the bilinear term is reassembled at each time step, which causes the mass matrix to evolve. It is therefore necessary to work on a copy of the modified matrix in the loop to ensure resolution.
- Different exporters: Initially, a single common exporter was used for wave and elasticity resolution. Ultimately, two separate exporters were chosen (as in the FSI toolbox), which allows for clearer visualization (fluid pressure separated from solid displacement).

Theoretical note: As coded, the matrix formulation is theoretical. We used the discretized variational formulation to solve the system with Feel++. Nevertheless, the matrix formulation could be useful later on in order to symmetrize the terms associated with time step $n + 1$ using a Gauss-Lobatto quadrature method. The mass matrix becomes symmetric, and solving the system would amount to inverting a symmetric matrix, which would make the solution extremely simpler and faster.

7.2 Solving coupled equations

For the coupling between the wave equation and the elasticity equation, we chose to use Picard iterations. The idea is to solve the two equations alternately within the same time step, each time feeding the solution of one into the other, until convergence is achieved. This strategy strengthens the coupling between the two systems.

The “Coupling” class has two main attributes: a Wave object and an Elastic object. As with the other solvers, it is structured around a constructor and dedicated functions.

Constructor: The constructor takes the Wave and Elastic objects as arguments, creates the scalar and vector function spaces, and retrieves the common exporter.

Picard(): This is a loop that performs Picard iterations within a single time step.

- We start by solving the elasticity equation with the coupling term given as a parameter (fluid pressure projected into solid space at the fluid-solid interface, Γ_{fsi})
- We retrieve the displacement and velocity of the solid obtained
- We solve the wave equation with the coupling term given as a parameter (solid velocity projected into fluid space at Γ_{fsi})
- We retrieve the fluid pressure, solution of the equation

These steps are repeated until a maximum number of iterations is reached or until the difference between two successive iterations is less than a fixed tolerance. Both parameters, the tolerance and the maximum number of iterations, are read from the .cfg file.

In practice, a small number of iterations is sufficient. However, the parameters must be chosen carefully. Too many iterations greatly increase the calculation time, while too few iterations or too high a tolerance may lead to insufficient coupling and a loss of accuracy.

Launching the simulation: As with the monolithic equation solver, the code is separated into .hpp and .cpp files. The .cpp file, which contains the main() function, reads the .cfg file to retrieve the path to the .json file, the dimension, and the mesh (.geo), then creates the exporter and constructs the Wave and Elastic objects, followed by the Coupling object. Finally, the simulation is launched by simply calling the run() function of the Coupling object.

8 Simulation

8.1 Choice of parameters

In this section, we will present the physical parameters chosen for the fluid-structure coupling simulation. Here, we will limit ourselves to simulations with coupling, rather than simulations of monolithic equations, but the principle remains the same in both cases.

To illustrate the behavior of the coupling, we have chosen to model a fluid represented by water and a solid represented by stainless steel. Water is the typical cooling fluid in many industrial contexts, and stainless steel is a material often used for the walls of structures such as tanks, due to its strength and durability.

The code implementation allows all physical parameters to be set in a .json file. It can be referenced in the .cfg file or directly entered as a command line, which allows the data to be modified without having to touch the source code.

8.1.1 Fluid parameters

The fluid chosen is therefore water. It is the basic fluid and the most studied, especially in fluid-structure interaction problems and nuclear engineering.

The parameters needed to solve the wave equation in the fluid are:

- $c = 1480 \text{ m/s}$: speed of sound in water, which determines the propagation speed of acoustic waves
- $\rho_f = 1000 \text{ kg/m}^3$: density of water, which is used in the coupling term of the wave equation

8.1.2 Solid parameters

The solid chosen is stainless steel, a material often used for the walls of structures such as tanks and reservoirs.

The physical parameters required to solve the elasticity equation are:

- $E = 2.1 \times 10^{11} \text{ Pa}$: Young's modulus, characterizes the stiffness of the material. The higher it is, the stiffer the material
- $\nu = 0.3$: Poisson's ratio, which expresses the ratio between transverse and longitudinal deformation when a stress is applied
- From E and ν , we obtain the first and second (or shear modulus) Lamé coefficients, respectively $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} = 1.6 \times 10^{11}$ and $\mu = \frac{E}{2(1+\nu)} = 8 \times 10^{10}$
- $\rho_s = 7870 \text{ kg/m}^3$: density of the solid

8.1.3 Source function used

The source function used simulates the impact on the outer wall of the tank, in the solid. This source function will take the form of a Ricker wavelet. It will trigger an initial disturbance that will propagate and interact with the fluid, which will form the core of the fluid-structure coupling.

In 2D, it is written as:

$$f(x, y, t) = A\psi(t), \quad \text{with} \quad \psi(t) = -\sin(2\pi f_c t) \exp(-5(f_c t - 2)^2)$$

This is a Gaussian controlled by:

- a factor of 5: controls the width of the shock (the larger the coefficient, the narrower the Gaussian)
- a time lag of 2 in the exponential: allows us to start with a zero initial condition

The parameters are:

- $A = 10^3 Pa$: amplitude of the source, sets the intensity of the generated acoustic wave
- $f_c = 10^5 Hz$: central frequency of the source. The larger f_c is, the shorter the wave it produces, which may be visible early in the simulation. Conversely, if f_c is small, the wave will be much more spread out over time and we will have to wait longer to observe the results.

8.1.4 Initial and boundary conditions

For the simulations, we simply use the initial and boundary conditions given in the complete system in the theory section. That is:

- System at rest: $p(t = 0) = 0$; $u_s(t = 0) = 0$

For the boundary conditions, we consider the tank resting on a surface by its flange and open at the top. This translates into Dirichlet conditions:

- zero pressure: $p = 0$ on Γ_{top}
- zero displacement: $u_s = 0$ on $\Gamma_{s_{fixed}}$

8.1.5 Stability conditions

The scheme for solving the wave equation is explicit, which subjects it to the CFL stability criterion:

$$\Delta t \leq \frac{h}{c\sqrt{d}}$$

Where:

- h : space step (in m)
- c : speed of sound in the fluid (in m/s)
- d : dimension (2D or 3D)

For example, for $h = 0.01$, we can choose $\Delta t = 5 \times 10^{-7}$

However, the scheme for solving the elasticity equation is implicit, so there is no imposed stability. In fact, the chosen implicit scheme (Newmark with $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$) is unconditionally stable.

When using an explicit scheme for elasticity, stability requires:

$$\Delta t \leq \frac{h}{c_p}, \quad \text{with} \quad c_p = \sqrt{\frac{\lambda + 2\mu}{\rho_s}}$$

With:

- h : space step (in m)
- c_p : compression wave velocity (in m/s)
- λ and ν : Lamé coefficients (in Pa)
- ρ_s : density of the solid (in kg/m^3)

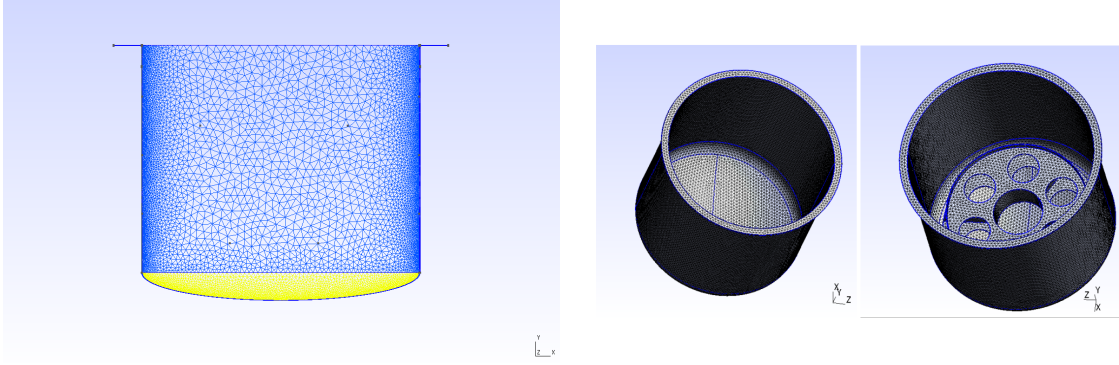
Therefore, in practice, we will most often take $h = 0.01$ and $\Delta t = 5 \times 10^{-7}$ to simulate fluid-structure coupling.

8.2 Geometries used

For the simulation geometry (2D and 3D), we reused Marie Sengler's work on modeling a simplified tank geometry.

The tank in question measures approximately 94 cm wide, 78 cm high, and has a wall thickness of 3 mm.

The geometries used for our simulations are simplified but retain the main mechanical characteristics of the problem. It is represented by a vertical cylinder with a rounded bottom and a flange on the upper part. A finer mesh is applied to the edges to allow the use of a coarse mesh while maintaining sufficient resolution at the solid wall (3 mm thick).



In order to simulate the impact, a small segment (2D) or a small surface (3D) is added to the left side of the wall to simulate the shock caused by a hammer blow. Measurement points have also been placed to enable the extraction of numerical results.

For a more realistic model, it would be necessary to add certain geometric details, such as the rounded reinforcements located at the bottom of the tank.

However, for the purposes of this work, this simplified model is still sufficient to analyze the behavior of the fluid-structure interaction.

8.3 Results

Using the parameters seen above, in this section we will look at the results obtained in 2D and 3D for fluid-structure coupling.

In the figures below, you can see the impact on the tank, the displacement of the solid, and the effects in the fluid with the propagation of acoustic waves.

Here are some excerpts from the videos generated with Paraview. The first figures are the 2D results, with the fluid pressure on the left and the displacement of the solid on the right. For the 3D results, it is the opposite, with the fluid pressure on the right and the displacement of the solid on the left.

For both visualizations, you can see the shock at the top left on the solid wall (source function with Ricker wavelet), the deformation of the solid, and the propagation of the waves generated. We observe that the propagation of the waves is consistent, both on the structure (wave moving across the thickness of the wall) and in the fluid (waves well transmitted from the structure to the fluid, due to fluid-structure coupling).

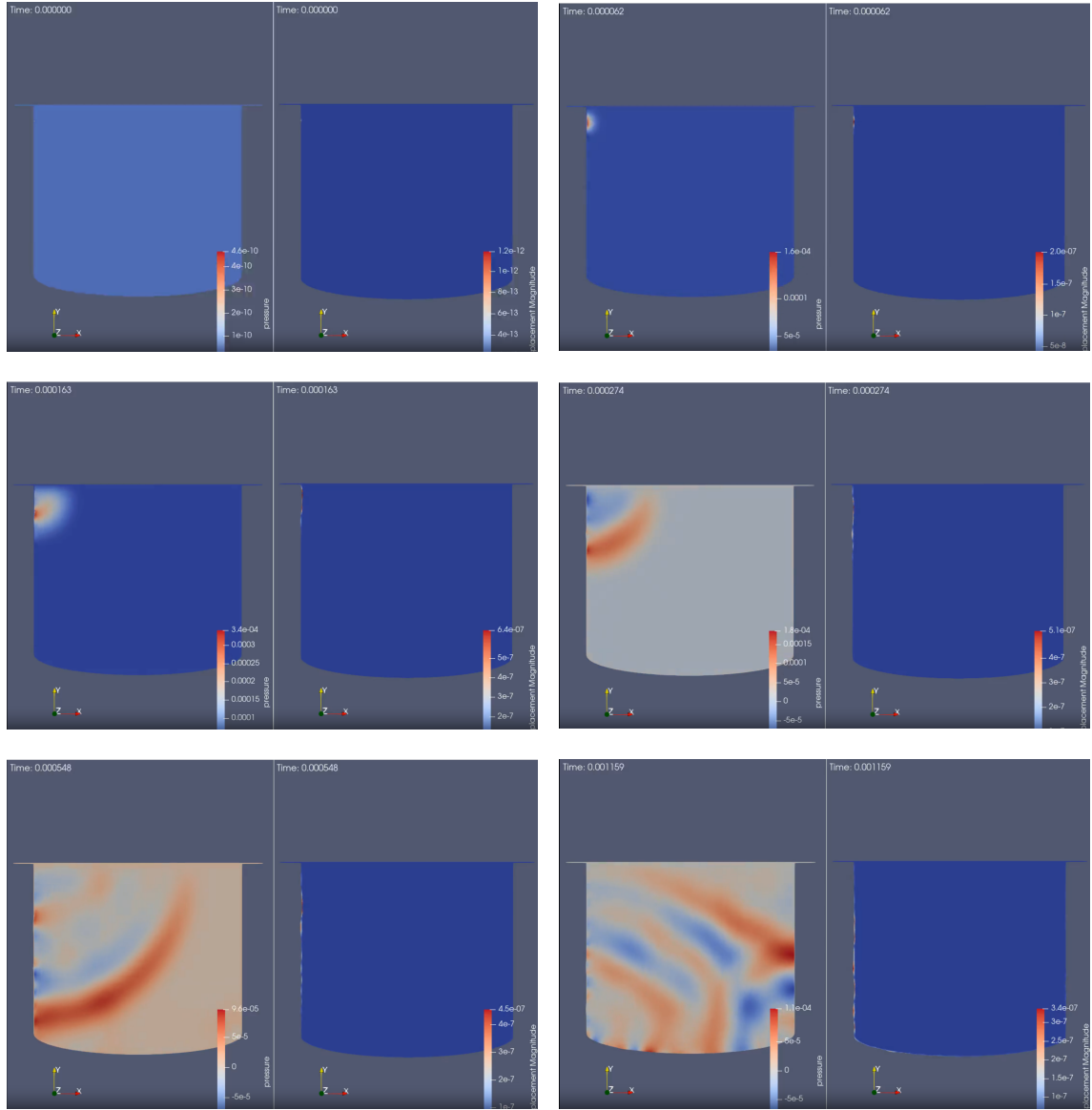
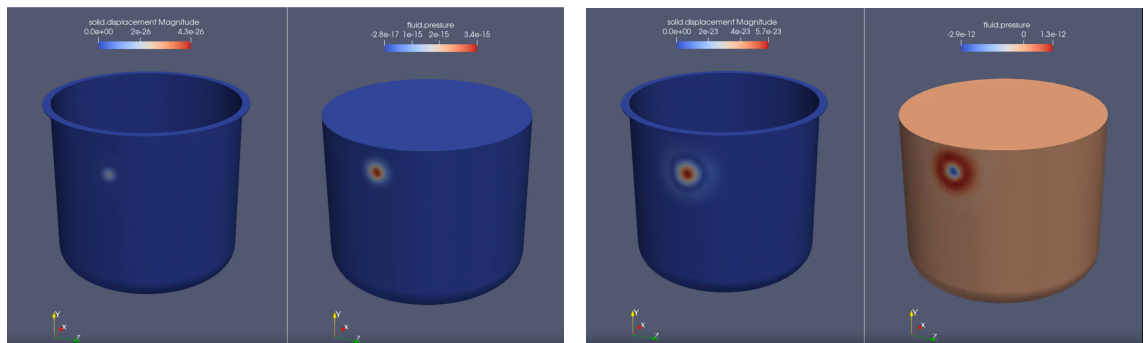


Figure 1: Visualisations 2D du couplage fluide-structure



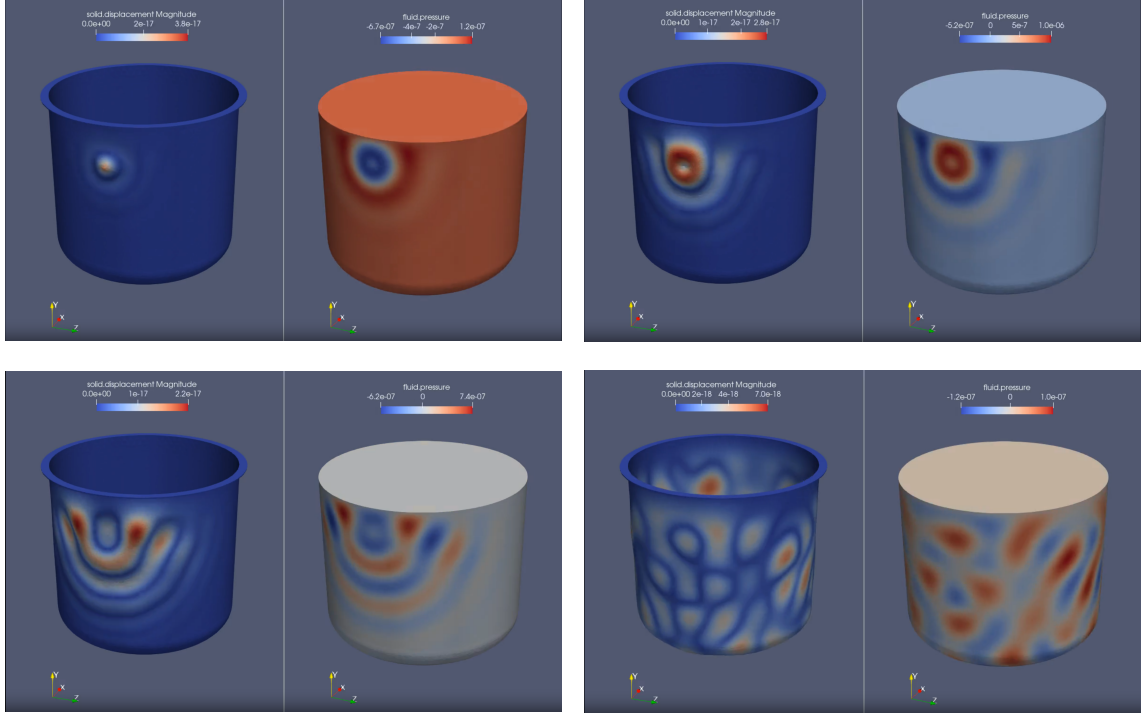


Figure 2: Visualisations 3D du couplage fluide-structure

9 Analysis of computation times

For our fluid-structure interaction simulations, we set the Picard iteration parameters to a maximum of 5 iterations and a machine tolerance of 10^{-16} . In practice, from the first iteration onwards, the difference between two successive iterations of the elasticity and wave equations is of the order of 10^{-10} . Thus, the number of Picard iterations remains low, generally 2 per time step and a maximum of 3 for longer time steps.

An important point to note is that we use explicit for the wave equation and implicit for elasticity. Explicit significantly improves the calculation time. However, the fluid domain is much larger than the solid domain, so the solution time remains higher for the wave equation.

For example, for a 2D simulation with a relatively coarse mesh of 12 mm, the calculation time for one time step on 128 cores is as follows:

```
[Solve Elasticity] Time : 0.015202 s
[Solve Wave] Time : 0.0163906 s
[Coupling: time per step] Time : 2.57761 s
```

We can see that the solution times for the wave equation and the elasticity equation are relatively close. This is because the wave equation is solved explicitly, but over a much larger number of elements (the solid wall is only 3 mm thick in a tank approximately 1 m wide and 0.8 m high).

The total calculation time for a time step for the coupling is much longer because it includes:

- two Picard iterations: two solutions of the wave equation and two solutions of the elasticity equation
- projection of the solution onto the other space to allow fluid-solid coupling
- export operations and updating of the next time step

For 3D simulations, the number of degrees of freedom increases significantly, which greatly increases the calculation time. In addition, mesh partitioning is necessary to parallelize across multiple cores. For a mesh refinement study, the following calculation times are obtained on 128 cores:

12 mm mesh:

[Solve Elasticity] Time : 0.1319 s
[Solve Wave] Time : 0.0651355 s
[Coupling: time per step] Time : 3.27349 s

6 mm mesh:

[Solve Elasticity] Time : 0.311 s
[Solve Wave] Time : 0.147052 s
[Coupling: time per step] Time : 5.31692 s

3 mm mesh:

[Solve Elasticity] Time : 0.621 s
[Solve Wave] Time : 0.482818 s
[Coupling: time per step] Time : 17.0882 s

Compared to 2D, the calculation time is much longer, which is related to the increase in degrees of freedom. We also note a much greater time saving with the explicit (wave equation) than in 2D.

A non-linear increase in calculation time can also be observed. When the mesh is refined, the number of elements increases, which explains the rapid increase in calculation time. The total coupling time increases much faster than the time for each individual solver, as it includes several resolutions, projections between spaces, export operations, and transition to the next time step.

Refinement improves numerical accuracy but increases computing time significantly (from 3 seconds to 17 seconds for meshes ranging from 12 mm to 3 mm). Therefore, it is necessary to find a compromise between resolution accuracy and simulation time.

Due to time constraints, we were unable to conduct a more in-depth study of performance. Nevertheless, a more detailed analysis in 2D and 3D could be carried out, including:

- tests for different mesh finenesses with graphical representation of simulation cost
- tests on multiple computing cores with a graph representing the computing cost as a function of the number of cores used
- comparison between explicit and implicit formulations (particularly for the wave equation, since the explicit formulation of elasticity is not yet functional)
- study of the effect of the order of finite elements and the order of quadrature on the gain in precision and calculation time (currently all elements are P1G1)

A complete benchmarking analysis could also be conducted to better understand the origin of computation times, evaluate the impact of increasing parameters for greater accuracy, and thus accurately document the effectiveness of fluid-structure coupling.

10 Conclusion

In conclusion, in this work we achieved our goal of developing a vibro-acoustic toolbox for treating fluid-structure problems, which can be used by Avnir Energy. This toolbox is easy to use: simply provide a .json file containing the correct parameters and sections, a .cfg file, and a .geo geometry file describing the fluid domain and the solid domain, and the simulation will start immediately.

The typical example studied is that of a hammer impact on the outer wall of the tank, allowing us to observe the system's response: the deformation and displacement of the solid wall, as well as the propagation of acoustic waves in the fluid, modeled by the pressure field.

For our test cases, we chose to use realistic physical parameters, with water as the fluid and stainless steel as the material for the solid. This choice corresponds to a simple and common example in the field of nuclear engineering.

The results obtained demonstrate the validity of the approach for fluid-structure coupling, but due to time constraints, we were unable to carry out all the planned analyses, in particular the performance study (benchmarking), comparing the numerical results with the FSI toolbox and the experimental results (Sonorch Technologies tank model), and the detailed evaluation of calculation times based on different numerical parameters.

In summary, this work constitutes a first step towards the tool sought by Avnir Energy, adapted to industrial needs and linked to the study of time reversal. The numerical foundations are in place, but several future improvements can be made to increase the speed, flexibility, and practicality of the tool.

11 Perspectives

This work thus constitutes a first basis for fluid-structure simulation. Since we built this toolbox from scratch, we have full access to the code, which opens the way to many possible developments:

- Explicit elasticity equation: allows for a significant reduction in computation time (particularly for 3D).
- Benchmarking studies: analysis of computation time in order to quantify the impact of numerical choices on computation time and accuracy.
- ALE map: addition of an ALE map to take mesh deformations into account. The next step will be to compare the results with and without ALE. The theoretical part has already been completed, as well as the beginning of the numerical part, which still needs to be tested.
- Docker: create a Docker image to simplify the installation of Feel++ and make the tool usable without compilation.
- More realistic boundary conditions.
- More realistic tank geometry.
- Exporting measurements at a point: work in progress.
- Analytical validation: known simple cases to verify the accuracy of the code.

12 Annex

To model mesh deformation, an ALE (Arbitrary Lagrangian-Eulerian) description is used. In practice, the solid is treated in a Lagrangian manner, i.e., each point of the solid is tracked directly as it moves, and the fluid is treated in ALE. This is the intermediate between Lagrangian (particle tracking) and Eulerian (fixed points in space) to manage the movement of the fluid around the solid without overly deforming the mesh.

12.1 Wave equation - With ALE method

12.1.1 Equation

$$\left\{ \begin{array}{l} \frac{\partial^2 p}{\partial t^2} + 2\dot{d}_f \cdot \nabla \left(\frac{\partial p}{\partial t} \right) + (\nabla \cdot \dot{d}_f) \frac{\partial p}{\partial t} - c^2 \Delta p = f_f \quad \text{in } \Omega_f \times [0, T] \\ d_f(t=0) = 0 \quad \text{in } \Omega_f \times \{0\} \\ p(t=0) = 0 \quad \text{in } \Omega_f \times \{0\} \\ d_f = u_s \quad \text{on } \Gamma_{fsi} \times [0, T] \\ p(x, y) = 0 \quad \text{on } \Gamma_{top} \times [0, T] \\ \frac{\partial p}{\partial n_f} = -\rho_f \left(\frac{\partial u_s}{\partial t} \cdot n_s \right) \quad \text{on } \Gamma_{fsi} \times [0, T] \end{array} \right.$$

With:

- p : pressure field (scalar, in Pa)
- d_f : mesh displacement (solution of the ALE equation, in m)
- $\dot{d}_f = \frac{\partial d_f}{\partial t}$: mesh displacement velocity (in m/s)
- u_s : solid displacement (solution to the elasticity equation, in m)
- c : sound velocity in the fluid (in m/s)
- f_f : source term (Ricker wavelet, in Pa/s²)
- Ω_f : spatial domain of the fluid (2D or 3D, in m² or m³)
- Γ_{top} : top of the tank, part of the edge of Ω_f (in m)
- Γ_{fsi} : edge between Ω_f and Ω_s (in m)
- $[0, T]$: time domain (in s)

12.1.2 ALE resolution

The ALE (Arbitrary Lagrangian-Eulerian) method is used to manage mesh deformation. The ALE displacement of the mesh is denoted by $d_f(x, t)$, which is a vector field.

ALE method equation

$$\left\{ \begin{array}{l} -\nabla \cdot (k d_f) = 0 \quad \text{in } \Omega_f \times [0, T] \\ d_f(x, 0) = 0 \quad \text{in } \Omega_f \times \{0\} \\ d_f = u_s \quad \text{on } \Gamma_{fsi} \times [0, T] \end{array} \right.$$

With:

- $d_f(x, t)$: ALE displacement of the mesh (vector field, in m)
- k : diffusion parameter

- u_s : displacement of the solid (solution of the elasticity equation, in m)
- Ω_f : spatial domain of the fluid (2D or 3D, in m^2 or m^3)
- Γ_{fsi} : boundary between Ω_f and Ω_s (solid domain, in m)
- $[0, T]$: time domain (in s)

12.1.3 Variational and matrix formulation of ALE

Let $d_f \in [H^1(\Omega_f)]^d$ and $w \in [H^1(\Omega_f)]^d$ ($d = 2$ or 3) be the test function with the same boundary conditions as d_f .

We multiply the ALE method equation by the test function and integrate over the domain Ω_f :

$$\int_{\Omega_f} -\nabla \cdot (k \nabla d_f) \cdot w \, d\Omega_f = 0$$

Reformulation of the second term using Green's formula:

$$\int_{\Omega_f} k \nabla d_f : \nabla w \, d\Omega_f = - \int_{\partial\Omega_f} k (\nabla d_f \cdot n_f) \cdot w \, d\partial\Omega_f$$

Considering Dirichlet boundary conditions: $d_f = u_s$ on Γ_{fsi} and $d_f = 0$ everywhere else, this gives us:

$$\int_{\Omega_f} k \nabla d_f : \nabla w \, d\Omega_f = 0$$

Note: Here we have taken all boundary conditions to be zero, but we must take into account and add the Dirichlet conditions $d_f = u_s$ in the implementation.

Temporal discretization: This equation is stationary, so no temporal discretization is necessary. However, the solution will still be updated and solved at each time step, since the boundary term u_s is dynamic.

Spatial discretization: Let $\{\varphi_j\}$ be the basis functions of $[H^1(\Omega_f)]^d$ ($d = 2$ or 3) and $w = \varphi_i$ the test functions. We write:

$$d_f(x, t) \approx \sum_j D_j(t) \varphi_j(x)$$

This gives us the following matrix formulation:

$$\sum_j K_{ij} D_j = 0$$

with $K_{ij} = \int_{\Omega_f} k \nabla \varphi_j : \nabla \varphi_i \, dx$, the associated stiffness matrix.

12.1.4 Variational formulation of the wave equation with ALE

Let $p \in H^1(\Omega_f)$ and $q \in H^1(\Omega_f)$ be a test function with the same boundary conditions as p .

We multiply by the test function and integrate over the domain Ω_f :

$$\int_{\Omega_f} \frac{\partial^2 p}{\partial t^2} q \, d\Omega_f + \int_{\Omega_f} 2 \dot{d}_f \cdot \nabla \left(\frac{\partial p}{\partial t} \right) q + (\nabla \cdot \dot{d}_f) \frac{\partial p}{\partial t} q \, d\Omega_f - \int_{\Omega_f} c^2 (\Delta p) q \, d\Omega_f = \int_{\Omega_f} f_f q \, d\Omega_f$$

Reformulation of the second term using Green's formula:

$$\int_{\Omega_f} \frac{\partial^2 p}{\partial t^2} q \, d\Omega_f + \int_{\Omega_f} 2 \dot{d}_f \cdot \nabla \left(\frac{\partial p}{\partial t} \right) q + (\nabla \cdot \dot{d}_f) \frac{\partial p}{\partial t} q \, d\Omega_f + \int_{\Omega_f} c^2 \nabla p \cdot \nabla q \, d\Omega_f = \int_{\Omega_f} f_f q \, d\Omega_f + \int_{\partial\Omega_f} c^2 \frac{\partial p}{\partial n_f} q \, d\partial\Omega_f$$

Considering the coupling boundary conditions: $\frac{\partial p}{\partial n_f} = -\rho_f \frac{\partial u_s}{\partial t} \cdot n_s$ and $d_f = u_s$ on Γ_{fsi} and Dirichlet conditions $p = 0$ everywhere else, this gives us:

$$\begin{aligned} \int_{\Omega_f} \frac{\partial^2 p}{\partial t^2} q \, d\Omega_f + \int_{\Omega_f} 2\dot{d}_f \cdot \nabla \left(\frac{\partial p}{\partial t} \right) q + (\nabla \cdot \dot{d}_f) \frac{\partial p}{\partial t} q \, d\Omega_f + \int_{\Omega_f} c^2 \nabla p \cdot \nabla q \, d\Omega_f = \int_{\Omega_f} f_f q \, d\Omega_f \\ - \int_{\Gamma_{fsi}} c^2 \rho_f \left(\frac{\partial u_s}{\partial t} \cdot n_s \right) q \, d\Gamma_{fsi} \end{aligned}$$

12.1.5 Temporal discretization

We consider a centered bdf scheme to discretize the terms $\frac{\partial p^{n+1}}{\partial t}$, $\frac{\partial^2 p^{n+1}}{\partial t^2}$ and $\dot{d}_f^{n+1} = \frac{\partial d_f^{n+1}}{\partial t}$ in time, and, for the boundary conditions, an approximation of the solid velocity for the term $\frac{\partial u_s^{n+1}}{\partial t}$ using a Newmark scheme. Thus, we have the following approximations:

$$\begin{aligned} \frac{\partial p^{n+1}}{\partial t} &\approx \frac{p^{n+1} - p^n}{\Delta t}, \quad \frac{\partial^2 p^{n+1}}{\partial t^2} \approx \frac{p^{n+1} - 2p^n + p^{n-1}}{\Delta t^2}, \\ \dot{d}_f^{n+1} &= \frac{\partial d_f^{n+1}}{\partial t} \approx \frac{d_f^{n+1} - d_f^n}{\Delta t} \quad \text{et} \quad \frac{\partial u_s^{n+1}}{\partial t} \approx v_s^{n+1} = v_s^n + \Delta t[(1 - \gamma)a_s^n + \gamma a_s^{n+1}] \end{aligned}$$

where p^{n+1} is the fluid pressure at time $n+1$ and p^n at time n (stored variable from the previous time), d_f^{n+1} is the displacement of the mesh at time $n+1$ and d_f^n at time n (stored variables), v_s^n is the velocity of the solid at time n (stored variable) and a_s^{n+1} is the acceleration of the solid at time $n+1$.

So, by discretizing the variational formulation, we get:

$$\begin{aligned} \int_{\Omega_f} \frac{p^{n+1} - 2p^n + p^{n-1}}{\Delta t^2} q \, d\Omega_f + \int_{\Omega_f} 2\dot{d}_f^{n+1} \cdot \nabla \left(\frac{p^{n+1} - p^n}{\Delta t} \right) q + (\nabla \cdot \dot{d}_f^{n+1}) \left(\frac{p^{n+1} - p^n}{\Delta t} \right) q \, d\Omega_f \\ + \int_{\Omega_f} c^2 \nabla p^n \cdot \nabla q \, d\Omega_f = \int_{\Omega_f} f_f^n q \, d\Omega_f - \int_{\Gamma_{fsi}} c^2 \rho_f (v_s^{n+1} \cdot n_s) q \, d\Gamma_{fsi} \end{aligned}$$

And isolating the term p^{n+1} :

$$\begin{aligned} \int_{\Omega_f} \left[\frac{1}{\Delta t^2} + \frac{\nabla \cdot \dot{d}_f^{n+1}}{\Delta t} \right] p^{n+1} q + \frac{2}{\Delta t} \dot{d}_f^{n+1} \cdot (\nabla p^{n+1}) q \, d\Omega_f = \int_{\Omega_f} \frac{2p^n - p^{n-1}}{\Delta t^2} q + \int_{\Omega_f} \frac{2}{\Delta t} \dot{d}_f^{n+1} \cdot (\nabla p^n) q \, d\Omega_f \\ + (\nabla \cdot \dot{d}_f^{n+1}) \frac{p^n}{\Delta t} q \, d\Omega_f - c^2 \int_{\Omega_f} \nabla p^n \cdot \nabla q \, d\Omega_f + \int_{\Omega_f} f_f^n q \, d\Omega_f - \int_{\Gamma_{fsi}} c^2 \rho_f (v_s^{n+1} \cdot n_s) q \, d\Gamma_{fsi} \end{aligned}$$

12.1.6 Spatial discretization and matrix formulation

Let $\{\psi_j\}$ be the basis functions of $H^1(\Omega_f)$ and $\{\varphi_j\}$ the basis functions of $[H^1(\Omega_f)]^d$ ($d = 2$ or 3). We set $q = \psi_i$ as the test function.

The spatial discretization is written as:

$$p(x, t) \approx \sum_j P_j(t) \psi_j(x) \quad \text{et} \quad \dot{d}_f(x, t) \approx \sum_j \dot{D}_j(t) \varphi_j(x)$$

This gives us the following matrix formulation:

$$(\frac{1}{\Delta t^2}\mathbf{M} + \frac{1}{\Delta t}\mathbf{C} + \frac{2}{\Delta t}\mathbf{N})p^{n+1} = \frac{2}{\Delta t^2}\mathbf{M}p^n - \frac{1}{\Delta t^2}\mathbf{M}p^{n-1} + \frac{2}{\Delta t}\mathbf{N}p^n + \frac{1}{\Delta t}\mathbf{C}p^n - c^2\mathbf{K}p^n + \mathbf{f}^n - \mathbf{v}_{bc}^{n+1}$$

With:

- $\mathbf{M}_{ij} = \int_{\Omega_f} \psi_i \psi_j dx$: mass matrix
- $\mathbf{C}_{ij} = \int_{\Omega_f} (\nabla \cdot \dot{d}_f^{n+1}) \psi_i \psi_j dx$: ALE dilation matrix
- $\mathbf{N}_{ij} = \int_{\Omega_f} (\dot{d}_f^{n+1} \cdot \nabla \psi_j) \psi_i dx$: ALE convection matrix
- $\mathbf{K}_{ij} = \int_{\Omega_f} \nabla \psi_j \cdot \nabla \psi_i dx$: stiffness matrix
- $\mathbf{f}_i^n = \int_{\Omega_f} f_f^n \psi_i dx$: source term for the fluid at time n
- $\mathbf{v}_{bc,i}^{n+1} = \int_{\Gamma_{fsi}} c^2 \rho_f (v_s^{n+1} \cdot n_s) \psi_i ds$: Neumann boundary condition on Γ_{fsi}