

# Hybrid 3D-2D Modeling of Wave Problems

October 4, 2021

Ron Efrati 322806480

Adviser: Prof. Dan Givoli

## 1 Introduction

In many engineering disciplines numerical simulations with a high number of degrees of freedom are often necessary. Mixed-dimensional modeling can decrease the number of degrees of freedom dramatically without harming the desired mesh resolution.

If the discussed problem consists of a 3D domain in which the solution is expected to be actually 2D, namely the solution does not depend on one of the coordinates, it is a waste to use a high resolution 3D computational mesh in this domain. Instead, one can attempt to decrease the number of nodes in a mesh by modeling a part of the domain as two-dimensional.

The dependency of the solution on one of the coordinates can be weakened mainly by the geometry of the problem and the boundary conditions.

This report will focus on elastodynamics problems. The main objective of this report is to analyze under what circumstances the solution of the hybrid problem is a good approximation of the full problem's solution using the Panasenko coupling method, considering coupling of 3D and 2D sub-models.

## 2 Statement of the problem and solution scheme

### 2.1 Full 3D problem

Consider a three-dimensional structure made of linear isotropic material. We are interested in finding the time-accurate displacements field due to a non-zero initial state or inhomogeneous tractions boundary conditions.

#### 2.1.1 The strong form

The governing equations of an undamped structure denoted  $\Omega$ , written with :

$$\rho \ddot{u}_i = \sigma_{ij,j} + f_i, \quad \text{in } \Omega \quad (1)$$

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl}, \quad \text{in } \Omega \quad (2)$$

$$\varepsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}), \quad \text{in } \Omega \quad (3)$$

$\rho$  is the material density,  $u_i(\mathbf{x}, t)$  are the displacements,  $f_i$  are body forces and  $C_{ijkl}$  is the material properties tensor. For an isotropic material,  $C_{ijkl}$  is given by:

$$C_{ijkl} = \delta_{ij}\delta_{kl}\lambda + G(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \quad (4)$$

where  $\lambda, G$  are Lamé's parameters of the material and  $\delta_{ij}$  is the Kronecker delta. The boundary conditions are given by:

$$u_i = \bar{u}_i, \quad \text{on } \Gamma_{u_i} \quad (5)$$

$$\sigma_{ij}n_j = \bar{T}_i, \quad \text{on } \Gamma_{T_i} \quad (6)$$

$\bar{u}_i(\mathbf{x}, t)$  and  $\bar{T}_i(\mathbf{x}, t)$  are the given displacements and traction vectors on the boundary of the domain  $\Omega$ .  $\Gamma_{u_i}$  is the part of  $\partial\Omega$  in which the displacements in the direction of  $x_i$  are known and  $\Gamma_{T_i}$  is the part of  $\partial\Omega$  in which the tractions in the direction of  $x_i$  are known, such that for each  $i$  independently:

$$\partial\Omega = \Gamma_{T_i} \cup \Gamma_{u_i} \quad (7)$$

$N_{sd}$  is the number of space dimensions: 3 for a 3D problem and 2 for a 2D problem.  $n_j$  is a unit vector normal to  $\Gamma_{T_i}$ .

The initial conditions are written as:

$$u_i(\mathbf{x}, 0) = U_{0_i}(\mathbf{x}), \quad \text{in } \Omega \quad (8)$$

$$\dot{u}_i(\mathbf{x}, 0) = V_{0_i}(\mathbf{x}), \quad \text{in } \Omega \quad (9)$$

### 2.1.2 The weak form

Let us multiply Eq. 1 by a weighting function  $w_i(\mathbf{x})$  and integrate over  $\Omega$ :

$$\int_{\Omega} \rho \ddot{u}_i w_i d\Omega = \int_{\Omega} \sigma_{ij,j} w_i d\Omega + \int_{\Omega} f_i w_i d\Omega \quad (10)$$

Using the chain rule and the divergence theorem we can write:

$$\begin{aligned} \int_{\Omega} \sigma_{ij,j} w_i d\Omega &= \int_{\Omega} (\sigma_{ij} w_i)_{,j} d\Omega - \int_{\Omega} \sigma_{ij} w_{i,j} d\Omega \\ \int_{\Omega} \sigma_{ij,j} w_i d\Omega &= \int_{\partial\Omega} \sigma_{ij} w_i n_j d\Omega - \int_{\Omega} \sigma_{ij} w_{i,j} d\Omega \end{aligned} \quad (11)$$

Plug Eq. 11 into Eq. 10:

$$\int_{\Omega} \rho \ddot{u}_i w_i d\Omega = \int_{\partial\Omega} \sigma_{ij} w_i n_j d\Omega - \int_{\Omega} \sigma_{ij} w_{i,j} d\Omega + \int_{\Omega} f_i w_i d\Omega \quad (12)$$

Using Eq. 7:

$$\int_{\partial\Omega} \sigma_{ij} w_i n_j d\Omega = \int_{\Gamma_{T_i}} \sigma_{ij} w_i n_j d\Gamma + \int_{\Gamma_{u_i}} \sigma_{ij} w_i n_j d\Gamma$$

From Eq. 6:

$$\int_{\partial\Omega} \sigma_{ij} w_i n_j d\Omega = \int_{\Gamma_{T_i}} \bar{T}_i w_i d\Gamma + \int_{\Gamma_{u_i}} \sigma_{ij} w_i n_j d\Gamma$$

Let us choose  $w_i$  such that  $w_i = 0$  on  $\Gamma_{u_i}$ :

$$\int_{\partial\Omega} \sigma_{ij} w_i n_j d\Omega = \int_{\Gamma_{T_i}} \bar{T}_i w_i d\Gamma$$

Substitute into Eq. 12 to obtain:

$$\int_{\Omega} \rho \ddot{u}_i w_i d\Omega + \int_{\Omega} \sigma_{ij} w_{i,j} d\Omega = \int_{\Gamma_{T_i}} \bar{T}_i w_i d\Gamma + \int_{\Omega} f_i w_i d\Omega$$

Using Eq. 2, Eq. 3 and the symmetry of  $C_{ijkl}$ :

$$C_{ijkl} = C_{ijlk}$$

$$\int_{\Omega} \rho \ddot{u}_i w_i d\Omega + \int_{\Omega} C_{ijkl} u_{k,l} w_{i,j} d\Omega = \int_{\Gamma_{T_i}} \bar{T}_i w_i d\Gamma + \int_{\Omega} f_i w_i d\Omega$$

Therefore, the weak form formulation:

Find  $\mathbf{u} \in S$  such that  $u_i(\mathbf{x}, 0) = U_{0_i}(\mathbf{x})$ ,  $\dot{u}_i(\mathbf{x}, 0) = V_{0_i}(\mathbf{x})$  and for all  $\mathbf{w} \in S_0$ :

$$b(\mathbf{w}, \ddot{\mathbf{u}}) + a(\mathbf{w}, \mathbf{u}) = l(\mathbf{w}) \quad (13)$$

Where:

$$S = \{ \mathbf{u} | \mathbf{u} \in C^0, u_i = \bar{u}_i \quad \text{on } \Gamma_{u_i} \}$$

$$S_0 = \{ \mathbf{w} | \mathbf{w} \in C^0, w_i = 0 \quad \text{on } \Gamma_{u_i} \}$$

$$b(\mathbf{w}, \mathbf{u}) = \int_{\Omega} \rho u_i w_i d\Omega$$

$$a(\mathbf{w}, \mathbf{u}) = \int_{\Omega} C_{ijkl} u_{k,l} w_{i,j} d\Omega$$

$$l(\mathbf{w}) = \int_{\Gamma_{T_i}} \bar{T}_i w_i d\Gamma + \int_{\Omega} f_i w_i d\Omega$$

$a(\mathbf{w}, \mathbf{u})$ ,  $b(\mathbf{w}, \mathbf{u})$  are symmetric and bi-linear, and  $l(\mathbf{w})$  is linear.

### 2.1.3 Galerkin FE approximation

Let us approximate the functions  $u_i, w_i$  by a linear combination of shape functions  $\{\phi_A(\mathbf{x})\}$ :

$$u_i(\mathbf{x}, t) \approx u_i^h(\mathbf{x}, t) = \sum_{A \in \hat{\eta}} \hat{d}_{(Ai)}(t) \phi_A(\mathbf{x}) = \sum_{A \in \eta_i} d_{(Ai)}(t) \phi_A(\mathbf{x}) + \sum_{A \in \eta_{u_i}} \bar{u}_{(Ai)}(t) \phi_A(\mathbf{x})$$

$$w_i(\mathbf{x}) \approx w_i^h(\mathbf{x}) = \sum_{A \in \eta_i} C_{(Ai)} \phi_A(\mathbf{x})$$

$$\mathbf{u}^h = u_i^h e_i, \quad \mathbf{w}^h = w_i^h e_i$$

Where  $e_i$  is a unit vector in the direction of  $x_i$ , and:

$$\eta_i = \{\text{set of all nodes that are open (with unknown displacements) in direction } i\}$$

$$\eta_{ui} = \{\text{set of all nodes that are closed (with known displacements) in direction } i\}$$

$$\hat{\eta} = \{\text{set of all nodes}\}$$

#### 2.1.4 The semi-discrete problem

Plug the approximation to the weak form (Eq. 13):

$$b(\mathbf{w}^h, \ddot{\mathbf{u}}^h) + a(\mathbf{w}^h, \mathbf{u}^h) = l(\mathbf{w}^h)$$

$$b(w_i^h e_i, u_j^h e_j) + a(w_i^h e_i, u_j^h e_j) = l(w_i^h e_i)$$

$$b\left(\sum_{A \in \eta_i} C_{(Ai)} \phi_{Ae_i}, \sum_{B \in \hat{\eta}} \ddot{d}_{(Bj)} \phi_{Be_j}\right) + a\left(\sum_{A \in \eta_i} C_{(Ai)} \phi_{Ae_i}, \sum_{B \in \hat{\eta}} \hat{d}_{(Bj)} \phi_{Be_j}\right) = l\left(\sum_{A \in \eta_i} C_{(Ai)} \phi_{Ae_i}\right)$$

Using the linearity of  $a(\mathbf{w}, \mathbf{u})$ ,  $b(\mathbf{w}, \mathbf{u})$ ,  $l(\mathbf{w})$ :

$$\sum_{A \in \eta_i} C_{(Ai)} \left[ b\left(\phi_{Ae_i}, \sum_{B \in \hat{\eta}} \ddot{d}_{(Bj)} \phi_{Be_j}\right) + a\left(\phi_{Ae_i}, \sum_{B \in \hat{\eta}} \hat{d}_{(Bj)} \phi_{Be_j}\right) - l(\phi_{Ae_i}) \right] = 0$$

Since the weak form should be satisfied for all  $\mathbf{w} \in S_0$ ,  $C_{(Ai)}$  are linealy independent:

$$b\left(\phi_{Ae_i}, \sum_{B \in \hat{\eta}} \ddot{d}_{(Bj)} \phi_{Be_j}\right) + a\left(\phi_{Ae_i}, \sum_{B \in \hat{\eta}} \hat{d}_{(Bj)} \phi_{Be_j}\right) - l(\phi_{Ae_i}) = 0, \quad A \in \eta_i, i = 1, \dots, N_{sd}$$

Using the linearity of  $a(\mathbf{w}, \mathbf{u})$ ,  $b(\mathbf{w}, \mathbf{u})$ :

$$\begin{aligned} \sum_{B \in \hat{\eta}} \left[ b(\phi_{Ae_i}, \phi_{Be_j}) \ddot{d}_{(Bj)}(t) + a(\phi_{Ae_i}, \phi_{Be_j}) \hat{d}_{(Bj)} \right] &= l(\phi_{Ae_i}), \quad A \in \eta_i, i = 1, \dots, N_{sd} \\ \sum_{j=1}^{N_{sd}} \sum_{B \in \eta_j} \left[ b(\phi_{Ae_i}, \phi_{Be_j}) \ddot{d}_{(Bj)} + a(\phi_{Ae_i}, \phi_{Be_j}) \hat{d}_{(Bj)} \right] &= F_{(Ai)}, \quad A \in \eta_i, i = 1, \dots, N_{sd} \end{aligned} \quad (14)$$

Where:

$$F_{(Ai)} = l(\phi_{Ae_i}) - \sum_{j=1}^{N_{sd}} \sum_{B \in \eta_{uj}} \left[ b(\phi_{Ae_i}, \phi_{Be_j}) \ddot{u}_{(Bj)} + a(\phi_{Ae_i}, \phi_{Be_j}) \bar{u}_{(Bj)} \right]$$

$$\bar{u}_{(Bj)} = \bar{u}_j(\mathbf{x}_B)$$

Eq. 14 can be written in matrix form as:

$$\mathbf{M} \cdot \ddot{\mathbf{d}} + \mathbf{K} \cdot \mathbf{d} = \mathbf{F}$$

Where:

$$\mathbf{d} = \{d_{(Ai)}\}$$

$$\mathbf{M} = [M_{(Ai)(Bj)}] = [b(\phi_A e_i, \phi_B e_j)] = \left[ \int_{\Omega} \rho \phi_B e_j \phi_A e_i d\Omega \right] = \left[ \delta_{ij} \int_{\Omega} \rho \phi_B \phi_A d\Omega \right]$$

$$\mathbf{K} = [K_{(Ai)(Bj)}] = [a(\phi_A e_i, \phi_B e_j)] = \left[ \int_{\Omega} \phi_{A,k} C_{ikjl} \phi_{B,l} d\Omega \right]$$

$$\mathbf{F} = \{F_{(Ai)}\} = \left\{ \int_{\Gamma_{T_i}} \bar{T}_i \phi_A e_i d\Gamma + \int_{\Omega} f_i \phi_A e_i d\Omega - \sum_{j=1}^{N_{sd}} \sum_{B \in \eta_{u_j}} [M_{(Ai)(Bj)} \ddot{u}_{(Bj)} + K_{(Ai)(Bj)} \bar{u}_{(Bj)}] \right\}$$

$\mathbf{M}$  is the global mass matrix,  $\mathbf{K}$  is the global stiffness matrix and  $\mathbf{F}$  is the global force vector. The initial conditions of the system of equations above can be derived from Eq. 8 and Eq. 9:

$$d_{(Ai)}(0) = U_{0_i}(\mathbf{x}_A), \quad \dot{d}_{(Ai)}(0) = V_{0_i}(\mathbf{x}_A)$$

Thus the semi-discrete problem formulation:

$$\begin{cases} \mathbf{M} \cdot \ddot{\mathbf{d}} + \mathbf{K} \cdot \mathbf{d} & = \mathbf{F} \\ \mathbf{d}(0) & = \mathbf{d}_0 \\ \dot{\mathbf{d}}(0) & = \mathbf{v}_0 \end{cases} \quad (15)$$

$$\mathbf{M} = [M_{(Ai)(Bj)}], \quad \mathbf{K} = [K_{(Ai)(Bj)}], \quad \mathbf{F} = \{F_{(Ai)}\}, \quad \mathbf{d} = \{d_{(Ai)}\}$$

$$\mathbf{d}_0 = \{d_{0(Ai)}\} = \{U_{0_i}(\mathbf{x}_A)\}, \quad \mathbf{v}_0 = \{v_{0(Ai)}\} = \{V_{0_i}(\mathbf{x}_A)\}$$

$$M_{(Ai)(Bj)} = \delta_{ij} \int_{\Omega} \rho \phi_B \phi_A d\Omega, \quad K_{(Ai)(Bj)} = \int_{\Omega} \phi_{A,k} C_{ikjl} \phi_{B,l} d\Omega$$

$$F_{(Ai)} = \int_{\Gamma_{T_i}} \bar{T}_i \phi_A e_i d\Gamma + \int_{\Omega} f_i \phi_A e_i d\Omega - \sum_{j=1}^{N_{sd}} \sum_{B \in \eta_{u_j}} [M_{(Ai)(Bj)} \ddot{u}_{(Bj)} + K_{(Ai)(Bj)} \bar{u}_{(Bj)}]$$

$$A \in \eta_i, \quad B \in \eta_j, \quad i, j \in [1, \dots, N_{sd}]$$

### 2.1.5 The element level

The global matrices and vectors will be calculated by the assembly of the element level contributions. Let us define the element level mass and stiffness matrices:

$$m_{(ai)(bj)}^e = \delta_{ij} \int_{\Omega^e} \rho \phi_a \phi_b d\Omega, \quad k_{(ai)(bj)}^e = \int_{\Omega^e} \phi_{a,k} C_{ikjl} \phi_{b,l} d\Omega \quad (16)$$

Where  $\Omega^e$  is the element domain and  $a, b$  are element level indices. The element level force vector:

$$f_{(ai)}^e = \int_{\Gamma_{T_i}^e} \bar{T}_i \phi_a e_i d\Gamma + \int_{\Omega^e} f_i \phi_a e_i d\Omega - \sum_{j=1}^{N_{sd}} \sum_{b \in \eta_{u_j}} \left[ m_{(ai)(bj)}^e \ddot{u}_{(bj)}^e + k_{(ai)(bj)}^e \bar{u}_{(bj)}^e \right] \quad (17)$$

A uniform mesh will be used, with hexahedral tri-linear elements for the 3D domain and rectangular bi-linear elements for the 2D domain. Specifically, cubic and squared elements will be used. Consider the hexahedral element in figure 1a. In order to solve the integrals in Eq. 16, one can transform the domain  $\Omega^e$  in figure 1a to the domain  $\Omega_\xi$  in figure 1b using the transformation:

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{a=1}^8 \phi_a(\boldsymbol{\xi}) \mathbf{x}_a^e \quad (18)$$

Where  $\mathbf{x}_a^e$  are the coordinates of node number  $a$  in the domain  $\Omega^e$ , and:

$$\phi_a(\boldsymbol{\xi}) = \frac{1}{8} (1 + \xi_a \xi) (1 + \eta_a \eta) (1 + \zeta_a \zeta) \quad (19)$$

| $a$ | $\xi_a$ | $\eta_a$ | $\zeta_a$ |
|-----|---------|----------|-----------|
| 1   | -1      | -1       | -1        |
| 2   | 1       | -1       | -1        |
| 3   | 1       | 1        | -1        |
| 4   | -1      | 1        | -1        |
| 5   | -1      | -1       | 1         |
| 6   | 1       | -1       | 1         |
| 7   | 1       | 1        | 1         |
| 8   | -1      | 1        | 1         |

Table 1: Coordinates of the nodes in the 3D transformed element.

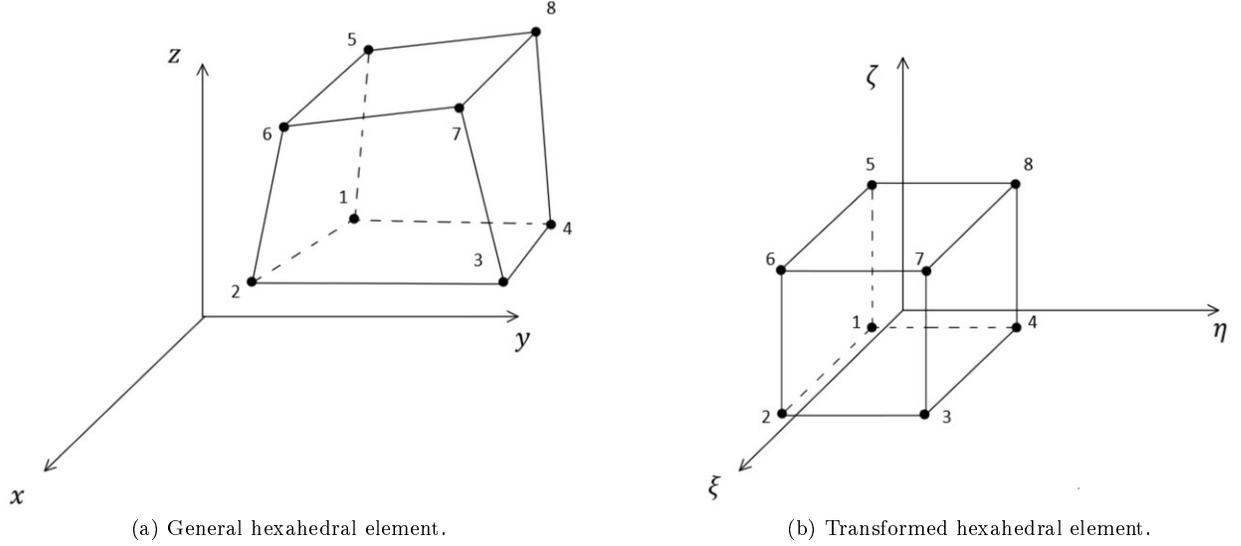


Figure 1: 3D element.

Apply the transformation to Eq. 16:

$$m_{(ai)(bj)}^e = \delta_{ij} \int_{\Omega^e} \rho \phi_a \phi_b d\Omega = \delta_{ij} \int_{\Omega_\xi^e} \rho \phi_a(\boldsymbol{\xi}) \phi_b(\boldsymbol{\xi}) |J| d\Omega \quad (20)$$

$|J|$  is the transformation Jacobian, defined as:

$$|J| = \left| \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right|$$

One can find with the help of a symbolic calculator that for a cubic element with a volume of  $h^3$ :

$$|J| = \frac{h^3}{8} \quad (21)$$

It is the ratio of the volumes of the two elements in Figure 1. Substitute Eq. 21 and Eq. 19 into Eq. 20:

$$m_{(ai)(bj)_{3D}}^e = \frac{\rho \delta_{ij}}{8^2} \int_{-1}^1 (1 + \xi_a \xi) (1 + \xi_b \xi) d\xi \int_{-1}^1 (1 + \eta_a \eta) (1 + \eta_b \eta) d\eta \int_{-1}^1 (1 + \zeta_a \zeta) (1 + \zeta_b \zeta) d\zeta |J|$$

$$m_{(ai)(bj)_{3D}}^e = \frac{\rho \delta_{ij}}{8^2} \left( \int_{-1}^1 (1 + \xi_a \xi) (1 + \xi_b \xi) d\xi \right)^3 \frac{h^3}{8} = \frac{\delta_{ij} \rho h^3}{64} \left( 1 + \frac{\xi_a \xi_b}{3} \right) \left( 1 + \frac{\eta_a \eta_b}{3} \right) \left( 1 + \frac{\zeta_a \zeta_b}{3} \right)$$

For the element level stiffness matrix, plug Eq. 4 into Eq. 16:

$$k_{(ai)(bj)}^e = \int_{\Omega^e} \phi_{a,k} (\delta_{ik} \delta_{jl} \lambda + G (\delta_{ij} \delta_{kl} + \delta_{il} \delta_{kj})) \phi_{b,l} d\Omega$$

Apply the transformation:

$$k_{(ai)(bj)}^e = \int_{\Omega_\xi^e} \phi_{a,k}(\boldsymbol{\xi}) (\delta_{ik} \delta_{jl} \lambda + G (\delta_{ij} \delta_{kl} + \delta_{il} \delta_{kj})) \phi_{b,l}(\boldsymbol{\xi}) |J| d\Omega$$

$$k_{(ai)(bj)}^e = \frac{h^3}{8} \left[ \lambda \int_{\Omega_\xi^e} \phi_{a,i}(\boldsymbol{\xi}) \phi_{b,j}(\boldsymbol{\xi}) d\Omega + G \delta_{ij} \int_{\Omega_\xi^e} \phi_{a,k}(\boldsymbol{\xi}) \phi_{b,k}(\boldsymbol{\xi}) d\Omega + G \int_{\Omega_\xi^e} \phi_{a,j}(\boldsymbol{\xi}) \phi_{b,i}(\boldsymbol{\xi}) d\Omega \right] \quad (22)$$

Calculate the shape functions derivatives  $\phi_{a,i}$  by:

$$\phi_{a,i}(\boldsymbol{\xi}) = \frac{\partial \phi_a(\boldsymbol{\xi})}{\partial x_i} = \frac{\partial \phi_a}{\partial \xi_j} \frac{\partial \xi_j}{\partial x_i} = \frac{\partial \phi_a}{\partial \xi_j} (J^{-1})_{ji} \quad (23)$$

$J^{-1}$  is the inverse of the transformation Jacobian matrix:

$$J^{-1} = \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}$$

$$(J^{-1})_{ij} = \frac{\partial \xi_i}{\partial x_j}$$

where the Jacobian matrix is calculated by:

$$J_{ij} = \frac{\partial x_i}{\partial \xi_j} = \sum_{a=1}^8 \frac{\partial \phi_a(\boldsymbol{\xi})}{\partial \xi_j} x_{a_i}^e \quad (24)$$

The calculations are done with the help of a symbolic calculator. As for the element level force vector, it is zero for all cases in this report since we discuss cases with:

$$\bar{u}_i = 0 \quad \forall \mathbf{x} \in \Gamma_{u_i}, \quad f_i \equiv 0, \quad \bar{T}_i = 0 \quad \forall \mathbf{x} \in \Gamma_{T_i}$$

The 2D element level matrices are calculated the same way. Consider the 2D original and transformed element in figure 2. The transformation between figure 2a and 2b is given by:

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{a=1}^4 \phi_a(\boldsymbol{\xi}) \mathbf{x}_a^e \quad (25)$$

$$\phi_a(\boldsymbol{\xi}) = \frac{1}{4} (1 + \xi_a \xi) (1 + \eta_a \eta) \quad (26)$$

Where the coordinates of the transformed element:

| $a$ | $\xi_a$ | $\eta_a$ |
|-----|---------|----------|
| 1   | -1      | -1       |
| 2   | 1       | -1       |
| 3   | 1       | 1        |
| 4   | -1      | 1        |

Table 2: Coordinates of the nodes in the 2D transformed element.

For a squared element with an area of  $h^2$ :

$$|J| = \frac{h^2}{4} \quad (27)$$

Similar to the 3D case, it is the ratio of areas of the elements in Figure 2. The 2D element level matrices are calculated by plugging Eq. 25, 26, 27 into Eq. 20, 22, 23, 24.

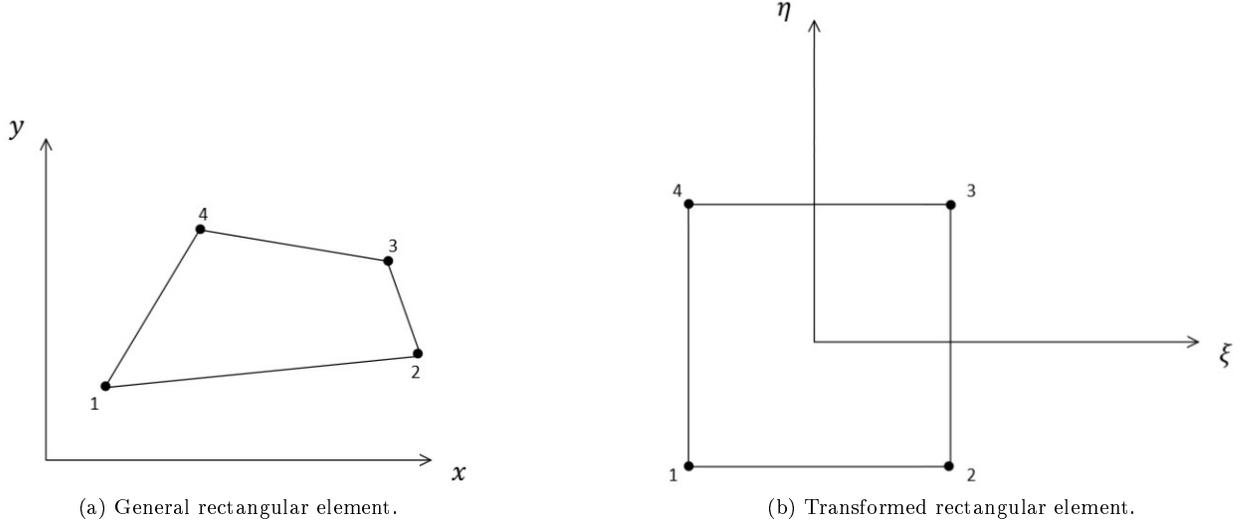


Figure 2: 2D element.

After obtaining the element level matrices, the global matrices are assembled by the following algorithm:

- Initialize  $M, K$
- Loop over all elements  $e = 1 \rightarrow N_{el}$ 
  - Loop over all nodes in the element  $a = 1 \rightarrow N_{en}$ 
    - Loop over all nodal DOF  $i = 1 \rightarrow N_{sd}$ 
      - Get unknown number  $p$
      - Loop over all nodes in the element  $b = 1 \rightarrow N_{en}$ 
        - Loop over all nodal DOF  $j = 1 \rightarrow N_{sd}$ 
          - Get unknown number  $q$
          - if  $q, p \neq 0$ 
            - $M(p, q) = M(p, q) + m_{(ai)(bj)}^e$
            - $K(p, q) = K(p, q) + k_{(ai)(bj)}^e$
          - Close if
        - Close  $j$  loop
      - Close  $b$  loop
    - Close  $i$  loop
  - Close  $a$  loop
- Close  $e$  loop

$N_{el}$  is the number of elements in the computational mesh and  $N_{en}$  is the number of nodes in an element.

Remark: In elastodynamics problems, the number of spatial dimensions  $N_{sd}$  equals the number of nodal degrees of freedom  $N_{ndof}$ . Therefore, while they are interchangeable in this report, they might not be when discussing other types of problem, such as fluid flow.

Let us denote the assembly operation of the global matrix or vector  $\mathbf{M}$  from the element level matrix/vector  $\mathbf{m}^e$  by  $\mathbf{M} = \mathcal{A}_{e=1}^{N_{el}} \{\mathbf{m}^e\}$ . The semi-discrete problem formulation, with the assembly operation is therefore:

$$\begin{cases} \mathbf{M} \cdot \ddot{\mathbf{d}} + \mathbf{K} \cdot \mathbf{d} & = \mathbf{F} \\ \mathbf{d}(0) & = \mathbf{d}_0 \\ \dot{\mathbf{d}}(0) & = \mathbf{v}_0 \end{cases}$$

$$\mathbf{M} = [M_{(Ai)(Bj)}], \quad \mathbf{K} = [K_{(Ai)(Bj)}], \quad \mathbf{F} = \{F_{(Ai)}\}, \quad \mathbf{d} = \{d_{(Ai)}\}$$

$$\mathbf{d}_0 = \{d_{0(Ai)}\} = \{U_{0i}(\mathbf{x}_A)\}, \quad \mathbf{v}_0 = \{v_{0(Ai)}\} = \{V_{0i}(\mathbf{x}_A)\}$$

$$\mathbf{M} = \mathcal{A}_{e=1}^{N_{el}} \{\mathbf{m}^e\}, \quad \mathbf{K} = \mathcal{A}_{e=1}^{N_{el}} \{\mathbf{k}^e\}, \quad \mathbf{F} = \mathcal{A}_{e=1}^{N_{el}} \{\mathbf{f}^e\}$$

$$\mathbf{m}^e = [m_{(ai)(bj)}^e], \quad \mathbf{k}^e = [k_{(ai)(bj)}^e], \quad \mathbf{f}^e = \{f_{(ai)}^e\}$$

$$m_{(ai)(bj)}^e = \delta_{ij} \int_{\Omega^e} \rho \phi_a \phi_b d\Omega, \quad k_{(ai)(bj)}^e = \int_{\Omega^e} \phi_{a,k} C_{ikjl} \phi_{b,l} d\Omega$$

$$f_{(ai)}^e = \int_{\Gamma_{\bar{T}_i}^e} \bar{T}_i \phi_a e_i d\Gamma + \int_{\Omega^e} f_i \phi_a e_i d\Omega - \sum_{j=1}^{N_{sd}} \sum_{b \in \eta_{u_j}} [m_{(ai)(bj)}^e \bar{u}_{(bj)}^e + k_{(ai)(bj)}^e \bar{u}_{(bj)}^e]$$

$$A \in \eta_i, \quad B \in \eta_j, \quad a, b \in [1, \dots, N_{en}], \quad i, j \in [1, \dots, N_{sd}]$$

## 2.2 Hybrid problem

Consider the structure described in section 2.1. Assume it consists of a sub-domain in which the solution is almost independent of one of the coordinates. This sub-domain will be modeled as two-dimensional, such as the finite element model of the entire structure consists of a 3D sub-domain and a 2D sub-domain. The interface boundary between the 3D and 2D sub-domains will be denoted  $\Gamma_B$ . The 3D and 2D sub-domains will be denoted  $\Omega_3, \Omega_2$ , respectively. Examples of hybrid models are presented in section 4 and are explained thoroughly later. In those cases,  $\Gamma_B = \{(x, y, z) | x = x_B\}$  and so we will assume this kind of boundary in the derivation of the strong and the weak form.

### 2.2.1 The strong form

Without loss of generality, assume that the solution in  $\Omega_2$  is independent of the coordinate  $x_3$ . The governing equations are:

$$\rho \ddot{u}_i^{(3)} = \sigma_{ij,j}^{(3)} + f_i^{(3)}, \quad i = 1, 2, 3 \text{ in } \Omega_3 \quad (28)$$

$$\rho \ddot{u}_i^{(2)} = \sigma_{ij,j}^{(2)} + f_i^{(2)}, \quad i = 1, 2 \text{ in } \Omega_2 \quad (29)$$

Where:

$$\sigma_{ij}^{(n)} = C_{ijkl} \varepsilon_{kl}^{(n)} = C_{ijkl} \frac{1}{2} (u_{k,l}^{(n)} + u_{l,k}^{(n)}) = C_{ijkl} u_{k,l}^{(n)}$$

The superscripts (3), (2) stand for a 3D and 2D quantities, respectively. The boundary conditions in the domain  $\Omega_3$  are the same as in the full 3D problem, except for the new boundary  $\Gamma_B$  which is chosen to be traction free. The boundary conditions in  $\Omega_2$  are reduced by averaging  $\bar{u}_i$  and  $\bar{T}_i$  with respect to  $x_3$  for  $i = 1, 2$ .

The initial conditions in  $\Omega_3$  are as in the full problem:

$$\mathbf{U}_0^{(3)} = \mathbf{U}_0, \quad \mathbf{V}_0^{(3)} = \mathbf{V}_0 \quad \text{in } \Omega_3$$

The initial conditions in  $\Omega_2$  are reduced by averaging with respect to  $x_3$  as before.

### 2.2.2 The weak form

In Panasenko's method, the continuity of the solution at the interface is enforced strongly. If  $\Gamma_B = \{(x, y, z) | x = x_B\}$  this condition can be simply written as:

$$u_i^{(3)}(x_B, y, z) = u_i^{(2)}(x_B, y); \quad i = 1, 2 \quad (30)$$

Define the function sets  $S, S_0$ :

$$S = \left\{ \mathbf{u} = \left( \mathbf{u}^{(2)}; \mathbf{u}^{(3)} \right) \mid \mathbf{u}^{(2)} \in C^0(\Omega_2), \mathbf{u}^{(3)} \in C^0(\Omega_3), \mathbf{u}^{(3)}(x_B, y, z) = \mathbf{u}^{(2)}(x_B, y), u_i = \bar{u}_i \text{ on } \Gamma_{u_i} \right\}$$

$$S_0 = \left\{ \mathbf{w} = \left( \mathbf{w}^{(2)}; \mathbf{w}^{(3)} \right) \mid \mathbf{w}^{(2)} \in C^0(\Omega_2), \mathbf{w}^{(3)} \in C^0(\Omega_3), \mathbf{w}^{(3)}(x_B, y, z) = \mathbf{w}^{(2)}(x_B, y), w_i = 0 \text{ on } \Gamma_{u_i} \right\}$$

The notation  $\mathbf{u} = (\mathbf{u}^{(2)}; \mathbf{u}^{(3)})$  should be interpreted by:

$$\mathbf{u} = \begin{cases} \mathbf{u}^{(2)}(x, y), & \mathbf{x} \in \Omega_2 \\ \mathbf{u}^{(3)}(x, y, z) & \mathbf{x} \in \Omega_3 \end{cases}$$

Multiplying Eq. 28, 29 by a weighting function  $\mathbf{w} \in S$  and integrating over  $\Omega_3 \cup \Omega_2$  yields:

$$\int_{\Omega_3 \cup \Omega_2} w_i \rho \ddot{u}_i d\Omega = \int_{\Omega_3 \cup \Omega_2} \sigma_{ij,j} w_i d\Omega + \int_{\Omega_3 \cup \Omega_2} f_i w_i d\Omega$$

Considering a hybrid model in which the structure in the reduced sub-domain was originally of thickness  $\epsilon$ , the integrals over the domain  $\Omega_3 \cup \Omega_2$  are calculated by:

$$\int_{\Omega} g(x, y, z) d\Omega \approx \int_{\Omega_3} g^{(3)}(x, y, z) d\Omega_3 + \epsilon \int_{\Omega_2} g^{(2)}(x, y) d\Omega_2$$

Where  $g^{(2)}(x, y)$  is the 2D representation of the original  $g(x, y, z)$  in the domain  $\Omega_2$ . The repercussion of this on the element mass matrices in the domain  $\Omega_2$  is that they should be multiplied by  $\epsilon$ . Consider the case where  $f_i = 0$ :

$$\int_{\Omega_3} w_i^{(3)} \rho \ddot{u}_i^{(3)} d\Omega_3 + \epsilon \int_{\Omega_2} w_i^{(2)} \rho \ddot{u}_i^{(2)} d\Omega_2 = \int_{\Omega_3} \sigma_{ij,j}^{(3)} w_i^{(3)} d\Omega_3 + \epsilon \int_{\Omega_2} \sigma_{ij,j}^{(2)} w_i^{(2)} d\Omega_2$$

Using the divergence theorem yields:

$$\begin{aligned} \int_{\Omega_3} w_i^{(3)} \rho \ddot{u}_i^{(3)} d\Omega_3 + \epsilon \int_{\Omega_2} w_i^{(2)} \rho \ddot{u}_i^{(2)} d\Omega_2 &= \int_{\partial\Omega_3} \sigma_{ij}^{(3)} w_i^{(3)} n_j d\Omega_3 - \int_{\Omega_3} \sigma_{ij}^{(3)} w_{i,j}^{(3)} d\Omega_3 \\ &\quad + \epsilon \int_{\partial\Omega_2} \sigma_{ij}^{(2)} w_i^{(2)} n_j d\Omega_2 - \epsilon \int_{\Omega_2} \sigma_{ij}^{(2)} w_{i,j}^{(2)} d\Omega_2 \end{aligned}$$

Let us choose  $\mathbf{w}$  such that  $w_i = 0$  on  $\Gamma_{u_i}$ . Additionally, the problem we are interested in has homogeneous Neumann boundary conditions on  $\Gamma_{T_i}$ , thus:

$$\int_{\Omega_3} w_i^{(3)} \rho \ddot{u}_i^{(3)} d\Omega_3 + \epsilon \int_{\Omega_2} w_i^{(2)} \rho \ddot{u}_i^{(2)} d\Omega_2 = - \int_{\Omega_3} C_{ijkl} u_{k,l}^{(3)} w_{i,j}^{(3)} d\Omega_3 - \epsilon \int_{\Omega_2} C_{ijkl} u_{k,l}^{(2)} w_{i,j}^{(2)} d\Omega_2$$

Denote:

$$b(\mathbf{w}, \mathbf{u}) = \int_{\Omega_3} w_i^{(3)} \rho u_i^{(3)} d\Omega_3 + \epsilon \int_{\Omega_2} w_i^{(2)} \rho u_i^{(2)} d\Omega_2, \quad a(\mathbf{w}, \mathbf{u}) = \int_{\Omega_3} C_{ijkl} u_{k,l}^{(3)} w_{i,j}^{(3)} d\Omega_3 + \epsilon \int_{\Omega_2} C_{ijkl} u_{k,l}^{(2)} w_{i,j}^{(2)} d\Omega_2$$

Therefore, the weak form of the problem:

Find  $\mathbf{u} \in S$  such that  $\mathbf{u}(\mathbf{x}, 0) = (\mathbf{U}_0^{(2)}; \mathbf{U}_0^{(3)})$ ,  $\dot{\mathbf{u}}(\mathbf{x}, 0) = (\mathbf{V}_0^{(2)}; \mathbf{V}_0^{(3)})$  and for all  $\mathbf{w} \in S_0$ :

$$b(\mathbf{w}, \ddot{\mathbf{u}}) + a(\mathbf{w}, \mathbf{u}) = 0$$

$$b(\mathbf{w}, \mathbf{u}) = \int_{\Omega_3} w_i^{(3)} \rho u_i^{(3)} d\Omega_3 + \epsilon \int_{\Omega_2} w_i^{(2)} \rho u_i^{(2)} d\Omega_2$$

$$a(\mathbf{w}, \mathbf{u}) = \int_{\Omega_3} C_{ijkl} u_{k,l}^{(3)} w_{i,j}^{(3)} d\Omega_3 + \epsilon \int_{\Omega_2} C_{ijkl} u_{k,l}^{(2)} w_{i,j}^{(2)} d\Omega_2$$

### 2.2.3 The semi-discrete problem and assembly of $\mathbf{M}, \mathbf{K}$

One can discretize the equations of the weak form using Galerkin FE approximation. The approximation is done at the element level for each sub-domain individually:

$$u_i^{(2)h} = \sum_{A \in \eta^{2D}} d_{(Ai)}(t) \phi_A^{(2)}(\mathbf{x}) + \sum_{A \in \eta_{u_i}^{2D}} \bar{u}_{(Ai)}^{(2)}(t) \phi_A^{(2)}(\mathbf{x}); \quad i = 1, 2$$

$$u_i^{(3)h} = \sum_{A \in \eta^{3D}} d_{(Ai)}(t) \phi_A^{(3)}(\mathbf{x}) + \sum_{A \in \eta_{u_i}^{3D}} \bar{u}_{(Ai)}^{(3)}(t) \phi_A^{(3)}(\mathbf{x}); \quad i = 1, 2, 3$$

$$\eta^{2D} = \{\text{set of all nodes in the 2D sub-domain which are open in direction } i\}$$

$$\eta_{u_i}^{2D} = \{\text{set of all nodes in the 2D sub-domain which are closed in direction } i\}$$

$$\eta^{3D} = \{\text{set of all nodes in the 3D sub-domain which are open in direction } i\}$$

$$\eta_{u_i}^{3D} = \{\text{set of all nodes in the 3D sub-domain which are closed in direction } i\}$$

$$\bar{u}_{(Ai)}^{(2)} = \bar{u}_i^{(2)}(\mathbf{x}_A), \quad \bar{u}_{(Ai)}^{(3)} = \bar{u}_i^{(3)}(\mathbf{x}_A)$$

The shape functions  $\phi^{(2)}, \phi^{(3)}$  are bi-linear and tri-linear shape functions defined by Eq. 26, 19 in  $\xi$  domain, respectively. The coupling between the two approximations by Eq. 30 is carried out by assigning the same open node numbers to all the nodes at a certain point on the interface plane on plane  $XY (x_B, y_0)$ , from both 3D and 2D sub-domains. Substituting the approximation in the weak form results in the linear system of equations:

$$\begin{cases} \mathbf{M} \cdot \ddot{\mathbf{d}} + \mathbf{K} \cdot \mathbf{d} & = \mathbf{0} \\ \mathbf{d}(0) & = \mathbf{d}_0 \\ \dot{\mathbf{d}}(0) & = \mathbf{v}_0 \end{cases}$$

It differs from the full 3D case semi-discrete problem by the number of unknowns and by the assembly of the matrices  $\mathbf{M}, \mathbf{K}$ . The hybrid assembly algorithm consists of two loops over the elements in the model -

one loop for the 3D elements and one loop for the 2D elements. Obviously, when looping over 3D elements one should use the 3D element matrices and when looping over 2D elements one should use the 2D element matrices (multiplied by  $\epsilon$ ). Let us denote the hybrid assembly operation of the global matrix or vector  $\mathbf{M}$  from the element level matrix/vector  $\mathbf{m}^e$  by  $\mathbf{M} = \mathcal{A}_{e=1}^{\mathcal{H}^{N_{el}}} \{\mathbf{m}^e\}$ . The full semi-discrete problem formulation for the hybrid case, with the assembly operation:

$$\begin{cases} \mathbf{M} \cdot \ddot{\mathbf{d}} + \mathbf{K} \cdot \mathbf{d} &= \mathbf{0} \\ \mathbf{d}(0) &= \mathbf{d}_0 \\ \dot{\mathbf{d}}(0) &= \mathbf{v}_0 \end{cases}$$

$$\mathbf{M} = [M_{(Ai)(Bj)}], \quad \mathbf{K} = [K_{(Ai)(Bj)}], \quad \mathbf{F} = \{F_{(Ai)}\}, \quad \mathbf{d} = \{d_{(Ai)}\}$$

$$\mathbf{d}_0 = \{d_{0(Ai)}\} = \{U_{0i}(\mathbf{x}_A)\}, \quad \mathbf{v}_0 = \{v_{0(Ai)}\} = \{V_{0i}(\mathbf{x}_A)\}$$

$$\mathbf{M} = \mathcal{A}_{e=1}^{\mathcal{H}^{N_{el}}} \{\mathbf{m}^e\}, \quad \mathbf{K} = \mathcal{A}_{e=1}^{\mathcal{H}^{N_{el}}} \{\mathbf{k}^e\}, \quad \mathbf{F} = \mathcal{A}_{e=1}^{\mathcal{H}^{N_{el}}} \{\mathbf{f}^e\}$$

$$\mathbf{m}^e = [m_{(ai)(bj)}^e], \quad \mathbf{k}^e = [k_{(ai)(bj)}^e], \quad \mathbf{f}^e = \{f_{(ai)}^e\}$$

$$m_{(ai)(bj)}^e = \begin{cases} \epsilon \delta_{ij} \int_{\Omega^e} \rho \phi_a^{(2)} \phi_b^{(2)} d\Omega_2, & e \in Q_{2D} \\ \delta_{ij} \int_{\Omega^e} \rho \phi_a^{(3)} \phi_b^{(3)} d\Omega_3, & e \in Q_{3D} \end{cases}, \quad k_{(ai)(bj)}^e = \begin{cases} \epsilon \int_{\Omega^e} \phi_{a,k}^{(2)} C_{ikjl} \phi_{b,l}^{(2)} d\Omega_2, & e \in Q_{2D} \\ \int_{\Omega^e} \phi_{a,k}^{(3)} C_{ikjl} \phi_{b,l}^{(3)} d\Omega_3, & e \in Q_{3D} \end{cases}$$

$$f_{(ai)}^e = \begin{cases} \epsilon \int_{\Gamma_{T_i}^e} \bar{T}_i \phi_a^{(2)} e_i d\Gamma_2 + \epsilon \int_{\Omega^e} f_i^{(2)} \phi_a^{(2)} e_i d\Omega_2 - \epsilon \sum_{j=1}^2 \sum_{b \in \eta_{u_j}} [m_{(ai)(bj)}^e \ddot{u}_{(bj)}^e + k_{(ai)(bj)}^e \bar{u}_{(bj)}^e], & e \in Q_{2D} \\ \int_{\Gamma_{T_i}^e} \bar{T}_i \phi_a^{(3)} e_i d\Gamma_3 + \int_{\Omega^e} f_i^{(3)} \phi_a^{(3)} e_i d\Omega_3 - \sum_{j=1}^3 \sum_{b \in \eta_{u_j}} [m_{(ai)(bj)}^e \ddot{u}_{(bj)}^e + k_{(ai)(bj)}^e \bar{u}_{(bj)}^e], & e \in Q_{3D} \end{cases}$$

$$a, b \in \begin{cases} [1, \dots, N_{en}^{(2)}], & e \in Q_{2D} \\ [1, \dots, N_{en}^{(3)}], & e \in Q_{3D} \end{cases}, \quad i, j \in \begin{cases} [1, 2], & e \in Q_{2D} \\ [1, 2, 3], & e \in Q_{3D} \end{cases}, \quad A \in \eta_i, \quad B \in \eta_j$$

$$Q_{2D} = \{\text{set of all 2D elements}\}, \quad Q_{3D} = \{\text{set of all 3D elements}\}$$

### 2.3 Solution of the semi-discrete problem: Newmark family of time stepping methods

Consider the time-dependent linear system of equations 15. Let us denote:

$$\Delta t - \text{time increment}, \quad n - \text{time step number}, \quad t_n = n\Delta t$$

$\mathbf{d}_n, \mathbf{v}_n, \mathbf{a}_n$  are the approximations  $\mathbf{d}, \dot{\mathbf{d}}, \ddot{\mathbf{d}}$  at  $t = t_n$ , respectively:

$$\mathbf{d}_n \approx \mathbf{d}(t_n), \quad \mathbf{v}_n \approx \dot{\mathbf{d}}(t_n), \quad \mathbf{a}_n \approx \ddot{\mathbf{d}}(t_n)$$

$\mathbf{F}_n = \mathbf{F}(t_n)$  is the exact force vector at  $t = t_n$ .

Newmark time-stepping method approximates  $\mathbf{d}_{n+1}, \mathbf{v}_{n+1}$  by:

$$\mathbf{v}_{n+1} \approx \mathbf{v}_n + \Delta t [(1 - \gamma) \mathbf{a}_n + \gamma \mathbf{a}_{n+1}], \quad 0 \leq \gamma \leq 1$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t \mathbf{v}_n + \frac{(\Delta t)^2}{2} [(1 - 2\beta) \mathbf{a}_n + 2\beta \mathbf{a}_{n+1}], 0 \leq \beta \leq \frac{1}{2}$$

Starting from the initial vector  $\mathbf{d}_0$ , the time stepping is carried out using a predictor-corrector algorithm:

Predictor:

$$\tilde{\mathbf{d}}_{n+1} = \mathbf{d}_n \Delta t \mathbf{v}_n + \frac{(\Delta t)^2}{2} (1 - 2\beta) \mathbf{a}_n$$

$$\tilde{\mathbf{v}}_{n+1} = \mathbf{v}_n \Delta t (1 - \gamma) \mathbf{a}_n$$

Corrector:

$$\mathbf{d}_{n+1} = \tilde{\mathbf{d}}_{n+1} + \beta (\Delta t)^2 \mathbf{a}_{n+1} \quad (31)$$

$$\mathbf{v}_{n+1} = \tilde{\mathbf{v}}_{n+1} + \gamma \Delta t \mathbf{a}_{n+1} \quad (32)$$

Solution:

$$\mathbf{M} \cdot \mathbf{a}_{n+1} + \mathbf{K} \cdot \mathbf{d}_{n+1} = \mathbf{F}_{n+1}$$

$$\mathbf{M} \cdot \mathbf{a}_{n+1} + \mathbf{K} \cdot (\tilde{\mathbf{d}}_{n+1} + \beta (\Delta t)^2 \mathbf{a}_{n+1}) = \mathbf{F}_{n+1}$$

$$\underbrace{\left( \mathbf{M} + \beta (\Delta t)^2 \mathbf{K} \right)}_{\mathbf{M}^*} \cdot \mathbf{a}_{n+1} = \underbrace{\mathbf{F}_{n+1} - \mathbf{K} \cdot \tilde{\mathbf{d}}_{n+1}}_{\mathbf{F}_{n+1}^*} \quad (33)$$

Solve [33](#) and plug into [31](#), [32](#) to obtain the solution at time step  $n + 1$ .  $\mathbf{d}_0, \mathbf{v}_0$  are given and  $\mathbf{a}_0$  is found from Eq. [33](#) with  $n + 1 = 0$ .

### 3 Description of the Computer Program

In each of the cases (1,2a,2b) described in section [4](#) the structure of the program is the same. The differences are in the geometric and mesh parameters and in the mesh generation. In each case the function “main” is divided into 5 sections:

1. Parameters: Definition of geometric, mesh, material, time stepping and initial conditions parameters.
2. Mesh: Generation of the mesh and the matrices  $\text{ID}(i, \hat{A}) = A$ ,  $\text{IEN}(a, e) = \hat{A}$ ,  $\text{LM}(i, a, e) = A$  where  $\hat{A}$  is the global node number,  $A$  is the global unknown number,  $i$  is the spatial dimension index - representing the number of the DOF in a certain node,  $e$  is the element number and  $a$  is the element level node number. If node  $\hat{A}$  is not an unknown in direction  $i$  (or, node  $a$  in element  $e$  is unknown in direction  $i$ ) the global unknown number of  $\hat{A}$  is  $A = 0$ . These matrices are used to obtain the initial conditions vectors and identify the unknowns in the assembly algorithm.
3. Assembly: The assembly of the global mass and stiffness matrices.
4. Initial Conditions: Get the initial conditions vectors  $\mathbf{d}_0, \mathbf{v}_0$ .
5. Time Stepping: Implementation of Newmark time stepping to obtain the time-accurate solution.

The relevant functions in each of the sections mentioned above are listed below.

## Mesh

- Mesh Generation: There are 4 functions which generate the computational mesh: Two for case 1 (“FullMeshCase1” for full & “HybridMeshCase1” for hybrid problem) and two for case 2 (“FullMeshCase2” for full & “HybridMeshCase2” for hybrid problem).
- Matrices: The function “Gen $a\_bc$ ” where  $a$  is “ID”/”IEN”/”LM”  $b$  is “Full” or “Hybrid” and  $c$  is “1” or “2” returns the matrix  $a$  of the  $b$  (full/hybrid) problem in case  $c$ . For example, “GenID\_Full12” returns the matrix ID of the full problem in case 2. In the hybrid problems, the matrices ID,IEN,LM are splitted in two - the matrices ID\_H, IEN\_H, LM\_H are of the higher-dimension sub-domain and ID\_L, IEN\_L, LM\_L are of the lower-dimension sub-domain. In the assembly process, the higher-dimension matrices are used in the first element loop (over the higher-dimension elements), and the lower-dimension matrices are used in the second loop (over the lower-dimension elements).

## Assembly

- The functions “asmMK” and “asmMK\_Hybrid” return the global mass and stiffness matrix of the full and hybrid problem, respectively. These functions remain unchanged between cases 1 and 2. The full and hybrid assembly algorithms are as written in sections 2.1.5, 2.2.3.

## Initial Conditions

- The function “IC $\_bc$ ” where  $b$  is “Full” or “Hybrid” and  $c$  is “1”, “2a” or “2b” returns the initial conditions vectors  $\mathbf{d}_0, \mathbf{v}_0$  of the  $b$  (full/hybrid) problem in case  $c$ . The initial conditions in each case are presented in detail in section 4.

## Time Stepping

- The function “Newmark $\_bc$ ” where  $b$  is “Full” or “Hybrid” and  $c$  is “1” or “2” carries out Newmark time-stepping and plots the solution at each time step (via a sub-function which plots the solution). It returns a matrix “sol” which contains the solution at all the nodes at all time steps, and an array “frames” which is used to export the solution to a video. The calculations and storage of the matrix “sol” require a lot of memory and it increases the computational time significantly. It is used only when calculating the error. Therefore, when not needed, it should not be computed or stored. Eq. 33 is being solved iteratively using Preconditioned Conjugate Gradients Method, implemented by MATLAB function “pcg”. The maximum number of iterations is limited to 50, and the iterative process ceases when an error of  $10^{-4}$  is achieved.

The functions listed above are independent, namely the functions do not call each other, but only being called from “main”. “main” stores all the parameters, matrices and arrays, and sends the relevant input to each function.

# 4 Numerical experiments

## 4.1 Case 1: Plane stress problem

We’ll first consider a simple case where the solution is expected to be almost entirely two-dimensional, and examine whether the hybrid solution can or cannot approximate the reference solution well. The reference solution in this case is the finite elements solution of the full 3D problem. If the reference solution is entirely 2D on the interface plane, then the error associated with the dimension reduction is expected to be negligible. However, the coupling error might not be.

Plane stress problems involve a very thin plate in which the stresses are zero in the direction normal to the plate plane. Since the stresses are zero in this direction the problem is essentially two-dimensional. For that reason this is the first case that will be inspected to validate the hybrid solution.

### 4.1.1 Full problem specification

A thin plate is subjected to non-zero initial displacements. It has one fixed face, a smooth contact type of boundary condition is enforced on the upper and lower surfaces (namely the displacements normal to the top and bottom faces are zero and in the other directions they are traction free) and the rest faces are traction free.

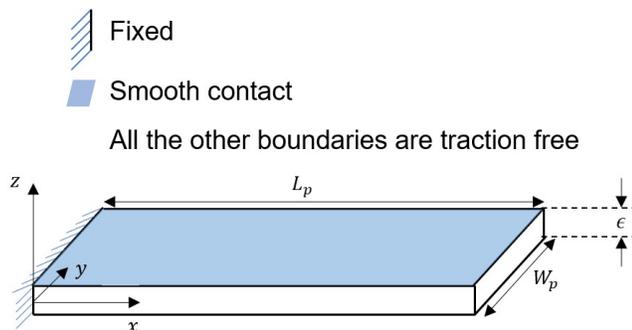


Figure 3: The plane stress problem discussed in case 1.

The initial displacement field is chosen to take the form:

$$\mathbf{U}_0(\mathbf{x}) = \begin{cases} A \cos(-4\pi r + \pi) \hat{\mathbf{r}}, & r \leq \frac{L_p}{4} \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (34)$$

$$\mathbf{V}_0(\mathbf{x}) \equiv \mathbf{0}$$

where  $r$  is the distance from the center of the plate and  $\hat{\mathbf{r}}$  is a unit radial vector:

$$r = \sqrt{\left(x - \frac{L_p}{2}\right)^2 + \left(y - \frac{W_p}{2}\right)^2}$$

$$\hat{\mathbf{r}} = \cos \theta \hat{\mathbf{x}} + \sin \theta \hat{\mathbf{y}}$$

$$\theta = \arctan \frac{y - \frac{W_p}{2}}{x - \frac{L_p}{2}}$$

The direction of the initial displacement field is purely radial with respect to the center of the plate. The chosen initial conditions are continuous and differentiable in the entire domain. It is shown on plane  $XY$  in Figure 4.

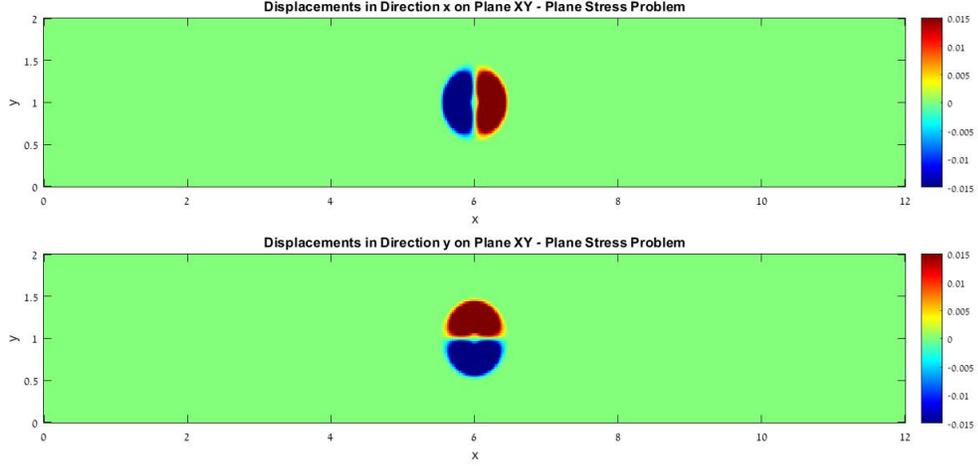


Figure 4: The initial conditions on plane  $XY$ , case 1.

Geometric parameters:  $L_p = 12 [m]$ ,  $W_p = 2 [m]$ ,  $\epsilon = 0.2 [m]$ .

Computational mesh: a uniform cubic tri-linear elements mesh with an edge length of  $h = 0.025 [m]$  was used. The mesh consists of 307,200 elements, 350,649 nodes and 972,000 degrees of freedom.

Time-stepping: an implicit method with  $\gamma = 0.5, \beta = 0.25$  was used, and the chosen time increment is  $\Delta t = 10^{-5} [sec]$  with a simulation time of  $3 \cdot 10^{-3} [sec]$  (300 time steps).

Material properties:  $\rho = 2.7 [\frac{gr}{cc}]$ ,  $G = 25.92 [GPa]$ ,  $\lambda = 60.49 [GPa]$  (of aluminum).

The amplitude of the initial displacements:  $A = 0.04 [m]$ .

#### 4.1.2 Hybrid problem specification

The hybrid problem domain is presented in the figure below. A part of the domain is remained unchanged while the other part is modeled as two-dimensional.

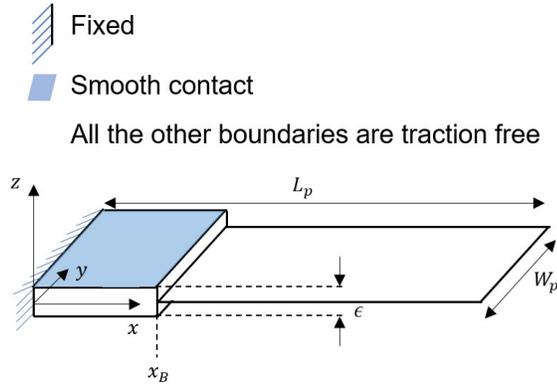


Figure 5: Hybrid plane stress problem discussed in case 1.

The same geometric parameters were chosen with interface location of  $x_B = 6 [m]$ , meaning that the interface plane is at the center of the plate. In the 3D sub-domain the same mesh as in the full 3D configuration was used, and a uniform squared elements mesh was used in the 2D sub-domain with  $h = 0.025 [m]$ . The mesh consists of 172,800 elements, 195,210 nodes and 523,584 degrees of freedom. The same time-stepping scheme as in the full 3D problem is considered. Since the initial conditions discussed in the

full problem are independent of the coordinate  $z$ , the exact same initial conditions are considered and there is no need to take the average (Eq. 34). The boundary conditions of the 2D sub-domain are all traction free, as well as the boundary conditions on the interface plane. The smooth contact boundary condition on the top and bottom faces is enforced naturally in the 2D sub-domain.

### 4.1.3 Results

The solution of the full and hybrid problems are presented in Figure 6 at a number of time steps. The interface plane in the hybrid problem's solution is marked by a thin red line.

The difference between the hybrid and reference solution is clearly small. The displacement field seems to be almost identical in both solution, while the hybrid solution has a significantly lower number of DOF. Now that it seems possible to benefit from the hybrid configuration, we can examine a more complicated problem in which the solution is not fully two-dimensional in the entire domain. A thorough analysis of the error will be carried out later.

## 4.2 Case 2: General 3D problem

### 4.2.1 Full 3D problem specification

Consider the structure shown in Figure 7. It consists of a cubic part with one of its faces fixed, and a thin plate with smooth contact boundary condition on the top and bottom faces. The rest of the boundaries are traction free. The structure is made from a linear isotropic material, such as aluminum. We are interested in the displacement field of the structure as a function of time due to non-trivial initial conditions.

Geometric parameters:  $L_c = 2 [m]$ ,  $L_p = 10 [m]$ ,  $\epsilon = 0.2 [m]$  in case (2a) and  $\epsilon = 0.25 [m]$  in case (2b) (both cases will be presented in detail later).

Computational mesh: uniform mesh, tri-linear cubic elements with edge length of  $h = 0.025 [m]$ . The grid is shown in Figure 8 for  $\epsilon = 0.2 [m]$ . The mesh consists of 832,000 elements, 887,841 nodes and 2,579,040 degrees of freedom with  $\epsilon = 0.25 [m]$ , and 768,000 elements, 823,041 nodes and 2,384,640 degrees of freedom with  $\epsilon = 0.2 [m]$ .

Time-stepping:  $\Delta t = 10^{-5}$  seconds, total simulation time of  $3 \cdot 10^{-3}$  seconds, a total of 300 time steps. The time stepping parameters are  $\gamma = \frac{1}{2}, \beta = \frac{1}{4}$ .

**Case (2a)** The initial displacements field is:

$$\mathbf{U}_0(\mathbf{x}) = \begin{cases} A \cos(-4\pi r + \pi) \hat{\mathbf{r}}, & r \leq \frac{L_c}{4} \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (35)$$

$$\mathbf{V}_0(\mathbf{x}) \equiv 0$$

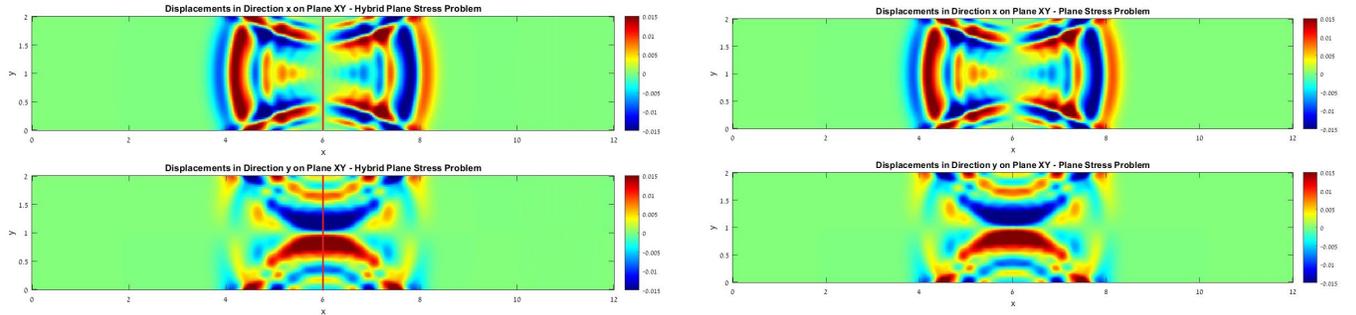
With:

$$A = 0.04 [m]$$

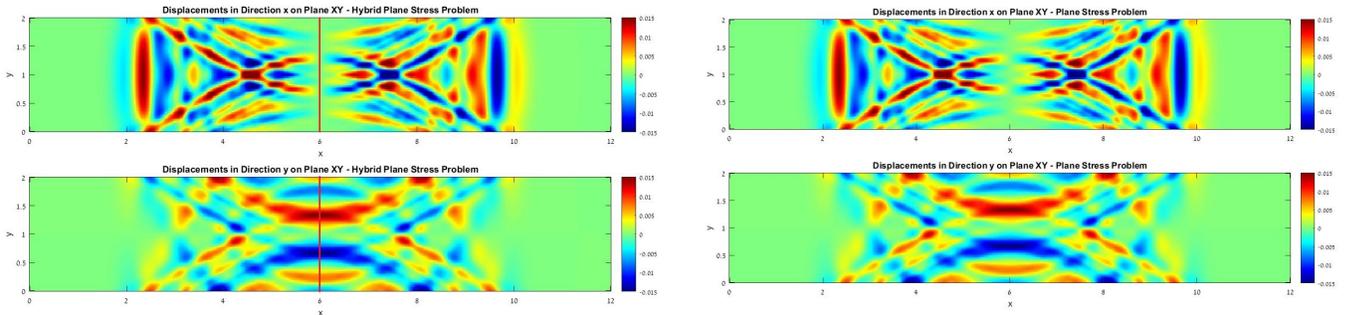
$$r = \sqrt{\left(x - \frac{L_c}{2}\right)^2 + \left(y - \frac{L_c}{2}\right)^2 + z^2}$$

$$\hat{\mathbf{r}} = \sin \theta \cos \phi \hat{\mathbf{x}} + \sin \theta \sin \phi \hat{\mathbf{y}} + \cos \theta \hat{\mathbf{z}}$$

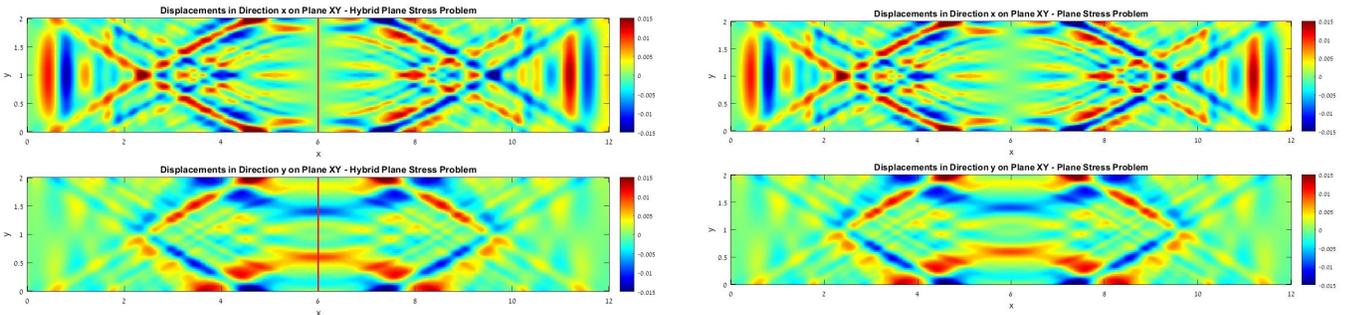
$$\theta = \arctan \frac{\sqrt{\left(x - \frac{L_c}{2}\right)^2 + \left(y - \frac{L_c}{2}\right)^2}}{z}$$



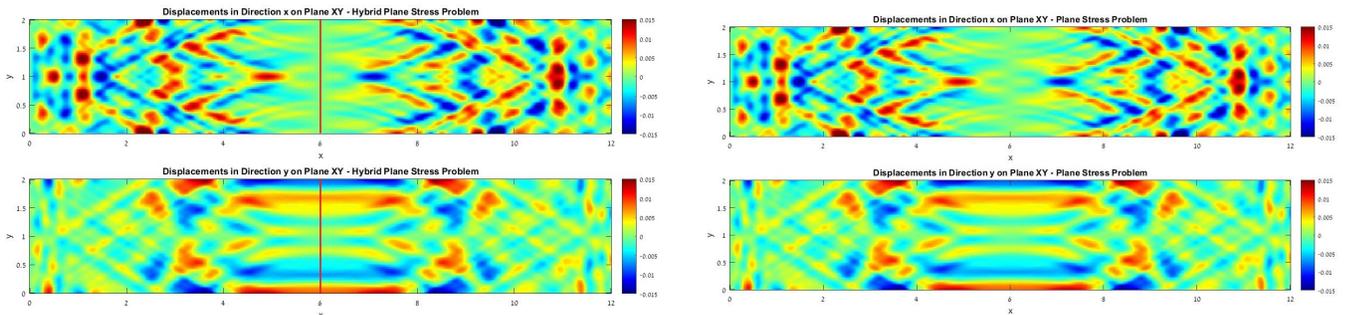
(a)  $t = 3 \cdot 10^{-4} [sec]$



(b)  $t = 6 \cdot 10^{-4} [sec]$



(c)  $t = 9 \cdot 10^{-4} [sec]$



(d)  $t = 1.2 \cdot 10^{-3} [sec]$

Figure 6: The solution of the full and hybrid problems at a number of time steps. Displacements in  $x$  direction are shown at the top of each sub-figure, displacements in  $y$  direction are at the bottom. The left part of each sub-figure shows the solution to the hybrid problem and the right part to the full problem.

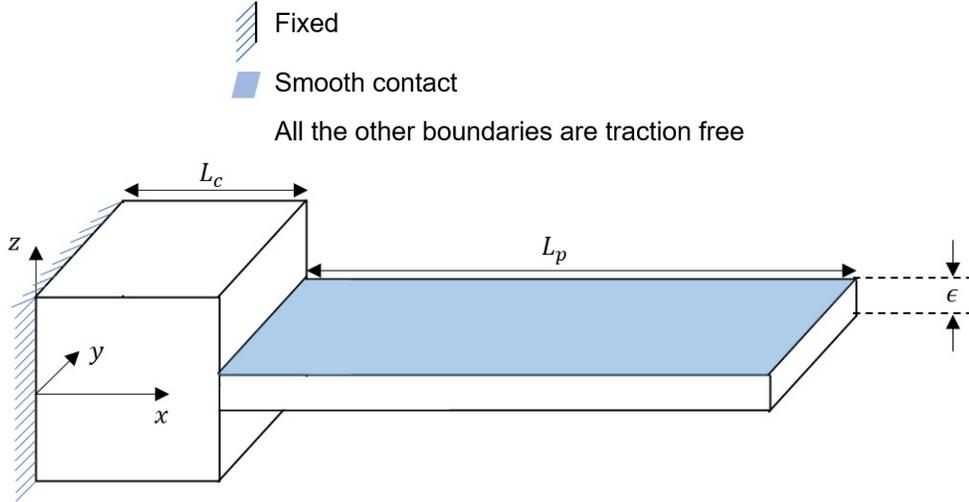


Figure 7: Full 3D problem structure discussed in case 2.

$$\phi = \arctan \frac{y - \frac{L_c}{2}}{x - \frac{L_c}{2}}$$

A snapshot of the initial conditions in  $XY$  plane is presented in Figure 9.

The initial displacements in  $z$  direction at the center of the cube  $x = y = \frac{L_c}{2}$  are shown in Figure 10.

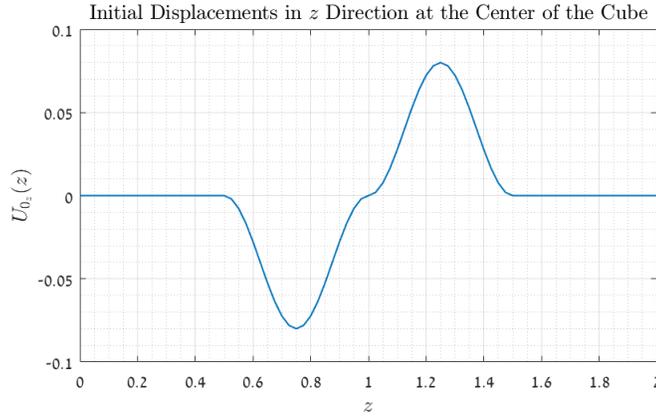
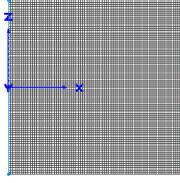


Figure 10: IC in  $z$  direction as a function of  $z$  at the center of the cube.

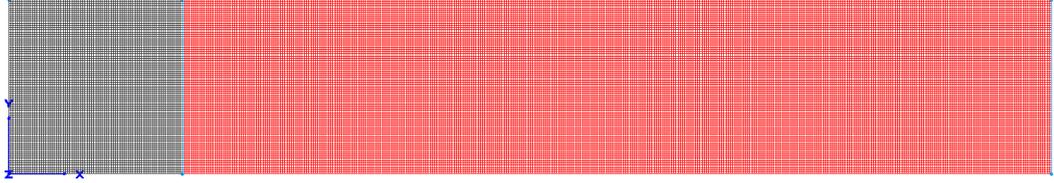
Figures 9, 10 emphasize the strong dependency of the initial conditions in all three coordinates, thus the solution in the cubic domain is expected to be fully three-dimensional and cannot be approximated using a 2D model. The solution in the thin plate, however, might be independent of the  $z$  coordinate far away from the cubic domain due to its small thickness and smooth contact boundary conditions.

**Case (2b)** In this case the initial conditions are given by:

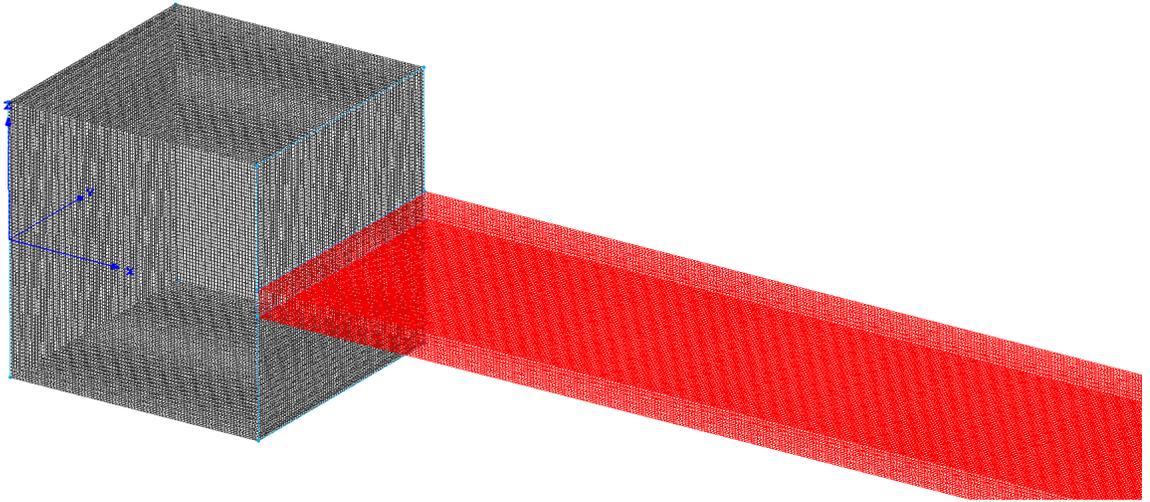
$$U_{0_x}(\mathbf{x}) = \begin{cases} \cos\left(\frac{n\pi(x-l/2)}{l}\right) \cos\left(\frac{n\pi(y-l/2)}{l}\right) \cos\left(\frac{n\pi z}{2l}\right), & x \in [0, l], y \in [0, l], z \in [-l, l] \\ 0, & \text{otherwise} \end{cases} \quad (36)$$



(a) Side view.



(b) Top view.



(c) Trimetric view.

Figure 8: Full problem's grid, case 2. Black grid is the cubic structure and red grid is the thin plate.

$$U_{0,y} \equiv U_{0,z} \equiv 0$$

$$\mathbf{V}_0(\mathbf{x}) \equiv \mathbf{0}$$

Where the parameters  $n, l$  are chosen arbitrarily. Obviously,  $l \leq L_c$ . The parameter  $l$  determines the volume of the non-trivial initial conditions domain and  $n$  determines the number of cycles of the displacement field (as it is periodic) in this domain. As  $n$  increases, the initial displacements are more oscillatory in the non-trivial IC domain. Notice that for even values of  $n$  the initial displacements are not continuous on the boundary of the non-trivial initial conditions domain.

For example, the initial displacements field for  $n = 0, 1, 2$  and  $l = L_c/2$  is presented in Figure 11.

In this case as well the solution in the cubic domain cannot be approximated as two dimensional.

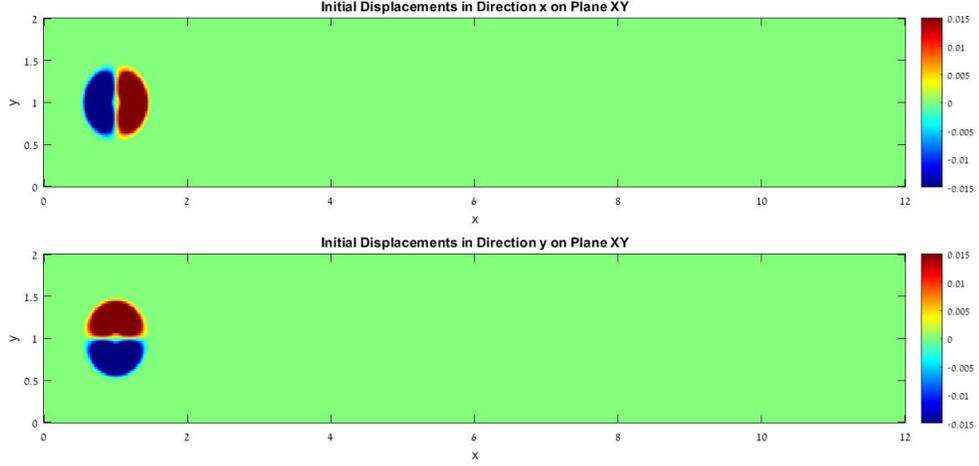


Figure 9: The initial conditions in case (2a).

#### 4.2.2 Hybrid problem specification

Because the plate is assumed to be much thinner than the cube and a smooth contact boundary condition is enforced along the thin plate, the variation of the solution in the direction of  $z$  is expected to be small in the thin plate. Therefore, let us approximate the model shown in Figure 7 by the model in Figure 12.

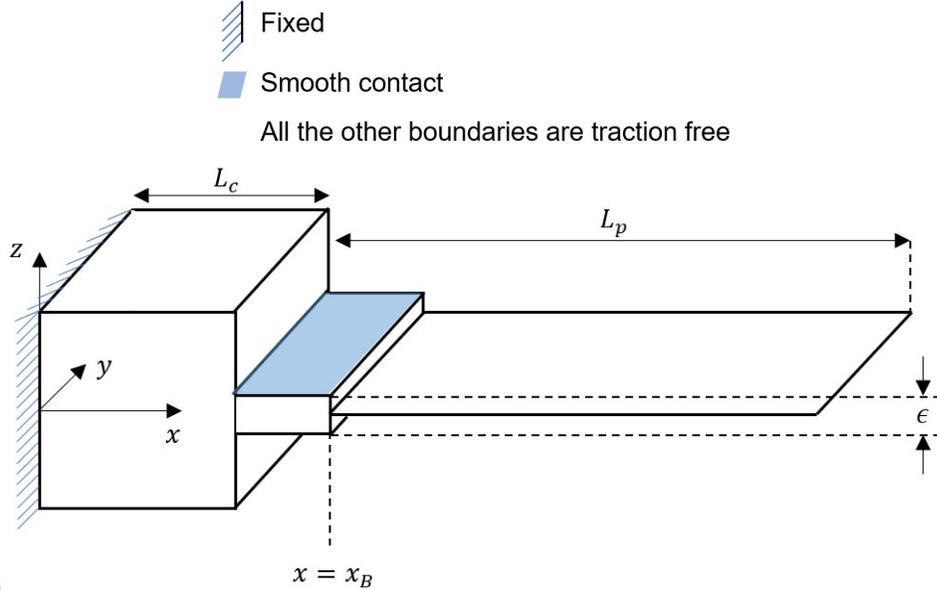


Figure 12: Hybrid problem model discussed in case 2.

The same geometric parameters as in the full problem are considered with interface location  $x_B = 4$ . A uniform mesh made of tri-linear cubic elements was used in the 3D sub-domain and a uniform squared bi-linear elements mesh was used in the 2D sub-domain, both with a mesh parameter  $h = 0.025 [m]$ . It is shown in Figure 13 for  $\epsilon = 0.2 [m]$ . The hybrid mesh consists of 601,600 elements, 628,722 nodes and 1,825,740 degrees of freedom with  $\epsilon = 0.25 [m]$ , and 588,800 elements, 615,762 nodes and 1,787,184 degrees of freedom for  $\epsilon = 0.2$ . The same time-stepping and material properties as in the full problem are discussed.

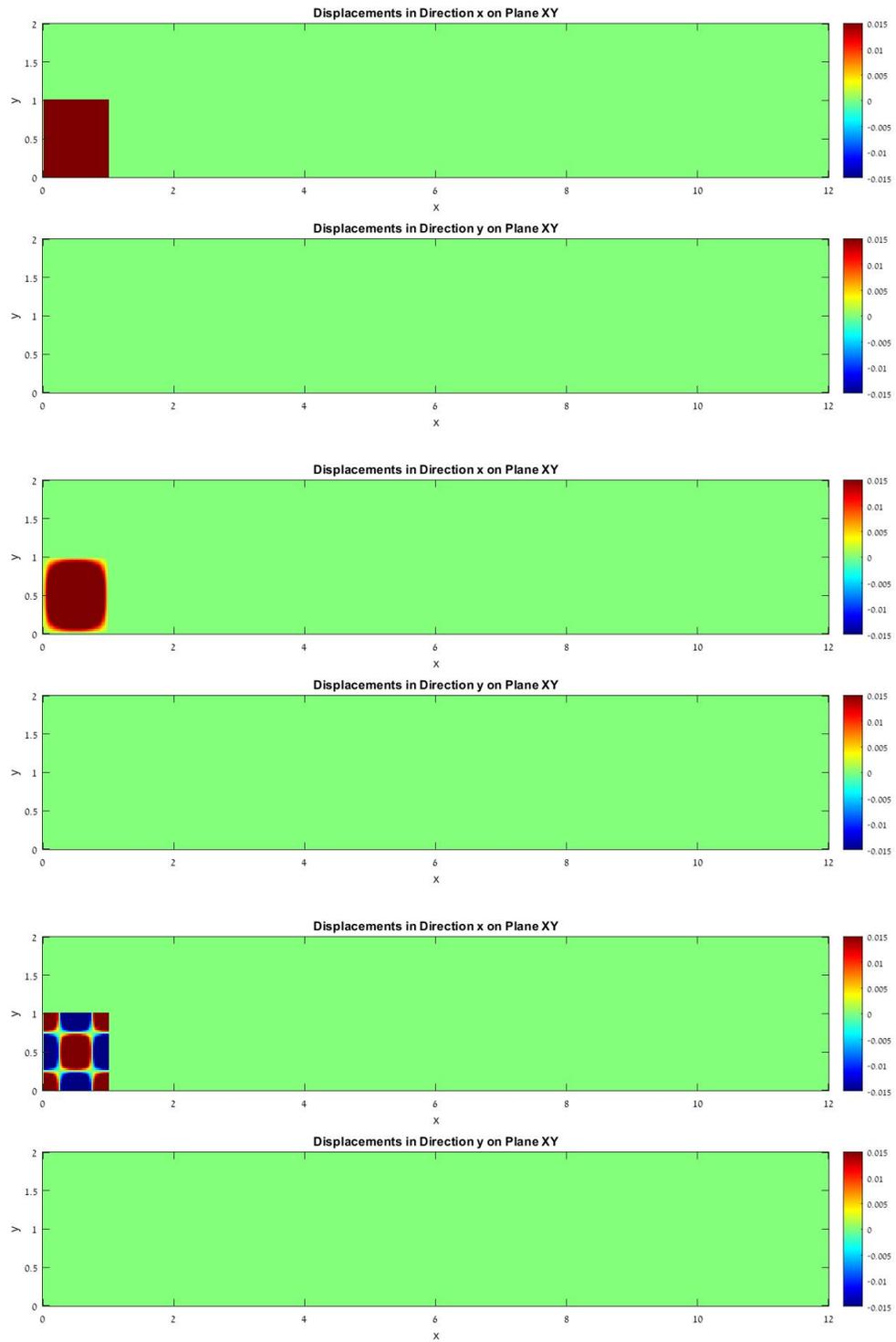


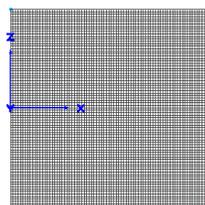
Figure 11: Initial displacements for  $n = 0, 1, 2$  and  $l = L_c/2$  (top to bottom).

In both cases (2a) and (2b) the initial conditions are zero in the 2D sub-domain, thus the initial conditions in the hybrid problem are the same as in Eq. 35, 36.

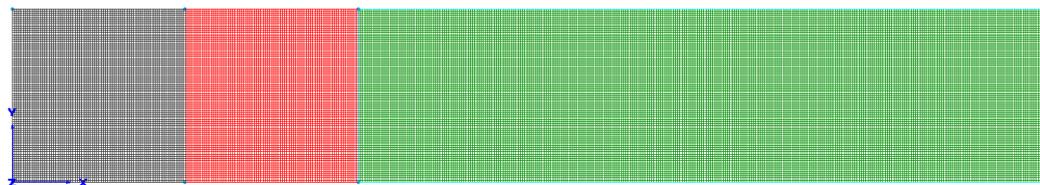
|                    | Full, $\epsilon = 0.2 [m]$ | Hybrid, $\epsilon = 0.2 [m]$ | Full, $\epsilon = 0.25 [m]$ | Hybrid, $\epsilon = 0.25 [m]$ |
|--------------------|----------------------------|------------------------------|-----------------------------|-------------------------------|
| Number of elements | 768,000                    | 588,800                      | 832,000                     | 601,600                       |
| Number of nodes    | 823,041                    | 615,762                      | 887,841                     | 628,722                       |
| Number of DOF      | 2,384,640                  | 1,787,184                    | 2,579,040                   | 1,825,740                     |

Table 3: Comparison between full and hybrid grid's number of elements, nodes and degrees of freedom

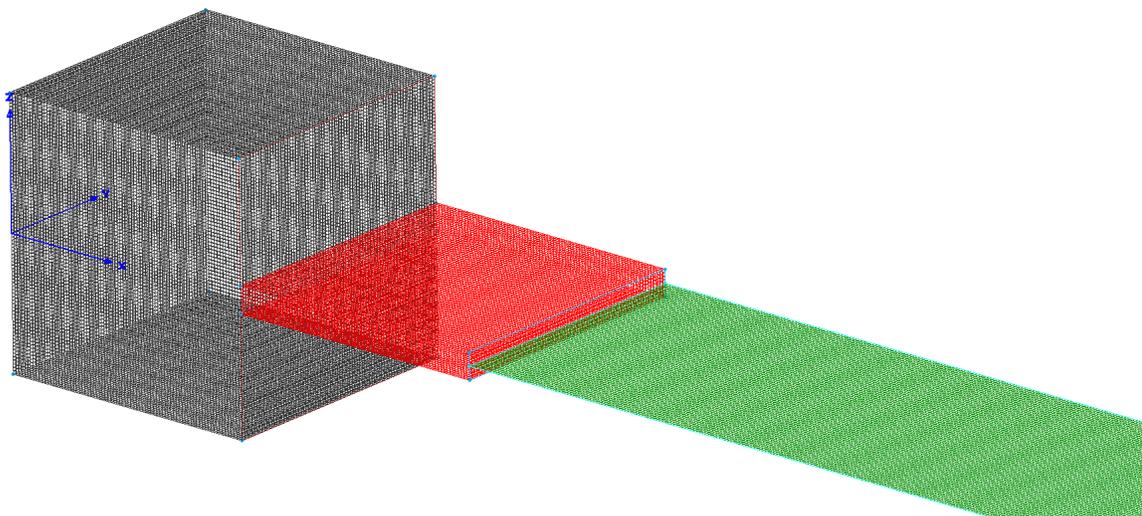
Table 3 shows a saving of about 600,000-800,000 (depending on  $\epsilon$ ) degrees of freedom with the use of a hybrid model.



(a) Side view.



(b) Top view.



(c) Trimetric view.

Figure 13: Hybrid problem's grid, case 2. Black grid is the cubic structure (3D), red grid is the 3D part of the thin plate and green grid is the 2D sub-domain.

### 4.2.3 Results

**Case (2a)** The magnitude of the displacements on plane  $XY$  at different times is presented in Figure 14, comparing between the full and the hybrid configurations. The interface of the hybrid model is marked by a red line. Similar to case 1, the difference between the solution of the hybrid problem and the full problem is hardly noticeable. However, recall that the full problem's solution in case 1 was inherently 2D, where in this case it is not.

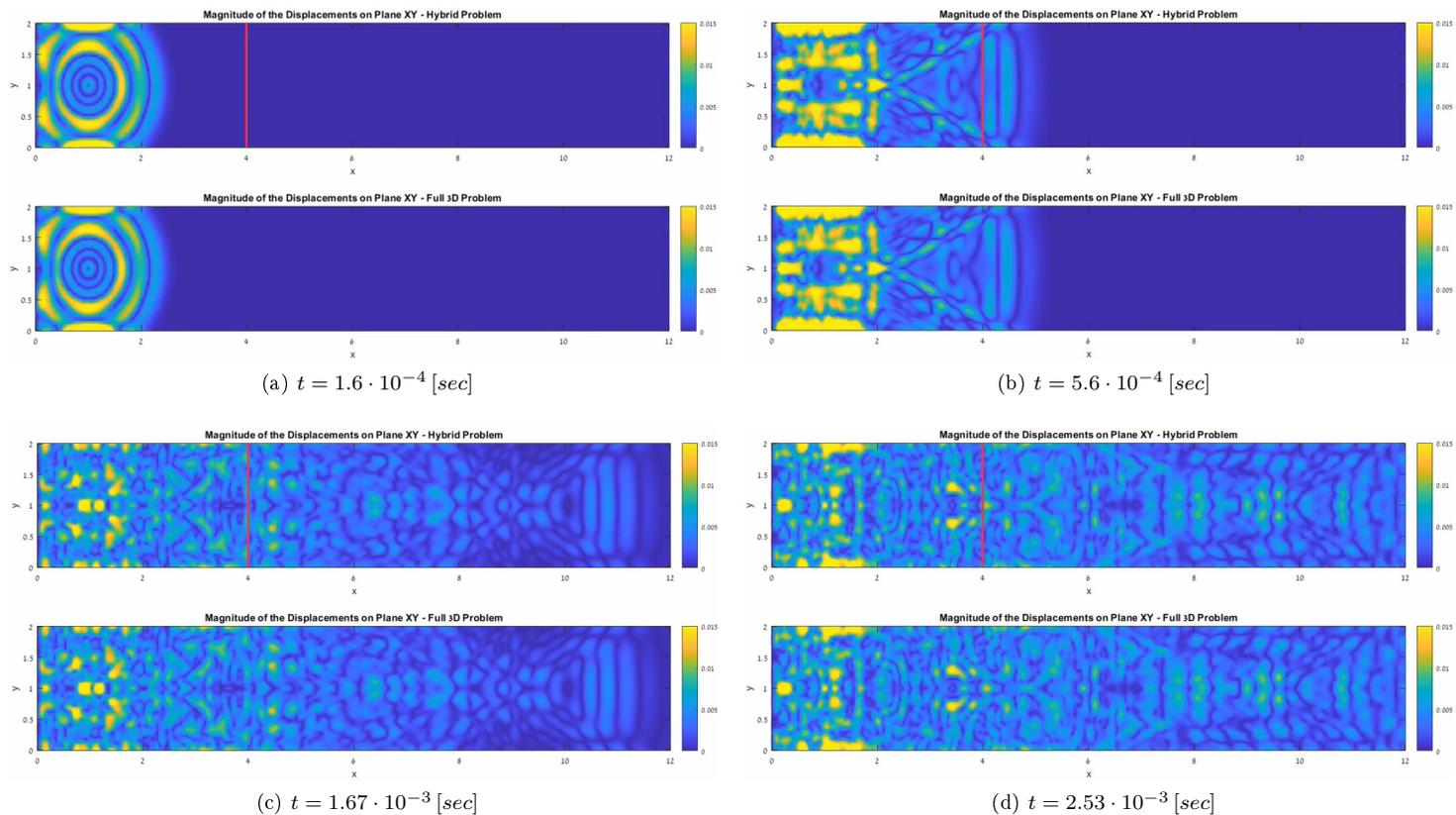


Figure 14: Magnitude of the displacements on plane  $XY$  at a number of time steps, hybrid and full problems comparison, case (2a). The solution to the hybrid problem is shown at the top of each sub-figure, and the solution to the full problem at the bottom.

**Case (2b)** Snapshots of the solution at several time steps are shown in Figure 15. In the hybrid solution, the interface plane is marked by a red line. The chosen initial conditions parameters are  $n = 1, l = \frac{L_c}{2}$ . Once again the hybrid solution follows the full solution with an unnoticeable error.

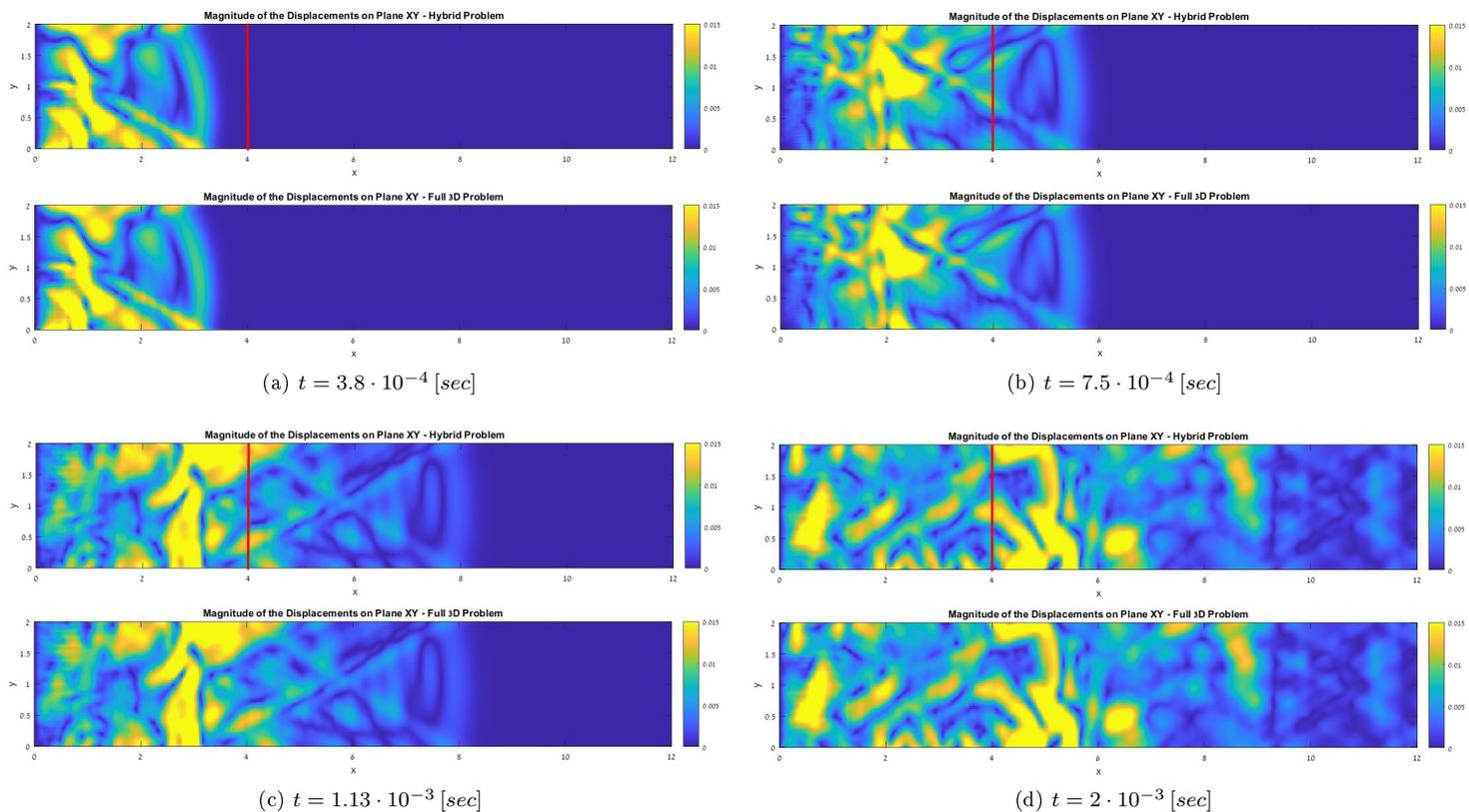


Figure 15: Magnitude of the displacements on plane  $XY$  at a number of time steps, hybrid and full problems comparison, case (2b). The solution to the hybrid problem is shown at the top of each sub-figure, and the solution to the full problem at the bottom.

### 4.3 Error Analysis

The error analysis will be carried out for the non-trivial cases (2a),(2b). Specifically, we will examine the variation of the error with the interface plane location  $x_B$  and thickness of the plate  $\epsilon$  in case (2a) and the dependency of the error on the initial conditions parameters  $n, l$  in case (2b). It is expected to obtain that the error drops as the interface gets farther away from the 3D sub-domain in the hybrid model in case (2a), as the reference solution becomes less dependent on  $z$  with the distance along the thin plate. Additionally, the solution is less 3D when the plate is thin, therefore we also expect the error to grow when discussing thicker plates.

#### 4.3.1 Variance Calculation

The variance of the displacements field  $\mathbf{u}(\mathbf{x})$  at a constant  $x$  section in the thin plate is defined as:

$$\text{var}(x) = \frac{1}{A} \frac{1}{T} \int_0^{L_c} \int_0^\epsilon \int_0^T |\mathbf{u}(x, y, z, t) - \bar{\mathbf{u}}(x, y, t)| dt dz dy$$

$$\bar{\mathbf{u}}(x, y, t) = \frac{1}{\epsilon} \int_0^\epsilon \mathbf{u}(x, y, z, t) dz$$

$\bar{\mathbf{u}}(x, y, t)$  is the average of the solution with respect to  $z$ ,  $A = L_c \epsilon$  is the cross section area of the plate,  $T$  is the total simulation time. The variance provides a quantitative representation of the 3D nature of the solution in a certain  $x$  section. If the variance is zero at a certain  $x$  value, it means that the solution does not depend on  $z$  at all in that  $x$  section. As explained before, Panasenko's coupling method strongly enforces that the solution on the interface plane is two-dimensional. The coupling error will be reduced if the true solution is somewhat two-dimensional on the interface plane. Expanding the variance definition to the finite elements solution  $\mathbf{u}^h$  at a constant  $x = x_i$  section results in:

$$\text{var}(x_i) = \frac{1}{A} \frac{1}{T} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \sum_{n=1}^{N_t} |\mathbf{u}^h(x_i, y_j, z_k, t_n) - \bar{\mathbf{u}}^h(x_i, y_j, t_n)| \Delta t \Delta z \Delta y \quad (37)$$

where  $\bar{\mathbf{u}}^h$  is the average displacement in the  $z$  direction:

$$\bar{\mathbf{u}}^h(x_i, y_j, t_n) = \frac{1}{\epsilon} \sum_{k=1}^{N_z} \bar{\mathbf{u}}^h(x_i, y_j, z_k, t_n) \Delta z$$

$N_y$  is the number of nodes along  $y$ -axis direction at  $x = x_i$ ,  $N_z$  is the number of nodes along  $z$ -axis at  $x = x_i$ ,  $N_t$  is the number of time steps. Since we consider a uniform mesh in all cases:  $\Delta z = \Delta y = h$ .

**Case (2a)** The variance of the full solution of case (2a) along the thin plate is presented in Figure 16 with two  $\epsilon$  values.  $\hat{x}$  is the normalized distance along the plate, defined as  $\hat{x} = \frac{x-L_c}{L_p}$ . At the vicinity of the intersection between the thin plate and the cubic structure the solution is still three-dimensional, and its dependency on  $z$  grows with the thickness of the plate. Thus the variance is larger for the thicker plate when  $\hat{x}$  is small. For larger values of  $\hat{x}$  the variance is small in both cases as discussed previously.

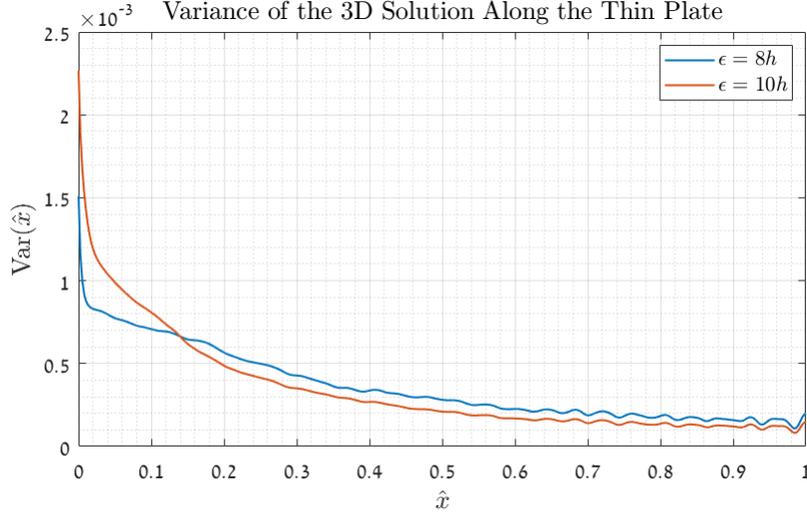


Figure 16: Variance of the full 3D solution of case (2a) for  $\epsilon = 8h, 10h$ .

**Case (2b)** The variance of the 3D solution of case (2b) is shown in Figure 17 for the initial conditions parameters:  $n = 2, 3$ ,  $l/L_c = 0.25, 0.5, 0.75, 1$ . At higher values of  $\hat{x}$  it is clear that as the region of non-trivial initial condition grows, the variance along the thin plate increases as well. For smaller  $\hat{x}$  the case  $l/L_c = 0.5$  does not follow this conclusion, but lower values of  $\hat{x}$  are usually not of interest since the coupling error would be large if the interface plane  $\hat{x}_B$  would be placed there.

### 4.3.2 Error Calculation

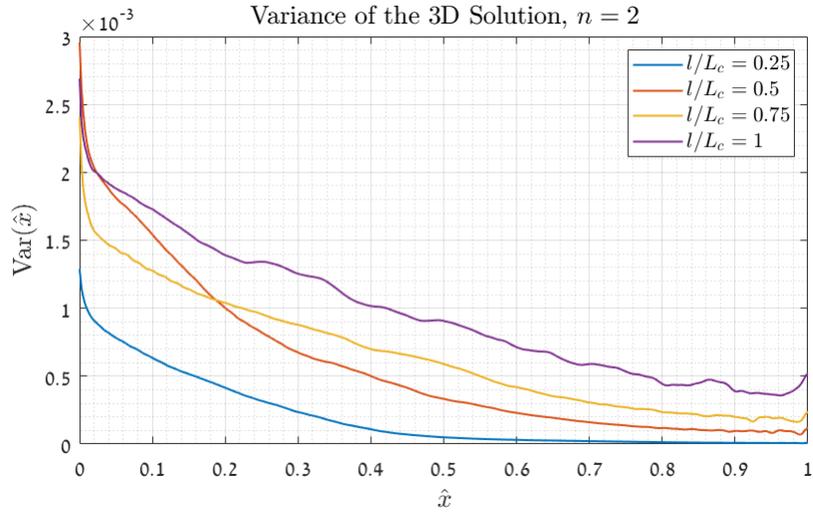
Let us define the error of the considered solution  $\mathbf{u}^h$  relative to the reference solution  $\mathbf{u}_{ref}$ :

$$E_1 = \frac{\sum_T \sum_{N_{cube}} |\mathbf{u}^h - \mathbf{u}_{ref}|}{\sum_T \sum_{N_{cube}} |\mathbf{u}_{ref}|}$$

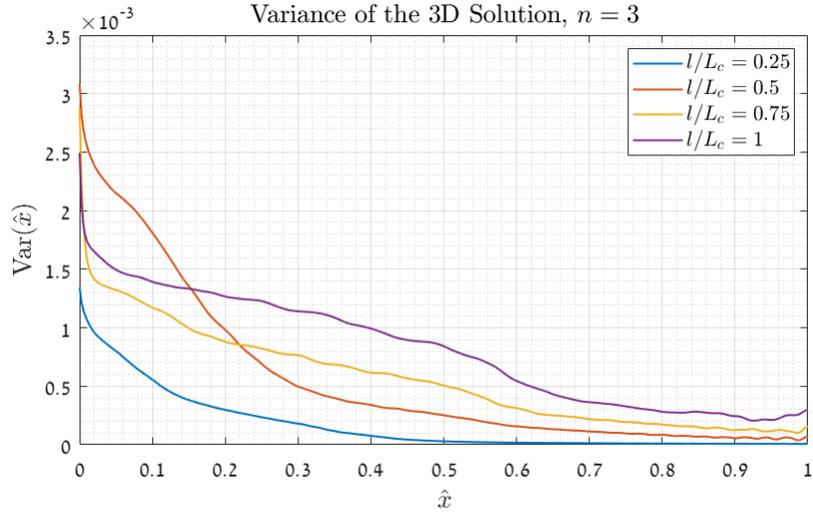
It is the summation of the norm  $|\mathbf{u}^h - \mathbf{u}_{ref}|$  at all time steps and at all nodes in the cubic structure, normalized by the sum of the norm  $|\mathbf{u}_{ref}|$  at the same time steps and nodes. The error is only calculated at the cubic part of the domain, since it is of the most interest.

**Case (2a)** The error of the hybrid solution relative to the full 3D finite elements solution of case (2a) was calculated for the same values of  $\epsilon$  as in Figure 16, and for several values of interface location  $x_B$ . The results are shown in Figure 18. Obviously, as the interface plane is placed farther away from  $x = L_c$  the error decreases. It follows the conjecture written at the beginning of section 4.3, and is consistent with Figure 16. The error is small when the interface is positioned at a plane where the variance is adequately small. The error reaches a constant value at around  $\hat{x}_B = 0.15$ , it is the error associated with the coupling method and cannot be minimized with optimizing the interface location. Since the error is calculated relatively to the full 3D finite elements solution, it consists of both the coupling and the dimension-reduction error, but not the spatial and time discretizations caused by using a finite element model and a finite differences time marching scheme. Thus, it better calrifies the expenses of using the hybrid model instead of a full 3D model.

Remark: The computer program written for this report failed to solve the hybrid problem with plate thickness  $\epsilon = 12h$ , probably due to an error in the program and not because of a limitation of the hybrid model.



(a) Variance for  $n = 2$ .



(b) Variance for  $n = 3$ .

Figure 17: Variance of the full 3D solution for  $n = 2, 3$  and several values of  $l$ , case (2b).

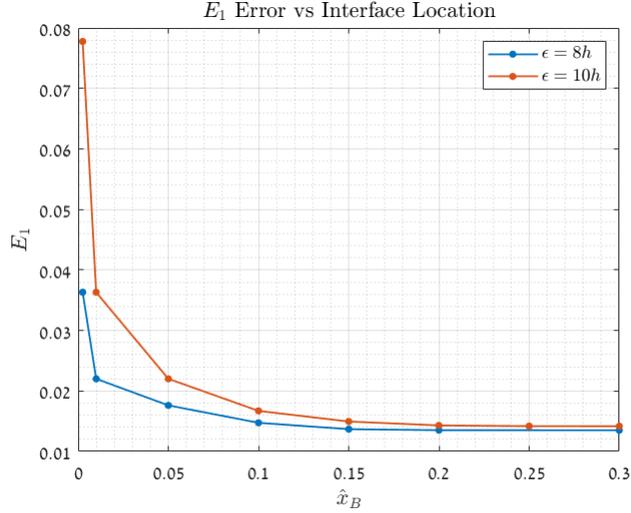


Figure 18:  $E_1$  error of the hybrid solution, case (2a).

**Case (2b)** The  $E_1$  error of the hybrid solution for a several values of initial conditions parameters  $n, l/L_c$  with plate thickness of  $\epsilon = 10h$  is shown in Figure 19. The error is the highest for  $n = 3$ , and is the lowest for  $n = 1$ . We would expect that as the wavelength of the initial displacements decreases (namely,  $n$  is higher and the initial displacements oscillation is at a higher spatial frequency) the error would be larger due to a rise in the three-dimensional nature of the problem. This is the case in the average sense, namely there is a certain rise in the error as  $n$  increases, but it is not monotonic. Recall that the initial conditions are only continuous for odd values of  $n$  (see Figure 11), and it might have an influence on the error.

Figure 17 implies (given that the interface is positioned far enough from the cubic structure) that the error should grow with  $l/L_c$ , as the variance increases with  $l/L_c$ . It seems that the error decreases when  $l/L_c$  increases (in the average sense), allegedly in contradiction to the results from Figure 17. However, notice that the differences in the error with the variation of the initial conditions parameters are about 1.5% at the most, where the differences in the error with the variation of the interface location is about 6% for the considered plate width ( $\epsilon = 10h$ ). Moreover, if observing specifically the influence of  $l/L_c$  on the error (taking a constant  $n$ ), the difference is about 1% at the most, and is even about 0.5% in most cases. Therefore, the dependency of the error on the initial condition parameters, especially the dependency on  $l/L_c$ , might be too weak to deduce solid conclusions on the dependency of the error on these parameters.

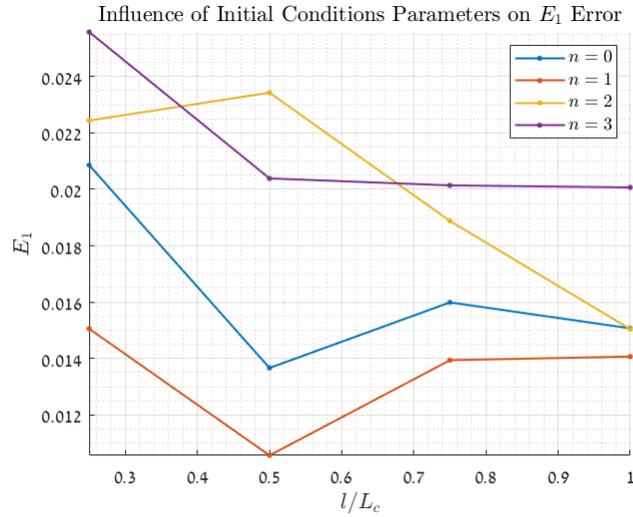


Figure 19:  $E_1$  error of the hybrid solution, case (2b).

## 5 Summary and Conclusions

Several cases were inspected in order to validate the hybrid finite-elements solution against the full solution. All resulted in the conclusion that if the interface plane is positioned well, the difference between the hybrid and full problem's solutions is small.

Different geometries, boundary conditions and initial conditions affect the three-dimensional nature of the solution in many ways, making the hybrid model less or more accurate. Still, if there is a domain of low dependency on one of the coordinates, the use of a low-dimension model in that domain will reduce the number of DOF in the entire model significantly with a very small harm to the accuracy and no harm to the resolution at all.