



HAL
open science

An algorithm to solve transient structural non-linear problems for non-matching time-space domains

Alain Combescure, Anthony Gravouil, B. Herry

► **To cite this version:**

Alain Combescure, Anthony Gravouil, B. Herry. An algorithm to solve transient structural non-linear problems for non-matching time-space domains. *Computers & Structures*, 2003, 81 (12), pp.1211-1222. <10.1016/S0045-7949(03)00037-3>. <hal-00454743>

HAL Id: hal-00454743

<https://hal.science/hal-00454743v1>

Submitted on 27 Jun 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

An algorithm to solve transient structural non-linear problems for non-matching time-space domains

A. Combescure^{a,*}, A. Gravouil^a, B. Herry^b

^a LMSO INSA Lyon 20 av Albert Einstein, 69621 Villeurbanne Cedex, France

^b LMT ENS Cachan 61 av du Président Wilson, 94235 Cachan Cedex, France

This paper presents a new transient algorithm to couple sub-domains having non-matching time and space interfaces. The time integrators in each of the sub-domains are of the Newmark family but can be different. The method is also developed for non-linear explicit transient computations for which the time steps are not constant. A number of examples are presented which illustrate the interest, quality and the efficiency of the proposed method.

Keywords: Transient analysis; Sub-domains; Non-matching interfaces; Finite element; Non-matching time steps

The object of this paper is to propose a general method for the resolution of structural transient problems associated with sub-domains having non-matching space and time interfaces. The computation of sub-domains with non-matching interfaces is of great practical interest because it allows the analysts to use incompatible meshes in each of the sub-domains. Some methods like the mortar method [1] are now well established for static linear problems. Non-matching time interfaces is also of some interest and has been also studied in the past years [2]. For all transient problems, time and space scales are usually coupled, as we have to take care of wave propagation: hence one should be able to consider time and space scales in a unique framework. A new approach is developed in this paper: first one recalls recent results obtained for time non-matching interfaces [3–5]: second a new method is presented for non-matching space interfaces, and compared to the mortar method. Both methods are then combined and imple-

mented for linear behaviour of the structures. An example consisting of different sub-domains where different numerical time integration schemes with different time steps are used is then presented. The paper shows afterwards how the method can be applied to non-linear structural behaviour. The corresponding algorithms are then presented in the case of implicit or explicit schemes for time integration. The particular case where all the sub-domains are computed using explicit solvers is then presented because this particular case makes the proposed formulation particularly attractive. Applications to coupled implicit explicit sub-domains are also presented.

1. Introduction

The main difficulties of the transient structural computations are the computation costs and the memory requirements for storing the data. The objective of this paper is to explore some new methods which permit to reduce time costs as well as memory necessary to solve the problem. All the paper is based on the use of Newmark time integration schemes with different parameters. The first idea is to divide the computation domain in s sub-domains Ω_k connected by l interfaces Σ_i (see

* Corresponding author. Address: LaMCoS, Batiment d’Alambert, 20 Allée des Sciences, Insa de Lyon, Villeurbanne Cedex 69621, France.

E-mail address: alain.combescure@insa-lyon.fr (A. Combescure).

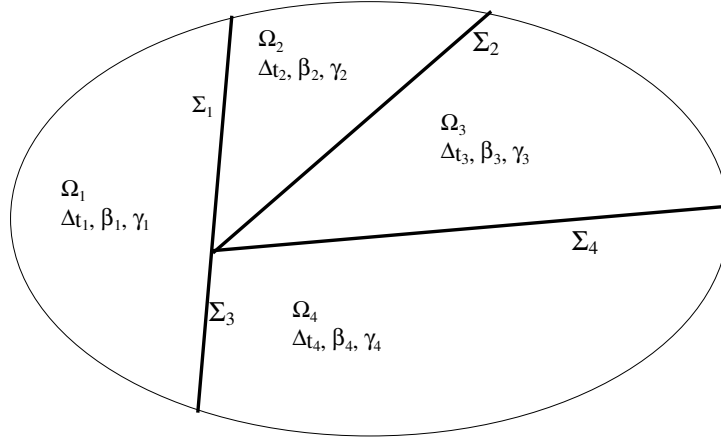


Fig. 1. Sub-domains and interfaces discretisation.

Fig. 1) and to use for each sub-domain the adequate mesh and the appropriate time integrator.

The problem is solved independently on each sub-domain then globally on the interfaces using Lagrange multiplier technique. This method allows to choose the appropriate numerical scheme (implicit or explicit) for each sub-domain as well as parallel processing of all sub-domains. The presented method is deduced from the observation of the discretised equilibrium equations of one sub-domain (denoted k) to be solved at any time:

$$M_k \ddot{\underline{u}}_k = \underline{F}_k^{\text{ext}} - \underline{F}_k^{\text{int}} + \underline{F}_k^{\text{link}} \quad \forall k \in \{1, s\} \quad (1.1)$$

In Eq. (1.1) M_k is the mass matrix of the sub-domain k , $\ddot{\underline{u}}_k$ the acceleration vector $\underline{F}_k^{\text{ext}}$ the applied load $\underline{F}_k^{\text{int}}$ the internal loads and $\underline{F}_k^{\text{link}}$ the reaction loads for the constrained displacements coming from the other sub-domains. Let us suppose that we know the forces $\underline{F}_k^{\text{link}}$. We can observe on Eq. (1.1) that the acceleration vector $\ddot{\underline{u}}_k$ is the sum of two terms: an acceleration $\ddot{\underline{u}}_k^{\text{free}}$ obtained from internal and external loads alone, and an acceleration $\ddot{\underline{u}}_k^{\text{link}}$ obtained from the displacement constraints. Eq. (1.2) gives the equations for each of these accelerations:

$$\begin{cases} \ddot{\underline{u}}_k = \ddot{\underline{u}}_k^{\text{free}} + \ddot{\underline{u}}_k^{\text{link}} \\ M_k \ddot{\underline{u}}_k^{\text{free}} = \underline{F}_k^{\text{ext}} - \underline{F}_k^{\text{int}} \\ M_k \ddot{\underline{u}}_k^{\text{link}} = \underline{F}_k^{\text{link}} \end{cases} \quad (1.2)$$

Let us suppose that the links equations are linear for example:

$$C \underline{u} = \underline{0} \quad C = \sum_{k=1, s} C_k \quad C_k = \sum_{i=1, l} C_k^i \quad (1.3)$$

where C is a constraint matrix, which links all sub-domains together and \underline{u} is the displacement vector for all sub-domains. The same equation is valid for the velocities $\dot{\underline{u}}$ and the accelerations $\ddot{\underline{u}}$. Let us now introduce one

Lagrange multiplier for each constraint and denote $\underline{\Lambda}$ the associated constraint vector. The equilibrium equations of the coupled system are obtained from the expression of the augmented Lagrangian of the system. This augmented Lagrangian is the sum of the total energy of each sub-domain augmented by an interface energy. This Lagrangian is given by Eq. (1.4)

$$L = \sum_{k=1}^s L_k + \sum_{i=1}^l \int_{\Sigma_i} \underline{\Lambda}' C \underline{w} dA = \sum_{k=1}^s L_k + L_{\text{links}} \quad (1.4)$$

The interface energy is the product of the Lagrange multiplier by constraint equation. If the constraints are permanent, we can choose to constraint any of the cinematic quantity (displacement, speed or acceleration). Hence can \underline{w} represents either the displacements or the velocities or the accelerations. For each choice of the constraint the Lagrange multiplier signification will change but the total Lagrangian will remain the same. If we impose continuity of the acceleration on the interface, it can be shown, for an explicit integration scheme, that the accelerations $\ddot{\underline{u}}_k^{\text{link}}$ are given by the following equations:

$$\ddot{\underline{u}}_k^{\text{link}} = M_k^{-1} C_k^t \underline{\Lambda} \quad (1.5.1)$$

$$\underline{\Lambda} = -H^{-1} C \ddot{\underline{u}}^{\text{free}} \quad \text{with } H = CM^{-1}C^t \quad (1.5.2)$$

In Eq. (1.5.2) the M matrix is the mass matrix for all sub-domains which is block diagonal, C is the constraint matrix for all sub-domains and $\underline{\Lambda}$ is the Lagrange multiplier vector. Eq. (1.5.2) is very interesting: it shows that the Lagrange multiplier vector is computed using only the $\ddot{\underline{u}}_k^{\text{free}}$ accelerations. Hence the solution of a constrained problem can be obtained in four steps:

- (a) compute the solution of the unconstrained problem for each sub-domain giving $\ddot{\underline{u}}_k^{\text{free}}$

- (b) compute the Lagrange multipliers \underline{A} for the global condensed problem using Eq. (1.5.2)
- (c) compute the constraint accelerations $\underline{\ddot{u}}_k^{\text{link}}$ using Eq. (1.5.1))
- (d) sum $\underline{\ddot{u}}_k^{\text{free}}$ and $\underline{\ddot{u}}_k^{\text{link}}$ to get the accelerations

This strategy is the key point of the whole method developed in the paper. It can be used with any Newmark type numerical scheme and for linear as well as non-linear problems.

The Newmark scheme approximates the evolution with time of displacements and speeds during the time step Δt by the following equations governed by and the two parameters $\beta\gamma$:

$$\begin{cases} \underline{u}^{n+1} = \underline{u}^n + \Delta t \underline{\dot{u}}^n + \frac{\Delta t^2}{2} (1 - 2\beta) \underline{\ddot{u}}^n + \Delta t^2 \beta \underline{\ddot{u}}^{n+1} \\ \underline{\dot{u}}^{n+1} = \underline{\dot{u}}^n + \Delta t (1 - \gamma) \underline{\ddot{u}}^n + \Delta t \gamma \underline{\ddot{u}}^{n+1} \end{cases} \quad (1.6)$$

In Eq. (1.6) n represents the number of the time step. The observation of Eq. (1.6) shows that the displacement or velocity at time step $n + 1$ is the sum of a predictor part obtained directly from the values of displacement, speed and acceleration at previous time step n and of a corrector term depending on the accelerations at time step $n + 1$, $\underline{\ddot{u}}^{n+1}$, which can be considered as the unknown of the problem. In the following we shall omit the symbols $n + 1$.

$$\begin{cases} \underline{u} = {}^p \underline{u} + \Delta t^2 \beta \underline{\ddot{u}} \\ \underline{\dot{u}} = {}^p \underline{\dot{u}} + \Delta t \gamma \underline{\ddot{u}} \end{cases} \quad (1.7)$$

where the predictors are given by

$$\begin{cases} {}^p \underline{u} = \underline{u}^n + \Delta t \underline{\dot{u}}^n + \frac{\Delta t^2}{2} (1 - 2\beta) \underline{\ddot{u}}^n \\ {}^p \underline{\dot{u}} = \underline{\dot{u}}^n + \Delta t (1 - \gamma) \underline{\ddot{u}}^n \end{cases} \quad (1.8)$$

We shall first show how one can couple different sub-domains each with its own Newmark time-integration scheme but all with the same time step.

In a second part we shall show how the scheme can be developed with different time steps in each sub-domain: this method will be called a non-matching time algorithm.

In the third part we shall extend the method to non-matching meshes of the sub-domains.

2. How to couple different Newmark schemes in each sub-domain

Let us now suppose that we have the same time step Δt for each sub-domain. Each sub-domain k has its own Newmark time integrator with parameters $\beta_k \gamma_k$. We will now give the algorithm to couple the sub-domains.

2.1. Method

The discretised dynamic equilibrium equations of the whole problem can be written for a given time step $n + 1$:

$$\begin{bmatrix} \alpha_1 \tilde{M}_1 & 0 & 0 & -\alpha_1 C_1^t \\ 0 & \dots & 0 & \dots \\ 0 & 0 & \alpha_s \tilde{M}_s & -\alpha_s C_s^t \\ -\alpha_1 C_1 & \dots & -\alpha_s C_s & 0 \end{bmatrix} \begin{Bmatrix} \underline{\ddot{u}}_1 \\ \dots \\ \underline{\ddot{u}}_s \\ \underline{A} \end{Bmatrix} = \begin{Bmatrix} \alpha_1 (F_1^{\text{ext}} - {}^p F_1^{\text{int}}) \\ \dots \\ \alpha_s (F_s^{\text{ext}} - {}^p F_s^{\text{int}}) \\ \sum_{k=1}^s C_k^p w_k \end{Bmatrix} \quad (2.1)$$

\tilde{M}_k is the following matrix:

$$\tilde{M}_k = M_k + \beta_k \Delta t^2 K_k \quad (2.2)$$

The α_k parameter and given vector ${}^p w_k$ are depending of the choice of the quantity which is constrained at the interface. Eq. (2.3) give their values in case of acceleration constraint, (2.4) in case of velocity constraint, and (2.5) in case of displacement constraint:

$$\alpha_k = 1, \quad \underline{w} = \underline{\ddot{u}} \quad (2.3)$$

$$\alpha_k = \gamma_k \Delta t, \quad \underline{w} = \underline{\dot{u}} \quad (2.4)$$

$$\alpha_k = \beta_k \Delta t^2, \quad \underline{w} = \underline{u} \quad (2.5)$$

The Eq. (2.1) can be solved directly. But we want to take advantage of the decomposition in sub-domains and the direct method would not be efficient because the matrix is full with the chosen numbering of the unknowns. We introduce the splitting of the cinematic unknowns in free and link part in Eq. (2.1). We get the equation giving $\underline{\ddot{u}}^{\text{free}}$:

$$\begin{bmatrix} \tilde{M}_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \tilde{M}_s \end{bmatrix} \begin{Bmatrix} \underline{\ddot{u}}_1^{\text{free}} \\ \dots \\ \underline{\ddot{u}}_s^{\text{free}} \end{Bmatrix} = \begin{Bmatrix} (F_1^{\text{ext}} - {}^p F_1^{\text{int}}) \\ \dots \\ (F_s^{\text{ext}} - {}^p F_s^{\text{int}}) \end{Bmatrix} \quad (2.6)$$

As can be seen from (2.6) the resolution of $\underline{\ddot{u}}^{\text{free}}$ can be performed independently for each sub-domain k . For the link part we get:

$$\begin{bmatrix} \alpha_1 \tilde{M}_1 & 0 & 0 & -\alpha_1 C_1^t \\ 0 & \dots & 0 & \dots \\ 0 & 0 & \alpha_s \tilde{M}_s & -\alpha_s C_s^t \\ -\alpha_1 C_1 & \dots & -\alpha_s C_s & 0 \end{bmatrix} \begin{Bmatrix} \underline{\ddot{u}}_1^{\text{link}} \\ \dots \\ \underline{\ddot{u}}_s^{\text{link}} \\ \underline{A} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \dots \\ 0 \\ \sum_{k=1}^s C_k^p w_k + \sum_{k=1}^s \alpha_k C_k \underline{\ddot{u}}_k^{\text{free}} \end{Bmatrix} \quad (2.7)$$

We can first observe that the last line of the right hand side of Eq. (2.7) can be written as

$$\sum_{k=1}^s C_k (\underline{p} \underline{w}_k + \alpha_k \underline{\dot{u}}^{\text{free}}) = \sum_{k=1}^s C_k \underline{w}^{\text{free}} \quad (2.8)$$

Eq. (2.7) can then be condensed on the multipliers and we get:

$$H \underline{A} = - \sum_{k=1}^s C_k \underline{w}_k^{\text{free}} \quad \text{with } H = \sum_{k=1}^s \alpha_k C_k \tilde{M}_k^{-1} C_k^t \quad (2.9)$$

$$\underline{\dot{u}}_k^{\text{link}} = M_k^{-1} C_k^t \underline{A} \quad (2.10)$$

$$\begin{cases} \underline{\dot{u}}_k = \underline{p} \underline{u}_k + \gamma_k \Delta t (\underline{\dot{u}}_k^{\text{free}} + \underline{\dot{u}}_k^{\text{link}}) \\ \underline{u}_k = \underline{p} \underline{u}_k + \beta_k \Delta t^2 (\underline{\dot{u}}_k^{\text{free}} + \underline{\dot{u}}_k^{\text{link}}) \end{cases} \quad (2.11)$$

For linear problems, the internal forces are explicit functions of the predicted displacements, and the solution for one time step is then obtained in five successive steps:

- solve for the free solution $\underline{\dot{u}}^{\text{free}}$ using Eq. (2.6)
- compute the multiplier vector \underline{A} using Eq. (2.9)
- solve for the link solution $\underline{\dot{u}}^{\text{link}}$ using Eq. (2.10)
- sum the two contributions to have the whole solution $\underline{\dot{u}} = \underline{\dot{u}}^{\text{free}} + \underline{\dot{u}}^{\text{link}}$
- update the displacements and velocities using Eq. (2.11)

All steps except step b can be performed in parallel on each sub-domain k .

2.2. Stability and precision conditions for linear systems

We suppose that we respect the stability conditions for time integration numerical scheme in each sub-domain. Under which condition the proposed solution for coupling the sub-domains do not destroy the stability of the whole system? To achieve this study we have used Hughes “energy” method [2] for evaluation of stability. We extend this method to the case of sub-domains. The energy will be augmented by the interface contributions E_{links} coming from L_{links} in the augmented Lagrangian. It is clear that if the interface energy is positive the introduction of the interfaces will destroy the stability of the numerical schemes of the sub-domains and hence of the whole system. If it is negative, some numerical “damping” will be introduced by the interface modelisation. The optimal choice is the one for which the interface energy is zero. If this choice is made, the stability as well as the precision of the numerical schemes are preserved for the whole model. This means that if we have chosen schemes which are second order accurate in each of the sub-domains, the coupled problem remains second order accurate if the interface energy is zero. As explained in [2–4] the interface energy is zero only under the fol-

lowing conditions:

Acceleration constraints:	$E_{\text{link}} = 0$ if γ_k is the same $\forall k$
Velocity constraints:	$E_{\text{link}} = 0 \forall \beta_k \forall \gamma_k$
Displacement constraint:	$E_{\text{link}} = 0$ if $2\beta_k - \gamma_k = 0 \forall k$ and $\underline{\dot{u}}_k$ are continuous with time

These results are very interesting and show that to impose the cinematic constraints on the velocities are the best choice because it allows the coupling between any Newmark scheme: for instance one can couple an implicit scheme with an explicit one.

The coupling via the accelerations implies that all schemes have the same γ Newmark parameter.

If we impose cinematic displacement constraint (as for example in the well-known FETI algorithm [6]) all schemes must verify the condition $2\beta - \gamma = 0$. This is why FETI method cannot be employed to couple explicit and implicit sub-domains. The second condition gives an explanation why the unilateral contact treatment by Lagrange multiplier with displacement control induce very large oscillations in the response.

2.3. Extension to non-linear cases

The algorithm given in the previous paragraph for linear approximation can be extended to non-linear cases with constant time steps, as shown in [1]. The stability and precision demonstrations given in the preceding section still hold but one has no proof of stability and precision for Newmark schemes in non-linear cases. Hence the stability and precision of the proposed scheme cannot be demonstrated for the coupled scheme.

In case of implicit non-linear sub-domains it is necessary to iterate for the computation of the acceleration vector $\underline{\dot{u}}$. Let us call i the iteration number. Eq. (2.6) becomes for iteration $i + 1$:

$$\begin{bmatrix} \tilde{M}_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \tilde{M}_s \end{bmatrix} \begin{Bmatrix} {}^{i+1} \underline{\dot{u}}_1^{\text{free}} \\ \dots \\ {}^{i+1} \underline{\dot{u}}_s^{\text{free}} \end{Bmatrix} = \begin{Bmatrix} (F_1^{\text{ext}} - {}^i F_1^{\text{int}} + \beta_1 \Delta t^2 {}^i \underline{\dot{u}}_1) \\ \dots \\ (F_s^{\text{ext}} - {}^i F_s^{\text{int}} + \beta_s \Delta t^2 {}^i \underline{\dot{u}}_s) \end{Bmatrix} \quad (2.12)$$

The other equations remain the same. The algorithm is then the following in case of non-follower loads (for one given time step):

- initialize the acceleration vector ${}^1 \underline{\dot{u}}$ for instance with the accelerations of previous time step, the iteration counter to 1; compute the external loads for final time.

- (a) compute final stresses and internal loads for iteration i
- (b) check small enough residual loads. If yes go to (j)
- (c) compute ${}^{i+1}\underline{\dot{u}}^{\text{free}}$ using Eq. (2.6)
- (d) compute multipliers with Eq. (2.9)
- (f) solve for the link solution ${}^{i+1}\underline{\dot{u}}^{\text{link}}$ using Eq. (2.10)
- (g) sum the two contributions to have the whole solution $\underline{\dot{u}} = \underline{\dot{u}}^{\text{free}} + \underline{\dot{u}}^{\text{link}}$
- (h) update the displacements and velocities using Eq. (2.11)
- (i) $i = i + 1$ and go to (a)
- (j) next time step

3. Non-matching time steps in the sub-domains

For the clarification of the paper we shall hereafter suppose that we have only two sub-domains A and B . The method can be used for more than two sub-domains as can be observed from the application example but it shall not be developed in this paper. We shall now suppose that the time steps in the sub-domains are different but that the time step ΔT for sub-domain A is a multiple of time step Δt for sub-domain B . The relation is supposed to be

$$\Delta T = n\Delta t \quad (3.1)$$

We define two times scales: the micro-time scale denoted m which, corresponds to time step Δt and the macro-time scale denoted M , which corresponds, to time step ΔT . We now want to know under which conditions we can still couple the dynamic computations in the two sub-domains. We shall suppose that we have chosen a velocity constraint to couple the sub-domains A and B .

3.1. The method

We will consider now one macro-time step ΔT . If we impose a velocity constraint on the micro-scale we shall have to compute one acceleration vector $\underline{\ddot{u}}_A$ for sub-domain A , n acceleration vectors $\underline{\ddot{u}}_B^m$ for sub-domain B and n Lagrange multiplier vectors $\underline{\Lambda}^m$ for the interface. In order to be able to compute the solution for the sub-domain B we need to know the constraints coming from sub-domain A . The Lagrangian of the two scale problem shall be:

$$L = L_A^M + L_B^m \quad (3.2)$$

We now add a kinematic constraint to link the macro and the micro-time scales. Let us introduce the constraint that the velocities are the same on the interface on the micro-time scale level we obtain:

$$\begin{aligned} L &= L_A^M + L_B^m + (\underline{\Lambda}^m)^T (C_A \underline{\dot{u}}_A^m + C_B \underline{\dot{u}}_B^m) \\ &= L_A^M + L_B^m + (\underline{\Lambda}^m)^T C_A \underline{\dot{u}}_A^m + (\underline{\Lambda}^m)^T C_B \underline{\dot{u}}_B^m \end{aligned} \quad (3.3)$$

Let us suppose that we can linearly interpolate the velocities and the multipliers using the linear time interpolator P to pass from the values at the macro-times M to the values at the micro-time m :

$$\begin{cases} \underline{\dot{u}}_A^m = P \underline{\dot{u}}_A^M \\ \underline{\Lambda}_A^m = \bar{P}^T \underline{\Lambda}_A^M \end{cases} \quad (3.4)$$

For example P operator can be defined by the following equation:

$$\underline{\dot{u}}_A^m = \underline{\dot{u}}_A^0 \left(1 - \frac{m}{n}\right) + \underline{\dot{u}}_A^m \frac{m}{n} \quad (3.5)$$

Using first line of (3.4) we obtain:

$$L = L_A^M + L_B^m + (\underline{\Lambda}^m)^T P C_A \underline{\dot{u}}_A^M + (\underline{\Lambda}^m)^T C_B \underline{\dot{u}}_B^m \quad (3.6)$$

Using second line we get:

$$L = L_A^M + L_B^m + (\underline{\Lambda}_A^M)^T C_A \underline{\dot{u}}_A^M + (\underline{\Lambda}_B^m)^T C_B \underline{\dot{u}}_B^m \quad (3.7)$$

Eq. (3.7) is the coupled multi-scale dynamic augmented Lagrangian. If relation (3.7) is verified, the equilibrium on the macro-scale (obtained by the verification of the Lagrangian when $m = n$ i.e. for time ΔT) is obtained as well as the continuity of the velocities at the interface on the micro-scale. In case of interpolation operator P defined by Eq. (3.5) the extrapolation of Lagrange multipliers from micro to macro-scale is given by

$$\underline{\Lambda}^M = \underline{\Lambda}^0 \left(1 - \frac{n}{m}\right) + \underline{\Lambda}^m \frac{n}{m} \quad (3.8)$$

Let us now express the constraint equations. Replacing the micro-velocity on A by the interpolated values given by (3.5) we obtain as expression of the cinematic constraint on the velocities:

$$C_A \underline{\dot{u}}_A^0 \left(1 - \frac{m}{n}\right) + \frac{m}{n} C_A \underline{\dot{u}}_A^m + C_B \underline{\dot{u}}_B^m = 0. \quad (3.9)$$

In Eq. (3.9) we use the velocities of sub-domain A at the beginning of time step ΔT denoted $\underline{\dot{u}}_A^0$. If we now use the expressions of the velocities deduced of the Newmark scheme of each sub-domain we obtain:

$$\begin{aligned} &\left(1 - \frac{m}{n}\right) C_A \underline{\dot{u}}_A^0 + \frac{m}{n} \left(C_A^p \underline{\dot{u}}_A^M + \gamma_A n \Delta t C_A \underline{\dot{u}}_A^M\right) \\ &+ C_B^p \underline{\dot{u}}_B^m + \gamma_B \Delta t C_B \underline{\ddot{u}}_B^m = 0. \end{aligned} \quad (3.10)$$

Using now the partition of acceleration if free and link part we get for micro-time step m :

$$\begin{aligned} &\left(1 - \frac{m}{n}\right) C_A \underline{\dot{u}}_A^0 + \frac{m}{n} \left(C_A^p \underline{\dot{u}}_A^M + \gamma_A n \Delta t C_A \underline{\dot{u}}_A^{M:\text{free}}\right) \\ &+ \gamma_A m \Delta t C_A \underline{\dot{u}}_A^{M:\text{link}} + C_B^p \underline{\dot{u}}_B^m + \gamma_B \Delta t C_B \underline{\dot{u}}_B^{m:\text{free}} \\ &+ \gamma_B \Delta t C_B \underline{\ddot{u}}_B^{m:\text{link}} = 0. \end{aligned} \quad (3.11)$$

Replacing the link acceleration in function of the Lagrange multiplier (Eq. (2.10)) we obtain:

$$\begin{aligned} & \left(1 - \frac{m}{n}\right) C_A \dot{\underline{u}}_A^0 + \frac{m}{n} \left(C_A^P \dot{\underline{u}}_A^M + \gamma_A n \Delta t C_A \ddot{\underline{u}}_A^{M\text{free}} \right) \\ & + \gamma_A m \Delta t C_A M_A^{-1} C_A^T \underline{A}^M + C_B^P \dot{\underline{u}}_B^m + \gamma_B \Delta t C_B \ddot{\underline{u}}_B^{m\text{free}} \\ & + \gamma_B \Delta t C_B M_B^{-1} C_B^T \underline{A}^m = 0. \end{aligned} \quad (3.12)$$

Using the relation between macro- and micro-time Lagrange multipliers (3.8) for $n = m$ we get:

$$\begin{aligned} & \left(1 - \frac{m}{n}\right) C_A \dot{\underline{u}}_A^0 + \frac{m}{n} \left(C_A^P \dot{\underline{u}}_A^M + \gamma_A n \Delta t C_A \ddot{\underline{u}}_A^{M\text{free}} \right) \\ & + \gamma_A \Delta t C_A M_A^{-1} C_A^T \underline{A}^m + C_B^P \dot{\underline{u}}_B^m + \gamma_B \Delta t C_B \ddot{\underline{u}}_B^{m\text{free}} \\ & + \gamma_B \Delta t C_B M_B^{-1} C_B^T \underline{A}^m = 0. \end{aligned} \quad (3.13)$$

If we now call $\dot{\underline{u}}_{\text{free}}$ the estimated free speeds given by

$$\begin{cases} \dot{\underline{u}}_A^{M\text{free}} = \dot{\underline{u}}_A^M + \gamma_A n \Delta t \ddot{\underline{u}}_A^{M\text{free}} \\ \dot{\underline{u}}_B^{m\text{free}} = \dot{\underline{u}}_B^m + \gamma_B \Delta t \ddot{\underline{u}}_B^{m\text{free}} \end{cases} \quad (3.14)$$

we obtain:

$$\begin{aligned} & \left(1 - \frac{m}{n}\right) C_A \dot{\underline{u}}_A^0 + \frac{m}{n} C_A \dot{\underline{u}}_A^{M\text{free}} + \gamma_A \Delta t C_A M_A^{-1} C_A^T \underline{A}^m \\ & + C_B \dot{\underline{u}}_B^{m\text{free}} + \gamma_B \Delta t C_B M_B^{-1} C_B^T \underline{A}^m = 0. \end{aligned} \quad (3.15)$$

Noting that we can write:

$$\dot{\underline{u}}_A^{m\text{free}} = \left(1 - \frac{m}{n}\right) \dot{\underline{u}}_A^0 + \frac{m}{n} \dot{\underline{u}}_A^{M\text{free}} \quad (3.16)$$

We finally obtain:

$$\begin{aligned} & C_A \dot{\underline{u}}_A^{m\text{free}} + \gamma_A \Delta t C_A M_A^{-1} C_A^T \underline{A}^m + C_B \dot{\underline{u}}_B^{m\text{free}} \\ & + \gamma_B \Delta t C_B M_B^{-1} C_B^T \underline{A}^m = 0. \end{aligned} \quad (3.17)$$

Let us now observe carefully Eq. (3.17) giving the Lagrange multipliers for one micro-time step m . We first observe that we need to know $\dot{\underline{u}}_A^{m\text{free}}$. This value is very easy to get: it is obtained by a linear interpolation using Eq. (3.16) and the known values of $\dot{\underline{u}}_A^0$ and $\dot{\underline{u}}_A^{M\text{free}}$. We also need to know the predicted velocities as well as the free accelerations on sub-domain B at time step m to compute the Lagrange multipliers. The final solution for sub-domain A is hence obtained by a recurrence algorithm. For first micro-time step we get the Lagrange multipliers \underline{A}^1 and deduce the accelerations velocities and displacements for sub-domain B and the end of time step one. Repeating these operations n times, we finally obtain the solution at the macro-time step. Let us also remark that the computation of Lagrange multipliers \underline{A}^m necessitates the solution of a linear system whose matrix (H) is constant all over the macro-time step and is proportional to Δt change if the n ratio is constant. This property is interesting because it allows to invert the H matrix only once for all the transient analysis. This matrix being some times very large this property is of prime importance for numerical efficiency of the proposed procedure. Eq. (3.17) gives the problem to be solved for Lagrange multiplier computation:

$$\begin{cases} H \underline{A}^m = -\underline{B} \\ H = \Delta t (n \gamma_A C_A M_A^{-1} C_A^T + \gamma_B C_B M_B^{-1} C_B^T) \\ \underline{B} = C_A \dot{\underline{u}}_A^{m\text{free}} + C_B \dot{\underline{u}}_B^{m\text{free}} \end{cases} \quad (3.18)$$

The algorithm can finally be described by the following flowchart in case of constant n and Δt values.

- (a) compute H and invert it once for all.
- (b) Loop over Macro-time steps ΔT
 - (1) compute $\dot{\underline{u}}_A^M$, $\ddot{\underline{u}}_A^{M\text{free}}$ and $\dot{\underline{u}}_A^{M\text{free}}$ using Eqs. (2.6) and (3.14)
 - (2) loop over micro-time steps Δt : for $0 < m < n + 1$
 - (21) compute $\dot{\underline{u}}_B^m$, $\ddot{\underline{u}}_B^{m\text{free}}$ and $\dot{\underline{u}}_B^{m\text{free}}$ using Eqs. (2.6) and (3.14)
 - (22) compute $\dot{\underline{u}}_A^{m\text{free}}$ using Eq. (3.16)
 - (22) compute \underline{A}^m using Eq. (3.18)
 - (23) compute $\dot{\underline{u}}_B^m$, $\dot{\underline{u}}_B^m$ and $\dot{\underline{u}}_b^m$
 - (3) endloop
 - (4) compute $\dot{\underline{u}}_A^M$, $\dot{\underline{u}}_A^M$ and $\dot{\underline{u}}_A^M$ using $\underline{A}^m = \underline{A}^M$ obtained at the end of loop (2) to (3)
- (c) endloop

Let us quote that this algorithm immediately extends to non-linear behaviour of sub-domains A and B in the case of explicit time integration. In this case the internal loads evaluation only requires the knowledge of the predictors of velocities and displacements and no iterations are required for equilibrium checks.

In case of implicit non-linear time integrator an iteration loop is necessary to ensure zero residual loads. As we only ensure the equilibrium on the interfaces at the macro-scale only it shall be sufficient to iterate for equilibrium on the macro-scale level. Hence the algorithm can be directly deduced from that given for the sub-domains with equal time steps. For each macro-time step we shall have to:

- (0) initialize the acceleration vector $\dot{\underline{u}}$ for instance with the accelerations of previous macro-time step, the iteration counter i to 1, compute the external loads for final time.
- (a) compute final stresses and internal loads for iteration i
- (b) check small enough residual loads on the macro-scale. If yes go to (k)
- (c) compute $\dot{\underline{u}}_A^{i+1;M\text{free}}$ and $\dot{\underline{u}}_B^{i+1;1}$ using Eq. (2.12)
- (d) compute $\dot{\underline{u}}_A^{i+1;M\text{free}}$ using Eq. (3.14)
- (e) loop over micro-time steps
 - (1) compute $\dot{\underline{u}}_B^{i+1;1}$, $\dot{\underline{u}}_B^{i+1;M\text{free}}$ and $\dot{\underline{u}}_B^{i+1;M\text{free}}$ using Eqs. (2.6) and (3.14)
 - (2) compute $\dot{\underline{u}}_A^{i+1;M\text{free}}$ using Eq. (3.16)
 - (3) compute $\underline{A}^{i+1;M}$ using Eq. (3.18)
 - (4) compute $\dot{\underline{u}}_B^{i+1;1}$, $\dot{\underline{u}}_B^{i+1;M}$ and $\dot{\underline{u}}_b^{i+1;M}$
- (f) endloop

- (g) compute \underline{u}_A^{i+1M} , $\dot{\underline{u}}_A^{i+1M}$ and $\ddot{\underline{u}}_A^{i+1A}$ using $\underline{A}^{i+1} = \underline{A}^{i+1M}$ obtained at the end of loop (f)
 (h) $i = i + 1$ go to (a)
 (k) update the displacements velocities stresses internal variables for plasticity

3.2. Stability conditions

For this case the same method as in paragraph 2.2 can be used to study the stability of the proposed scheme. It has been shown in [4] that the proposed method to couple the sub-domains with a velocity constraint does not destroy the stability of the numerical scheme in each sub-domain. In fact it has been shown that the interface modelisation introduces a little dissipation in the system.

3.3. Varying time steps

The results obtained in the preceding sections easily extend to the case where the small time steps Δt are changing with time. We just have to reduce the last time step such that it ends at the same time as the large time step ΔT . This result is important for the application of the method to explicit non-linear computations where the time step has to be changed very often but also for implicit cases with severe non-linearities where the steps have to adapt to ensure convergence of the iterative process.

If the large time steps ΔT are not constant the link energy has to be slightly modified. The following equation taking into account the change of one large time step ΔT to the next one has to be added. Let us suppose that we have just got the results at the end of time step ΔT_A^1 and that we want the solution at the end of the time step ΔT_A^2 . Eq. (3.18) becomes:

$$\begin{cases} H \underline{A}^m = -\underline{B} \\ H = (\Delta T_A^2 \gamma_A C_A M_A^{-1} C_A^T + \Delta t_B \gamma_B C_B M_B^{-1} C_B^T) \\ \underline{B} = C_A \dot{\underline{u}}_A^{m, \text{free}} + C_B \dot{\underline{u}}_B^{m, \text{free}} + C_A (1 - \alpha_A) (\beta_A - 1) \frac{\Delta T_A^2}{2} \gamma_{\text{link}} \end{cases} \quad (3.19)$$

where we have:

$$\begin{cases} \beta_A = \frac{\Delta T_A^1}{\Delta T_A^2} \\ \alpha_A = \frac{t_B - T_A^1}{\Delta T_A^2} \end{cases} \quad (3.20)$$

4. Non-matching meshes

The mortar method has been widely used to couple sub-domains with non-matching interfaces in case of static linear problems. We can use it directly for dynamic non-linear problems. We also propose here an alternative formulation which ensures exact cinematic conti-

nity at the interface. This new method shall hence ensure no dissipation at interface due to cinematic constraint, but shall introduce some over stiffening of the interface in case of very crude interface meshing. On the other hand the mortar method does not ensure perfect cinematic continuity at the interface so that some energy will be dissipated at the interface, but it will never over constraint the interfaces.

4.1. Principle of the method

We shall introduce Lagrange multipliers to apply the constraints. The interface between the sub-domains shall be discretised with three surfaces having the same geometry but three different meshes: one for the “left” sub-domain S^{left} , one for the “right” one S^{right} and one for the Lagrange multiplier surface S^{λ} .

Let us suppose that each of these three surface has its own discretisation coming from the mesh of the sub-domains. We shall simply describe the unknowns variation by using nodal values (n for the right sub-domain surface, m for the left one and p for the Lagrange multiplier function) and a set of shape functions for each of the set of variable. We then have:

$$\begin{cases} \underline{u}^{\text{right}} = \sum_{i=1}^n N_i^{\text{right}}(\xi) \underline{u}_i^{\text{right}} \\ \underline{u}^{\text{left}} = \sum_{j=1}^m N_j^{\text{left}}(\xi) \underline{u}_j^{\text{left}} \\ \underline{\lambda} = \sum_{k=1}^p P_k(\xi) \underline{\lambda}_k \end{cases} \quad (4.1)$$

Integrating on a reference surface S , the interface energy can be expressed by

$$W^{\text{interface}} = \int_S \underline{\lambda}^t \cdot (\underline{u}^{\text{right}} - \underline{u}^{\text{left}}) dS \quad (4.2)$$

Using (4.1) we obtain:

$$W^{\text{interface}} = \int_S \left[\sum_{k=1}^p P_k \underline{\lambda}_k^t \cdot \sum_{i=1}^n N_i^{\text{right}} \underline{u}_i^{\text{right}} - \sum_{j=1}^m N_j^{\text{left}} \underline{u}_j^{\text{left}} \right] dS \quad (4.3)$$

4.2. Main results

We shall now restrict ourselves to the choice of linear shape functions for the discretisation of all the unknowns. The sets of nodes for right and left surfaces are simply those supported by the discretisation of the left and right sub-domains. How do we choose the set of nodes supporting the Lagrange multipliers. Using the Ladyzenskaia–Babuska–Brezzi condition, one knows that p (the number of Lagrange multipliers nodes) should not be too large so that the condensation operator H is singular, but if it is too small the kinematic continuity of the interfaces is very poorly verified.

The mortar method consists to choose for the Lagrange multipliers nodes either the nodes belonging to the left or those belonging to the right sub-domain boundary. If the number of nodes is very different from one side to the other the choice of the set of nodes leads to a very poor kinematic continuity and a relatively large part of the energy can be absorbed in the work of the interface.

Let us now denote c the number of common nodes (i.e. the nodes having the same coordinates) between the left and right discretisation. Let us also define p by

$$p = n + m - c \quad (4.4)$$

It can be shown [7] that if we choose a Lagrange multiplier set of nodes (dimension p) which is the symmetric difference of left and right set of nodes, the kinematic continuity is exactly verified. This implies that no energy shall be dissipated in the interface. On the other hand the value of p is in general greater than the maximum of n and m , and hence this matching is stiffer than the usual mortar method. It has also been shown in [7] that p is the largest possible number of Lagrange multiplier which do not lead to locking and that the dimension of the interface displacement set is c .

Let us quote that if c equals n or m (hierarchical meshes) the proposed method is strictly equivalent that the fine mortar method, and that we have shown in that case that the exact continuity is verified.

5. Application examples

Three application examples shall be given in this part. The first one is elastic linear, the second is elastoplastic with large displacements, and the third uses non-matching meshes, in two sub-domains. (Figs. 2 and 3).

5.1. Linear case

The linear case is described just below. One applies a step-bending load at the extremity of the beam as shown in Fig. 4.

The computed response with two implicit and one explicit sub-domains with a time step ratio of 10000 is compared to the completely implicit solution (denoted reference) on Fig. 5 just below. We can observe that the solution obtained with a step ratio of 10000 is in excellent agreement with the reference solution obtained with an implicit central difference Newmark scheme.

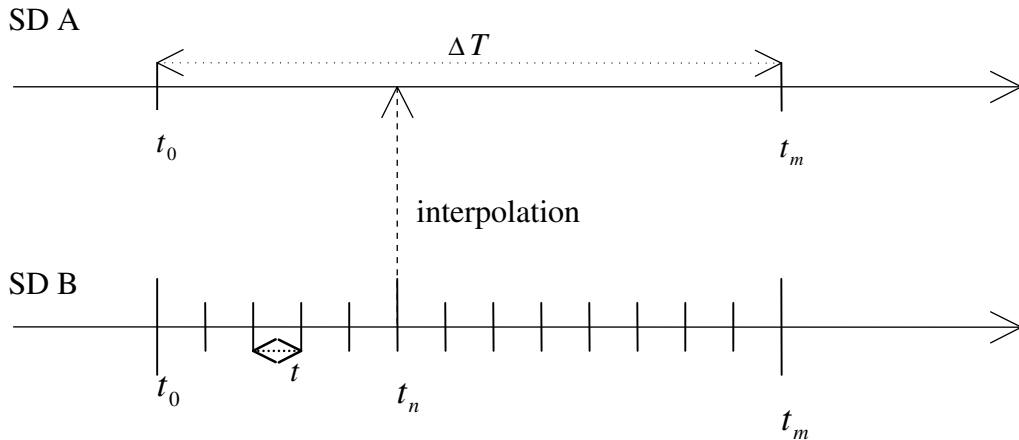


Fig. 2. Definition of one macro-step ΔT and interpolation rules.

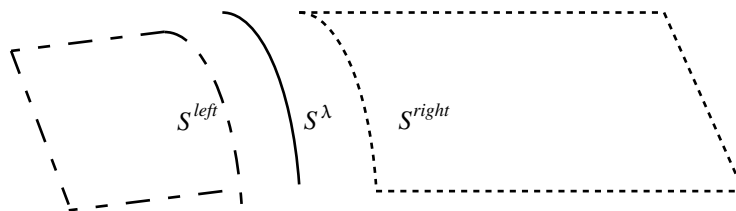


Fig. 3. Interface meshes.

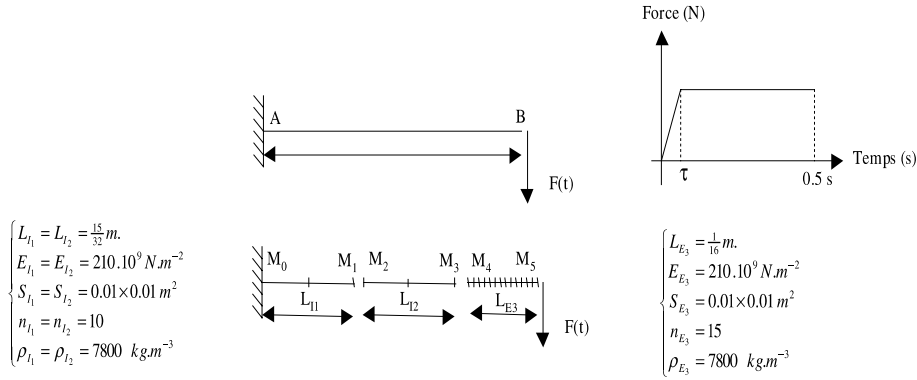


Fig. 4. Definition of linear example.

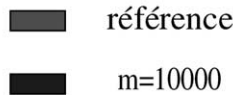
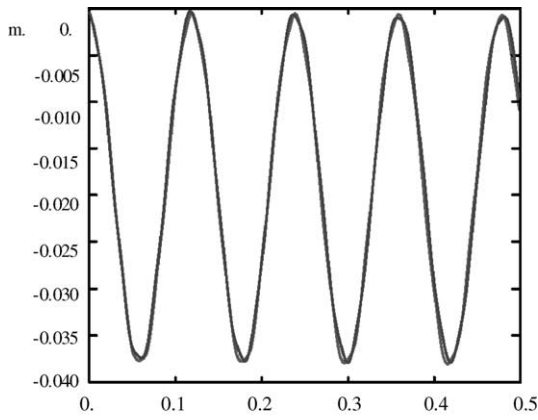


Fig. 5. Linear sub-domains for a beam under bending step load: transient axial displacement.

5.2. Non-linear case

The non-linear application problem is shown just below. We study a thin stiffened cylinder clamped at its two extremities and subjected to a step vertical load applied on the small square area on the top of the beam (Fig. 6). The material behaviour is elastoplastic with linear cinematic hardening. The mesh is done with DKT shell elements. The computation is done with six sub-domains and different time steps in the sub-domains. Clear sub-domains are implicit and dark ones are explicit: all implicit sub-domains have been chosen to have the same time step, as well as all explicit ones (Fig. 7). The ratio of time step between the sub-domains is chosen to be 10 or 100. The reference computation is done using the implicit non-linear solver of Castem 2000 code.

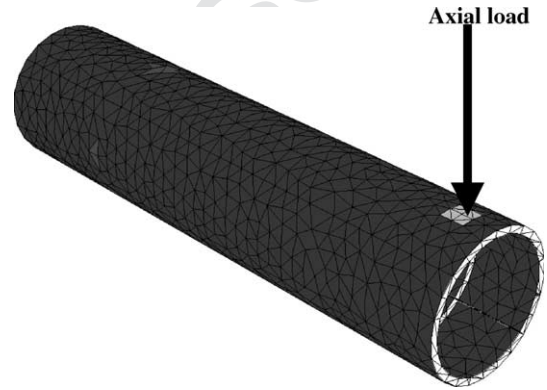
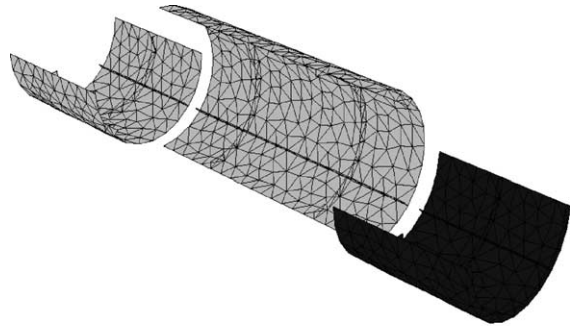


Fig. 6. Definition of non-linear problem.



(clear implicit sub-domains - behind -, dark explicit sub domains 6 sub-domains)

Fig. 7. Sub-domain decomposition for non-linear problem.

The main geometrical parameters are the following: radius $R = 1$ m, length $L = 8$ m, thickness $t = 5$ mm. The cylinder is clamped at its origin and a vertical bending load of -2×10^5 N is applied at its extremity. The Young's modulus is chosen to be 200 000 MPa the Poisson's ratio is 0.3 and the material plasticity model is a linear cinematic Von Mises model: the yield stress is chosen to be 200 MPa and the tangent modulus is

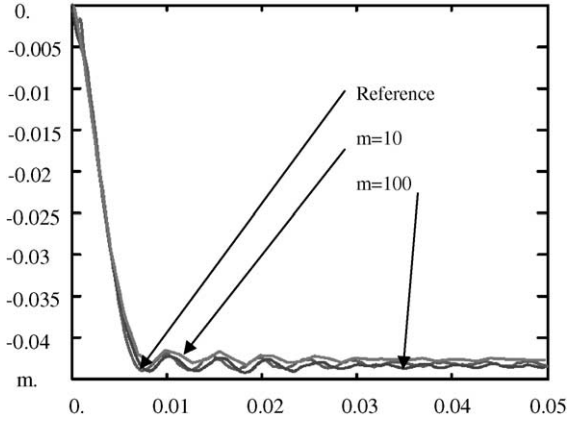


Fig. 8. Transient non-linear response.

$E t = 2000$ MPa. The computed vertical displacement of the point at the center of applied load with the sub-domain with different steps is compared to the reference solution on Fig. 8 below. The quality of the prediction is very good. The efficiency ratio is also quite good because one needs only one quarter of the time to get the solution with the sub-domains decomposition (Fig. 9).

5.3. Non-matching meshes case

The problem is once more the bending of an elastic plane stress beam. The beam is clamped at its origin and its length is $L = 10$ m, its width is $t = 1$ m. The dimen-

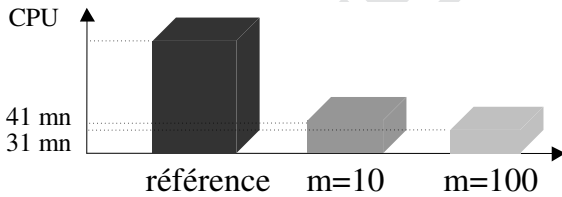


Fig. 9. Transient non-linear response.

sion perpendicular to the plane is denoted b . The Young's modulus is denoted E and the Poisson's ratio is chosen to be 0.3. One applies a vertical step load at its extremity P . The beam is decomposed in two non-matching meshes as shown on Fig. 10 below.

The elements used are 4 node quadrilateral elements. The right sub-domain has 2420 elements and 22 across the thickness whereas the left one has 980 elements and 14 across the thickness. For our case we have:

$$n = 15, \quad m = 23, \quad p = 35, \quad c = 3 \quad (5.1)$$

The computation of the transient response is done using an implicit mean acceleration Newmark scheme ($\beta = 1/4, \gamma = 1/2$) and only one time step in all the sub-domains. The computed maximum displacement $w_{\max}^{\text{computed}}$ is compared to the analytical maximum vertical displacement at the extremity given by the equation.

$$w_{\max}^{\text{analytical}} = \frac{8PL^3}{Ebt^3} \quad (5.2)$$

The following table shows the ratios obtained for the different technique used to link the sub-domains. (Table 1).

It shows that on this case the coarse mortar method leads to a dissipation of energy in the interface, and that our method gives similar results to that of fine mortar choice. The energy dissipated at the interface with our method is strictly zero. The energy dissipated with the fine mortar method has been evaluated to be very small in this case. The transient response is given in Fig. 11 for the three cases.

Table 1
Non-matching interface schemes comparison

Type of method used	$\frac{w_{\max}^{\text{computed}}}{w_{\max}^{\text{analytical}}}$	$\frac{\text{Max interface energy}}{\text{Max kinetic energy}}$
Our method $p = 35$	0.998	0
Fine mortar $p = 23$	0.999	1.E-8
Coarse mortar $p = 15$	0.79	0.2

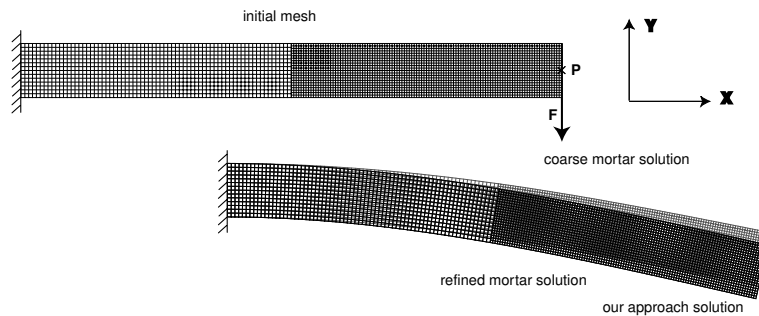


Fig. 10. Non-matching interface mesh.

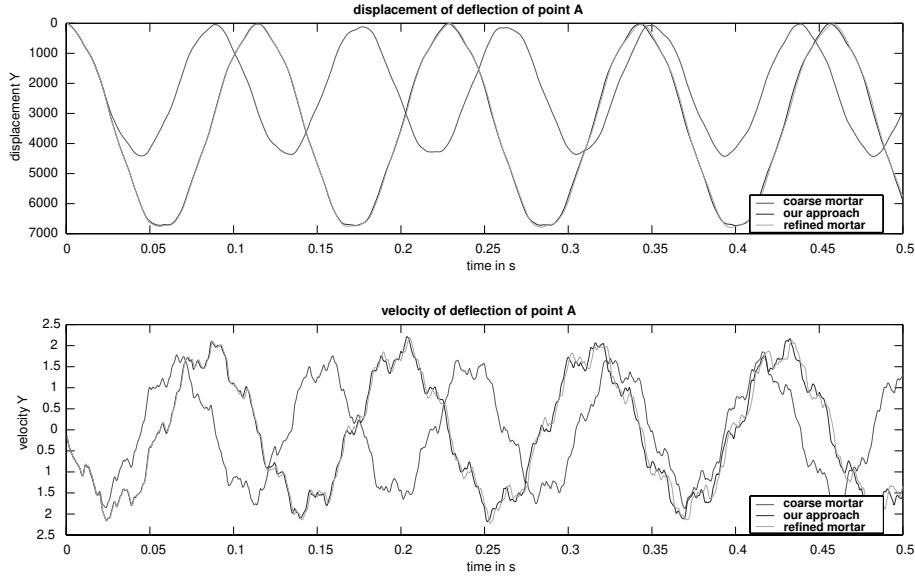


Fig. 11. Non-matching meshes: transient solution.

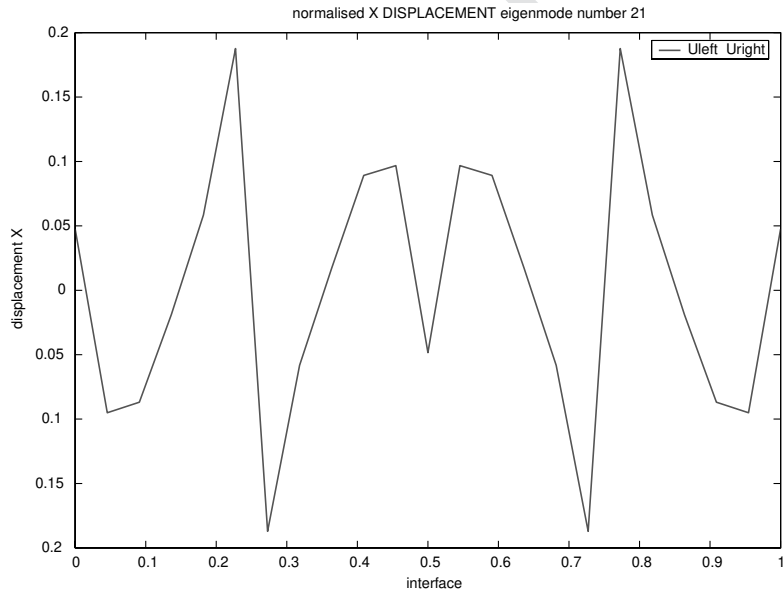


Fig. 12. Non-matching meshes: interface relative difference of displacements.

Fig. 12 represents the displacement difference across the common boundary in case interface vibration mode and of fine mortar method. The displacement difference is normalized by the mean displacement across the thickness. One observes that the amplitude of the normalized difference is 0.2. For the proposed method this difference is strictly null. This differential normal displacement is the cause of the energy dissipation at the interface. The energy dissipated is the sum of a kinetic energy and a strain energy as explained above.

6. Conclusions

This paper has given a general framework to be able to compute a structure decomposed in sub-domains and subjected to a transient loading. The proposed method allows to couple sub-domains having non-matching meshes at the interfaces as well as non-matching time steps. The feasibility has been proven as well as on linear and non-linear problems with different time steps but with matching interface, and for linear problems with

non-matching interfaces. This last part should extend easily to non-linear non-matching sub-domains with different time steps. This verification is now undertaken, but there is no theoretical difficulty in this part. The proposed methodology is very interesting for the practical applications because it gives the maximum possible versatility to the engineers who have no longer to use compatible meshes in the sub-domains. This possibility to use this scheme together with the non-matching time step scheme gives the maximum theoretical efficiency to couple explicit sub-domains with very different mesh size and hence very different time steps.

References

- [1] Bernardi C, Maday Y, Patera AT. A new non-conforming approach to domain decomposition; the mortar method. Non-linear partial differential equations and their applications Collège de France seminar, Pitman, 1990.
- [2] Hughes TJR. In: Belytschko T, Hughes T, editors. Analysis of transient algorithms with particular reference to stability behaviour Computational methods for transient analysis. Amsterdam: North Holland; 1983. p. 67–155.
- [3] Gravouil A, Combescure A. A multi time step explicit–implicit method for non-linear structural dynamics. IJNME 2001;50:199–225.
- [4] Combescure A, Gravouil A. A numerical scheme to couple subdomains with different time-steps for predominantly linear transient analysis. Comp Meth Appl Mech Engng 2002;191:1129–57.
- [5] Gravouil A. Une méthode multi-échelle espace-temps pour le calcul de la dynamique transitoire des structures. PhD thesis defended on October 25th 2000 at ENS Cachan, France.
- [6] Fahrat C, Crivelli L. A transient FETI methodology for large-scale parallel implicit computation in structural dynamics. IJNME 1994;37:1945–75.
- [7] Herry B, Di Valentin L, Combescure A. An approach to the connection between subdomains with non-matching meshes for transient mechanical analysis. IJNME 2002;55: 973–1003.