Available online at www.eccomasproceedia.org
Eccomas Proceedia COMPDYN (2023) 1947-1964

ECCOMAS
Proceedia

COMPDYN 2023
9th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, M. Fragiadakis (eds.)
Athens, Greece, 12-14 June 2023

ASYNCHRONOUSLY IN TIME INTEGRATED INTERFACE DYNAMICS PROBLEM WHILE MAINTAINING ZERO INTERFACE ENERGY

Radim Dvořák[1,2], Radek Kolman[1], Ondřej Jiroušek[2], José A. González[3], K.C. Park[4]

[1]Institute of Thermomechanics of the CAS, v. v. i., Dolejškova 1402/5, 182 00 Praha 8, Czech Republic
e-mail: {radimd,kolman}@it.cas.cz

[2] CTU in Prague Faculty of Transportation Sciences, Department of Mechanics and Materials, Na Florenci 25, 110 00 Praha 1, Czech Republic
e-mail: {dvorara9,jirouond}@fd.cvut.cz

[3] Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Camino de los Descubrimientos s/n, Seville E-41092,Spain
e-mail: japerez@us.es

[4] Department of Aerospace Engineering Sciences, University of Colorado at Boulder CO 80309-429, USA
e-mail: kcpark@colorado.edu

Abstract. *The problem of the linear elastodynamics including domain decomposition via localized Lagrange multipliers method is solved using finite element method and direct time integration. The time integration of the subdomains is performed separately with different time steps with arbitrary ratio. The asynchronous integrator scheme is generalized for multiple subdomain problem with any number of constraints between them. The exact continuity of the displacement, velocity, and acceleration fields at the interface is satisfied. The proposed method is applied to the rectangular step pulse propagation problem considering the linearly varying Young modulus in space as well as the bi-material interface problem. To prove the robustness and the accuracy, the comparison with analytical solution and conventional codes output is provided.*

Keywords: Finite element method, Direct time integration, Domain decomposition, Localized Lagrange multipliers, Asynchronous integrator, Multi time step

ISSN:2623-3347
Peer-review under responsibility of the organizing committee of COMPDYN 2023.
doi: 10.7712/120123.10533.20510

1 Introduction

Transient phenomena are generally challenging to solve numerically since, besides the spatial domain, the time domain also comes into play. The basic problem in space is then solved n-times in n time levels. These are, for example, problems described by the transport equation, wave equation, equation for heat conduction, or unsteady Navier-Stokes equations.

In this contribution, we focus only on the problem of elastodynamics. For spatial discretization, we use the finite element method with the domain decomposition technique. Although theoretically it is possible to choose the Mortar method most commonly used for this purpose, we prefer the method of Localized Lagrange multipliers (LLM) [1, 2]. Lagrange multipliers $\boldsymbol{\ell}_s$ are individually localized to each subdomain s in this case (see Figure 1). Their equilibrium is then satisfied at the subdomain interface, which is described by the displacement $\boldsymbol{u}^{\mathrm{I}}$.

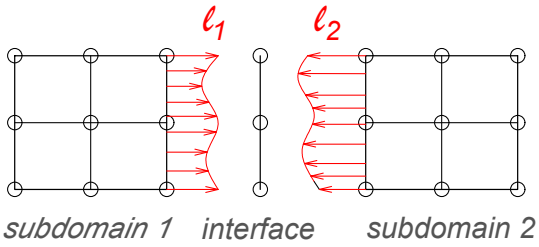


Figure 1: Localization of the Lagrange multipliers to the subdomains

The object of this contribution is to demonstrate how to integrate different subdomains over time with different time steps and, at the same time, preserve the *exact* continuity of the displacement, velocity, and acceleration fields. We prefer to call this process as asynchronous integration [3, 4, 5] - in the literature, one may also find the names as multirate [6], multi-time-step [7] or heterogeneous [5, 8] integration.

The first references to asynchronous integration can be found in the second half of the twentieth century [9, 10, 11]. However, every single method was not robust or accurate enough or even stable. Several counterstudies have been published on these problems [12, 13].

The first robust integrators appeared in the 21st century, in 2002 by Combescure and Gravouil [3] and Cho et al. [8] in 2019. However, both methods still suffer from limitations. The first one [3] is capable of solving the problem of subdomains with arbitrarily different timesteps; however, it tends to dissipate the energy at the interface due to the growth of displacement discontinuity at the interface during the solution (so-called drifting) in specific cases. However, if the ratio of neighboring subdomains steps is integer, the algorithm is energy conserving [5, 7]. The second listed algorithm [8] guarantees the continuity of displacement and is energy-conserving, however, it is able to solve only the case of the integer time-step ratio.

Our proposed method combines the advantages of both methods listed, while eliminating their restrictions and limitations. We propose a method that allows individual subdomains to be time-integrated with arbitrary relative ratios of time steps while maintaining the continuity of the displacement, velocity and acceleration field.

The strong formulation of the problem is defined in Section 2 followed by the weak formulation in Section 3. Spatial discretization by means of finite elements is presented in Section 4. The interface equations are derived in Section 5. In Section 6 temporal discretization of the subdomains and of the interface is introduced. The main contribution represents Section 7, where the proposed method of asynchronous integration is explained on both the single subdomain

pair and the general problem. Finnaly, the asynchronous integrator is validated by comparison with the conventional (non-decomposed model with a single time step) and with the analytical solution of representative benchmarks (Section 8).

2 Strong formulation

The problem of elastodynamics in a linear elastic continuum is governed by the balance of linear momentum, including Hooke's law, complemented by initial and boundary conditions [14]:

$$\begin{aligned} \operatorname{div}\boldsymbol{\sigma} + \boldsymbol{b} &= \rho\ddot{\boldsymbol{u}} \qquad \text{on} \quad \Omega \times \Upsilon\,, \\ \boldsymbol{\sigma} &= \mathbb{C} : \boldsymbol{\varepsilon}\,, \\ \boldsymbol{\varepsilon} &= \frac{1}{2}\left[(\operatorname{grad}\boldsymbol{u})^{\mathsf{T}} + \operatorname{grad}\boldsymbol{u}\right]\,, \end{aligned} \tag{1}$$

$$\begin{aligned} \boldsymbol{u} = \boldsymbol{u}_0(\boldsymbol{x}, t=0) \quad \text{on}\ \Omega\,, \qquad \dot{\boldsymbol{u}}(\boldsymbol{x}, t=0) = \dot{\boldsymbol{u}}_0 \quad \text{on}\ \Omega\,, \\ \boldsymbol{u} = \hat{\boldsymbol{u}} \quad \text{on} \quad \Gamma_{\mathsf{D}}\,, \qquad \boldsymbol{n}\cdot\boldsymbol{\sigma} = \hat{\boldsymbol{t}} \quad \text{on} \quad \Gamma_{\mathsf{N}}\,, \end{aligned}$$

where $\boldsymbol{\sigma}(\boldsymbol{x}, t)$ is the stress tensor, $\boldsymbol{\varepsilon}(\boldsymbol{x}, t)$ is the infinitesimal strain tensor, $\rho(\boldsymbol{x})$ is the mass density, $\boldsymbol{b}(\boldsymbol{x})$ is the vector of the intensity of the volume forces, $\boldsymbol{u}(\boldsymbol{x}, t)$ is the displacement field, $\mathbb{C}(\boldsymbol{x})$ is the elastic 4th order tensor, $\boldsymbol{u}_0$ and $\dot{\boldsymbol{u}}_0$ are the initial displacement and velocity, respectively, $\hat{\boldsymbol{u}}$ is given displacement at the boundary Γ_{D}, $\hat{\boldsymbol{t}}$ is given traction at the boundary Γ_{N}, $\boldsymbol{n}$ is the outward surface normal unit vector, $\boldsymbol{x} \in \Omega$ is the position of the material point, and t is the time. The symbol Ω represents the spatial domain of interest, and Υ refers to the time domain of interest. The dot superscript represents the time derivative. For simplicity, further no Dirichlet boundary conditions are assumed.

Furthermore, we can state that the region Ω consists of n_{s} subdomains Ω_s, $s = 1 \ldots n_{\mathrm{s}}$ (see Figure 2).

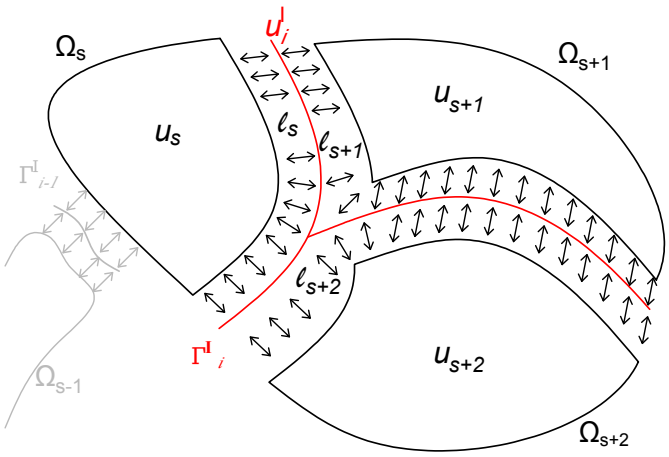


Figure 2: Arbitrary part of the system Ω divided into subdomanins Ω_s connected with i-th interface Γ_i^{I}.

The interface is described by its displacement $\boldsymbol{u}^{\mathrm{I}}(\boldsymbol{x})$. The stress on the interface is $\boldsymbol{\sigma}^{\mathrm{I}}(\boldsymbol{x})$.

The coupling of the subdomains is provided by satisfying

$$\begin{aligned} \boldsymbol{u}_s(\boldsymbol{x}) &= \boldsymbol{u}^{\mathrm{I}}(\boldsymbol{x}) \quad \text{on } \Gamma^{\mathrm{I}} \quad \forall s\,, \\ \boldsymbol{\sigma}_s(\boldsymbol{x}) &= \boldsymbol{\sigma}^{\mathrm{I}}(\boldsymbol{x}) \quad \text{on } \Gamma^{\mathrm{I}} \quad \forall s\,, \end{aligned} \tag{2}$$

what enforces displacement and stress continuity, respectively.

3 Weak formulation

For the weak formulation, we use Hamilton's principle for partitioned constrained elastodynamics. The variation of the Hamiltonian can be written as [15]:

$$\delta\mathcal{H}\left(\boldsymbol{u}, \boldsymbol{u}^{\mathrm{I}}, \boldsymbol{\ell}\right) = \delta\mathcal{H}_{\text{free}}\left(\boldsymbol{u}\right) + \delta\mathcal{W}_{\mathrm{I}}\left(\boldsymbol{u}, \boldsymbol{u}^{\mathrm{I}}, \boldsymbol{\ell}\right)\,, \tag{3}$$

where $\delta\mathcal{H}_{\text{free}}$ consists of the virtual kinetic energy and potential of the system and of the virtual work done by external loads. The virtual work of the interface $\delta\mathcal{W}_{\mathrm{I}}$ can be expressed as

$$\delta\mathcal{W}_{\mathrm{I}}\left(\boldsymbol{u}, \boldsymbol{u}^{\mathrm{I}}, \boldsymbol{\ell}\right) = \delta \sum_{i=1}^{n_{\mathrm{i}}} \sum_{s=1}^{n_{\mathrm{s}}^{i}} \int_{\Gamma^{\mathrm{I}}} \left[\boldsymbol{\ell}_s(\boldsymbol{u}_s - \boldsymbol{u}_i^{\mathrm{I}})\right]\, \mathrm{d}\Gamma\,, \tag{4}$$

where $\boldsymbol{u}_s$ and $\boldsymbol{u}_s^{\mathrm{I}}$ are the displacements of the s-th subdomain and of the i-th interface, respectively (see Figure 2). The $\boldsymbol{\ell}_s$ stands for the Lagrange multiplier field of the s -th subdomain. The whole system consists of n_{i} interfaces and of n_{s} subdomains. The i-th interface is connected to n_{s}^{i} of subdomains. The s-th subdomain is connected with n_{i}^{s} of interfaces.

4 Spatial discretization

For the dicretization, we use conventional finite elements. For the approximation of $\boldsymbol{u}$ and $\boldsymbol{u}^{\mathrm{I}}$ linear shape functions are used. The continuous field $\boldsymbol{\ell}$ is approximated by the discrete field described by the Lagrange multiplier vector $\boldsymbol{\lambda}$. Substituting this approximation into (3) and searching for a stationary solution produces the semi-discrete system.

$$\begin{bmatrix} \bar{\mathbf{K}} & \mathbf{B} & \mathbf{0} \\ \mathbf{B}^{\mathrm{T}} & \mathbf{0} & -\mathbf{L} \\ \mathbf{0} & -\mathbf{L}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \\ \mathbf{u}^{\mathbf{I}} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}\,, \quad \text{where} \quad \bar{\mathbf{K}} = \mathrm{M}\mathrm{D}^2 + \mathbf{K}\,, \quad \mathrm{D} = \frac{\mathrm{d}}{\mathrm{d}t}\,, \tag{5}$$

where the matrices $\mathbf{M}$ and $\mathbf{K}$ in (5) are the subdomain-block-diagonal global mass matrix and the global stiffness matrix, respectively, $\mathbf{f}$ is the vector of external load and can be written with block terms as in the case of matrices

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{M}_{n_{\mathrm{s}}} \end{bmatrix}\,, \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{K}_{n_{\mathrm{s}}} \end{bmatrix}\,, \quad \mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_{n_{\mathrm{s}}} \end{bmatrix}\,. \tag{6}$$

The vector of Lagrange multipliers $\boldsymbol{\lambda}$ similarly consists of n_{s} subvectors $\boldsymbol{\lambda}_s$ and the interface displacement vector $\mathbf{u}^{\mathrm{I}}$ consists of n_{i} subvectors $\mathbf{u}_i^{\mathrm{I}}$.

The matrices $\mathbf{B}^{\mathrm{T}}$ and $\mathbf{L}$ are projectors defined as the scalar product of the appropriate shape functions [16]. The diagonal block matrix $\mathbf{B}^{\mathrm{T}}$ consists of n_{s} matrices $\mathbf{B}_s^{\mathrm{T}}$, the block column vector matrix $\mathbf{L}$ consists of n_{i} matrices $\mathbf{L}_i$. To find the solution of the formal system (5), we have to solve the following equations:

• $\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} - \mathbf{B}\boldsymbol{\lambda}$	(7)	Equation of motion,
• $\mathbf{B}^{\mathrm{T}}\,\ddot{\mathbf{u}} - \mathbf{L}\,\ddot{\mathbf{u}}^{\mathrm{I}} = \mathbf{0}$	(8)	Kinematic interface constraints,
• $\mathbf{L}^{\mathrm{T}}\,\boldsymbol{\lambda} = \mathbf{0}$	(9)	Interface equilibrium condition.

Where the first equation represents the dynamic equilibrium, the second kinematic continuity, and the third traction continuity at the interface. Note that the second-time derrivative of the kinematic continuity is enforced, which in itself does not guarantee the continuity of the displacement and velocity field, and techniques avoiding this phenomenon should be employed.

5 Isolation of the interface problem

We will demonstrate that the unknowns $\boldsymbol{\lambda}$ and $\ddot{\mathbf{u}}^{\mathrm{I}}$ related to the i-th interface can be explicitly expressed from knowledge of the interface solution of connected n_{s}^i subdomains, i.e., from knowledge of $\mathbf{u}$ on Γ^{I}.

By defining the acceleration predictor from the Equation of motion (7)

$$\tilde{\ddot{\mathbf{u}}} = \mathbf{M}^{-1}(\mathbf{f} - \mathbf{K}\mathbf{u}) = \ddot{\mathbf{u}} + \mathbf{M}^{-1}\mathbf{B}\boldsymbol{\lambda}\,, \tag{10}$$

and by its substitution and combining with the the second time-derivative of Kinematic interface constraints we obtain the equation

$$\mathbf{B}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{B}\boldsymbol{\lambda} = \mathbf{B}^{\mathrm{T}}\tilde{\ddot{\mathbf{u}}} - \mathbf{L}\ddot{\mathbf{u}}^{\mathrm{I}}\,, \tag{11}$$

which can be supplemented by the Interface equilibrium condition (9) leading to the system

$$\begin{bmatrix} \mathbf{B}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{B} & \mathbf{L} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \ddot{\mathbf{u}}^{\mathrm{I}} \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{\mathrm{T}}\tilde{\ddot{\mathbf{u}}} \\ \mathbf{0} \end{bmatrix}\,, \tag{12}$$

where the unknows can be explicitly expressed as

$$\begin{aligned} \ddot{\mathbf{u}}^{\mathrm{I}} &= \left[\mathbf{L}^{\mathrm{T}}\left(\mathbf{B}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{B}\right)^{-1}\mathbf{L}\right]^{-1}\left[\mathbf{L}^{\mathrm{T}}\left(\mathbf{B}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{B}\right)^{-1}\mathbf{B}^{\mathrm{T}}\tilde{\ddot{\mathbf{u}}}\right]\,, \\ \boldsymbol{\lambda} &= \left(\mathbf{B}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{B}\right)^{-1}\left(\mathbf{B}^{\mathrm{T}}\tilde{\ddot{\mathbf{u}}} - \mathbf{L}\ddot{\mathbf{u}}^{\mathrm{I}}\right)\,. \end{aligned} \tag{13}$$

From (12), we can extract the problem of a single i-th interface coupled with n_{s}^i subdomains as

$$\begin{bmatrix} {\mathbf{B}_1^i}^{\mathrm{T}}\mathbf{M}_1^{-1}\mathbf{B}_1^i & \mathbf{0} & \dots & \mathbf{0} & \mathbf{L}_i^s \\ \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & {\mathbf{B}_{n_{\mathrm{s}}^i}^i}^{\mathrm{T}}\mathbf{M}_{n_{\mathrm{s}}^i}^{-1}\mathbf{B}_{n_{\mathrm{s}}^i}^i & \mathbf{L}_i^{n_{\mathrm{s}}^i} \\ \mathbf{L}_i^{1\,\mathrm{T}} & \dots & \dots & \mathbf{L}_i^{n_{\mathrm{s}}^i\,\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{L}_1^i\boldsymbol{\lambda}_1 \\ \vdots \\ \vdots \\ \mathbf{L}_i^{n_{\mathrm{s}}^i}\boldsymbol{\lambda}_{n_{\mathrm{s}}^i} \\ \ddot{\mathbf{u}}_i^{\mathrm{I}} \end{bmatrix} = \begin{bmatrix} {\mathbf{B}_1^i}^{\mathrm{T}}\tilde{\ddot{\mathbf{u}}}_1 \\ \vdots \\ \vdots \\ {\mathbf{B}_{n_{\mathrm{s}}^i}^i}^{\mathrm{T}}\tilde{\ddot{\mathbf{u}}}_{n_{\mathrm{s}}^i} \\ \mathbf{0} \end{bmatrix}\,, \tag{14}$$

where matrix ${\mathbf{B}_s^i}^{\mathrm{T}}$ consists only of the columns of the matrix $\mathbf{B}_s^{\mathrm{T}}$ those are responsible for the connection of the s-th subdomain with the i-th interface. Similarly, the matrix $\mathbf{L}_i^s$ consists only of those columns of the matrix $\mathbf{L}_i$ responsible for the connection of the interface i to the s-th

subdomain. Now, we can rewrite the general form (13) of interface equations for the problem of the i-th interface coupled with the n_s^i subdomains.

$$\begin{aligned} \ddot{\mathbf{u}}_i^{\mathrm{I}} &= {\mathbf{M}_i^{\mathrm{I}}}^{-1}\, \mathbf{f}_i^{\mathrm{I}}\,, \\ \boldsymbol{\lambda}_s &= {\mathbf{M}_i^{\boldsymbol{\lambda}_s}}^{-1}\, \mathbf{f}_i^{\boldsymbol{\lambda}_s}\,, \end{aligned} \tag{15}$$

where $s = 1 \ldots n_s^i$ and

$$\begin{aligned} \mathbf{M}_i^{\mathrm{I}} &= \sum_{s=1}^{n_s^i} \left[\mathbf{L}_i^{s\mathrm{T}} \left({\mathbf{B}_s^i}^{\mathrm{T}} \mathbf{M}_s^{-1} \mathbf{B}_s^i \right)^{-1} \mathbf{L}_i^s \right], \qquad \mathbf{f}_i^{\mathrm{I}} = \sum_{s=1}^{n_s^i} \left[\mathbf{L}_i^{s\mathrm{T}} \left({\mathbf{B}_s^i}^{\mathrm{T}} \mathbf{M}_s^{-1} \mathbf{B}_s^i \right)^{-1} {\mathbf{B}_s^i}^{\mathrm{T}} \tilde{\ddot{\mathbf{u}}}_s \right], \\ \mathbf{M}_i^{\boldsymbol{\lambda}_s} &= \left({\mathbf{B}_s^i}^{\mathrm{T}} \mathbf{M}_s^{-1} \mathbf{B}_s^i \right)^{-1}, \qquad \mathbf{f}_i^{\boldsymbol{\lambda}_s} = {\mathbf{B}_s^i}^{\mathrm{T}} \tilde{\ddot{\mathbf{u}}}_s - \mathbf{L}_i^s \ddot{\mathbf{u}}_i^{\mathrm{I}} \end{aligned} \tag{16}$$

are the interface mass matrix, interface load, boundary mass matrix, and boundary load, respectively. Looking at (16) we can recognize that subdomain displacement $\mathbf{u}_s$ is always transformed using the operator ${\mathbf{B}_s^i}^{\mathrm{T}}$. The operation

$${\mathbf{B}_s^i}^{\mathrm{T}}\, \mathbf{u}_s = {\mathbf{u}^{\mathbf{B}^i}}_s \tag{17}$$

can be understood as selecting only those degrees of freedom of the subdomain s, which are located at the interface i, that is, we obtain the vector of boundary displacement ${\mathbf{u}^{\mathbf{B}^i}}_s$. Thus, we see that it is sufficient to know the solution of the subdomains only at their boundary related to the interface i to obtain the primary unknowns of the i-th interface $\boldsymbol{\lambda}$ and $\ddot{\mathbf{u}}^{\mathrm{I}}$, as we claimed at the beginning of this chapter.

6 Temporal discretization

For time integration, various methods can be used such as the Newmark method [17], Central Difference method [14], or Pushforward-pullback method [18]. For the sake of clarity, we will demonstrate the Central Difference method in its fundamental form [14], that is, the substructure displacement and velocity in time t^{n+1} are obtained as

$$\begin{aligned} \mathbf{u}^{n+1} &= \mathbf{u}^n + \Delta t\, \dot{\mathbf{u}}^n + \frac{\Delta t^2}{2} \ddot{\mathbf{u}}^n\,, \\ \dot{\mathbf{u}}^{n+1} &= \dot{\mathbf{u}}^n + \frac{\Delta t}{2} \left(\ddot{\mathbf{u}}^n + \ddot{\mathbf{u}}^{n+1} \right). \end{aligned} \tag{18}$$

To obtain the acceleration $\ddot{\mathbf{u}}^{n+1}$ one has to solve the system of equations (10) and (12) in time t^{n+1}.

However, one must proceed with caution, since the second time derivative of Kinematic interface constraints (8) in the system (12) is used and therefore phenomena called drifting will occur at the interfaces, i.e., the discontinuity of the displacement and velocity field will develop at the interface during time integration. For that reason, Cho et al. [8] introduced the procedure of *avoiding drifting* for the mesh matching case:

1. After obtaining a subdomain solution at the time level t^{n+1}, evaluate the velocity and displacement of the interface from the known acceleration $(\ddot{\mathbf{u}}^{\mathrm{I}})^{n+1}$ of the interface.

$$(\dot{\mathbf{u}}^{\mathrm{I}})^{n+1} = (\dot{\mathbf{u}}^{\mathrm{I}})^n + \frac{\Delta t}{2} \left((\ddot{\mathbf{u}}^{\mathrm{I}})^n + (\ddot{\mathbf{u}}^{\mathrm{I}})^{n+1} \right), \qquad (\mathbf{u}^{\mathrm{I}})^{n+1} = (\mathbf{u}^{\mathrm{I}})^n + \Delta t\, (\dot{\mathbf{u}}^{\mathrm{I}})^{n+1}\,, \tag{19}$$

2. Rewrite the degrees of freedom of the solution $\mathbf{u}^{n+1}$ and $\dot{\mathbf{u}}^{n+1}$ on Γ^{I} so the equations

$$\mathbf{B}^{\mathrm{T}}\left(\mathbf{u}\right)^{n+1} - \mathbf{L}\left(\mathbf{u}^{\mathrm{I}}\right)^{n+1} = \mathbf{0}\,, \quad \mathbf{B}^{\mathrm{T}}\left(\dot{\mathbf{u}}\right)^{n+1} - \mathbf{L}\left(\dot{\mathbf{u}}^{\mathrm{I}}\right)^{n+1} = \mathbf{0} \tag{20}$$

   are satisfied.

Although Step 2 is trivial in the case of matching meshes, in the nonmatching case one has to face the problem of projection and be careful about consistency with variational principle and stability of the scheme.

7 The asynchronous time integration

The crucial difference from the conventional approach is that each subdomain has its own time step. The challenge of managing precise and versatile asynchronous integration consists of the following.

- using arbitrary and time-varying time step length for each subdomain (i.e., possibly non-integer time step ratio of each subdomain pair),
- respecting both kinematic and traction coupling equations at each time level t_s^n of any subdomain s.

A remarkable outcome follows from the second point, with the consequence of theoretically zero-energy dissipation at the interfaces.

7.1 Definition of *interface regions*

The crucial point of asynchronous integration lies in the complete understanding of the *finite wave speed* propagation and *the critical time step* term.

Assume that the free square 2D domain Ω consisting of equal square finite elements with constant material properties is loaded by constant stress on its right edge. We want to find a solution in time $2\Delta t_C$, where Δt_C is the critical time step of the central difference method of a given system assuming the diagonal mass matrix. Then the solution obtained by the time integration is the same, whenever it is obtained by time integration

- of the full system or
- of *interface region* $\Omega_{\mathbf{R}}$ and of the *domain complement* $\Omega \setminus \Omega_{\mathbf{R}}$ separately followed by the union of both solutions (see Fig. 3).

In this specific case, *interface region* consists of two layers of the elements.

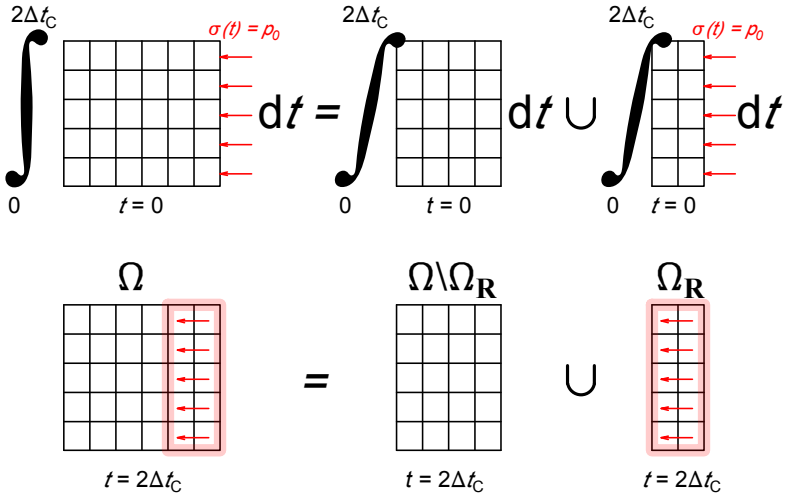


Figure 3: Equality of the time integration of the whole set Ω and divided problem

The *interface region* definition is crucial ingredient in asynchronous integration. Generally, *interface region* is obtained from the original model Ω by the transformation (which is a de facto reduction of degrees of freedom corresponding to the domain complement $\Omega \setminus \Omega_{\mathbf{R}}$) given by the block diagonal boolean operator (transformation matrix) $\mathbf{R}$ consisted of block diagonal operators $\mathbf{R}_s$ for each subdomain $s = 1 \dots n_s$ consisted of operators $\mathbf{R}_s^i$, which selects the interface region of the s-th subdomain connected to the interface i.

By *interface region* $\Omega_{\mathbf{R}_s^i}$ we mean the area of the subdomain s affected by the propagating wave from the interface i during at least two critical steps of this area $\Delta t_{C\mathbf{R}_s^i}$.

This allows us to solve the problem of the i-th interface (15) at the time level $t + \Delta t$ (recall $\Delta t < \Delta t_C$) including only regions $\Omega_{\mathbf{R}_s^i}$ $s = 1 \dots n_s^i$ instead of entire subdomains Ω_s (see Figure 4).

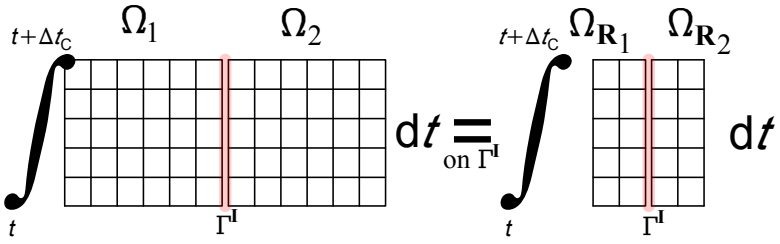


Figure 4: The equality of the i-th interface solution assuming whole subdomains Ω_s vs. *interface regions* $\Omega_{\mathbf{R}_s}$ only

We can compactly formulate the interface region subproblems in block diagonal form by the substitution $\mathbf{u} = \mathbf{R}^{\mathrm{T}}\,\mathbf{u}^{\mathbf{r}}$ to (5) as

$$\begin{bmatrix} \bar{\mathbf{K}}^{\mathbf{r}} & \mathbf{B}^{\mathbf{r}} & \mathbf{0} \\ \mathbf{B}^{\mathbf{r}\mathrm{T}} & \mathbf{0} & -\mathbf{L} \\ \mathbf{0} & -\mathbf{L}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{\mathbf{r}} \\ \boldsymbol{\lambda} \\ \mathbf{u}^{\mathbf{I}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{\mathbf{r}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad \text{where} \quad \bar{\mathbf{K}}^{\mathbf{r}} = \mathbf{M}^{\mathbf{r}}\mathrm{D}^2 + \mathbf{K}^{\mathbf{r}}, \quad \mathrm{D} = \frac{\mathrm{d}}{\mathrm{d}t}, \tag{21}$$

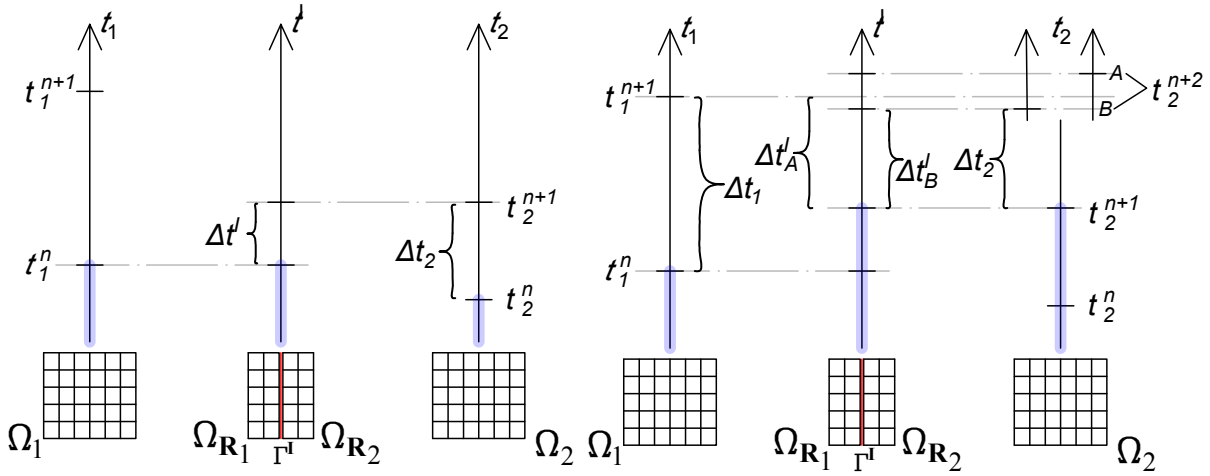
where

$$\mathbf{K}^{\mathbf{r}} = \mathbf{R}\mathbf{K}\mathbf{R}^{\mathrm{T}}, \quad \mathbf{M}^{\mathbf{r}} = \mathbf{R}\mathbf{M}\mathbf{R}^{\mathrm{T}}, \quad \mathbf{f}^{\mathbf{r}} = \mathbf{R}\mathbf{f}, \quad \mathbf{B}^{\mathbf{r}} = \mathbf{R}\mathbf{B} \tag{22}$$

are the *interface region* matrices and vectors.

7.2 Fundamental problem of two subdomains connected with one interface

We will present one computational step of the proposed asynchronous integration of a single subdomain pair. Assume that the subdomains Ω_1 and Ω_2 are connected via the interface Γ^{I} (see Figure 5a). The blue highlight has the meaning of an already known solution, that is, the last known solution of Ω_1 is at the time level t_1^n, of Ω_2 is at t_2^n and we assume that we know the solution of the interface ($\boldsymbol{\lambda}$, $\mathbf{u}^{\mathbf{I}}$, $\dot{\mathbf{u}}^{\mathbf{I}}$, $\ddot{\mathbf{u}}^{\mathbf{I}}$) at the time level t_1^n. Our challenge is to obtain the solution of Ω_1 at the time level t_1^{n+1} and Ω_2 at t_2^{n+1}.



(a) Initial state of the algorithm (b) Deciding point of the algorithm (scenario A and B)

Figure 5: Two consequent states of the system during the asynchronous time integration

We propose the following algorithm:

1. Initialize the interface time step as $\Delta t^{\mathrm{I}} = t_2^{n+1} - t_1^n$.
2. Solve the interface in time t_2^{n+1} ($t_2^{n+1}\ldots^{n+1}$, $t_1^n\ldots^n$ simplified notation)
   (a) $(\mathbf{u}^{\mathbf{r}})^{n+1} = (\mathbf{u}^{\mathbf{r}})^n + \Delta t^I\,(\dot{\mathbf{u}}^{\mathbf{r}})^n + \frac{\Delta t^{I\,2}}{2}(\ddot{\mathbf{u}}^{\mathbf{r}})^n$
   (b) $(\tilde{\ddot{\mathbf{u}}}^{\mathbf{r}})^{n+1} = \mathbf{M}^{\mathbf{r}-1}\left((\mathbf{f}^{\mathbf{r}})^{n+1} - \mathbf{K}^{\mathbf{r}}\,(\mathbf{u}^{\mathbf{r}})^{n+1}\right)$
   (c) $(\mathbf{f}^{\mathrm{I}})^{n+1} = \sum_{s=1}^{2}\left[\mathbf{L}^{s\mathrm{T}}\left(\mathbf{B}_s^{\mathbf{r}\mathrm{T}}\mathbf{M}_s^{-1}\mathbf{B}_s^{\mathbf{r}}\right)^{-1}\mathbf{B}_s^{\mathbf{r}\mathrm{T}}(\tilde{\ddot{\mathbf{u}}}_s^{\mathbf{r}})^{n+1}\right]$
   (d) $(\ddot{\mathbf{u}}^{\mathrm{I}})^{n+1} = \mathbf{M}^{\mathrm{I}-1}\,(\mathbf{f}^{\mathrm{I}})^{n+1}$
   (e) $(\dot{\mathbf{u}}^{\mathrm{I}})^{n+1} = (\dot{\mathbf{u}}^{\mathrm{I}})^n + \frac{\Delta t^I}{2}\left(\left(\ddot{\mathbf{u}}^{\mathrm{I}}\right)^n + \left(\ddot{\mathbf{u}}^{\mathrm{I}}\right)^{n+1}\right)$
   (f) $(\mathbf{u}^{\mathrm{I}})^{n+1} = (\mathbf{u}^{\mathrm{I}})^n + \Delta t^I\,(\dot{\mathbf{u}}^{\mathrm{I}})^{n+1}$
3. Complete the solution of $\Omega_{\mathbf{R}_1}$ in time t_2^{n+1} ($t_2^{n+1}\ldots^{n+1}$, $t_1^n\ldots^n$)
   (a) $(\mathbf{f}^{\boldsymbol{\lambda}_1})^{n+1} = \left(\mathbf{B}^{\mathbf{r}\mathrm{T}}_1(\tilde{\ddot{\mathbf{u}}}_1^{\mathbf{r}})^{n+1} - \mathbf{L}^1(\ddot{\mathbf{u}}^{\mathrm{I}})^{n+1}\right)$
   (b) $(\boldsymbol{\lambda}_1)^{n+1} = \mathbf{M}_1^{\boldsymbol{\lambda}_1-1}(\mathbf{f}^{\boldsymbol{\lambda}_1})^{n+1}$
   (c) $(\ddot{\mathbf{u}}_1^{\mathbf{r}})^{n+1} = (\tilde{\ddot{\mathbf{u}}}_1^{\mathbf{r}})^{n+1} - \mathbf{M}_1^{\mathbf{r}-1}\mathbf{B}_1^{\mathbf{r}}(\boldsymbol{\lambda}_1)^{n+1}$
   (d) $(\dot{\mathbf{u}}_1^{\mathbf{r}})^{n+1} = (\dot{\mathbf{u}}_1^{\mathbf{r}})^n + \frac{\Delta t^I}{2}\,((\ddot{\mathbf{u}}_1^{\mathbf{r}})^n + (\ddot{\mathbf{u}}_1^{\mathbf{r}})^{n+1})$

(e) Avoid drifting of $\{\mathbf{u}^{\mathbf{r}}, \dot{\mathbf{u}}^{\mathbf{r}}\}_1^{n+1}$

4. Solve free subdomain Ω_2 in time t_2^{n+1} $\quad (t_2^{n+1} \ldots^{n+1}, t_2^n \ldots^n)$

(a) $\mathbf{u}_2^{n+1} = \mathbf{u}_2^n + \Delta t_2\, \dot{\mathbf{u}}_2^n + \frac{\Delta(t_2)^2}{2} \ddot{\mathbf{u}}_2^n$

(b) $\tilde{\ddot{\mathbf{u}}}_2^{n+1} = \mathbf{M}^{-1}(\mathbf{f}_2^{n+1} - \mathbf{K}_2\, \mathbf{u}_2^{n+1})$

(c) Rewrite (overwrite) appropriate degrees of freedom $\tilde{\ddot{\mathbf{u}}}_2^{n+1} \rightarrow \ddot{\mathbf{u}}_2^{n+1}$ with respect to $\mathbf{B}_2^{\mathrm{T}}\, \ddot{\mathbf{u}}_2^{n+1} - \mathbf{L}^2\, (\ddot{\mathbf{u}}^{\mathrm{I}})^{n+1} = \mathbf{0}$

(d) $\dot{\mathbf{u}}_2^{n+1} = \dot{\mathbf{u}}_2^n + \frac{\Delta t_2}{2} \left(\ddot{\mathbf{u}}_2^n + \ddot{\mathbf{u}}_2^{n+1}\right)$

(e) Avoid drifting of $\{\mathbf{u}, \dot{\mathbf{u}}\}_2^{n+1}$

(f) Reset interface region $\{\mathbf{u}^{\mathbf{r}}, \dot{\mathbf{u}}^{\mathbf{r}}, \ddot{\mathbf{u}}^{\mathbf{r}}\}_2^{n+1} = \mathbf{R}_2\{\mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}}\}_2^{n+1}$

At this point, two different situations *A* and *B* may occur (see Fig. 5b):

Scenario A: $t_2^{n+2} > t_1^{n+1}$ In this case, the process is in the initial state but with the switched role of subdomains Ω_1 and Ω_2. Initialize the interface time step as $\Delta t^{\mathrm{I}} = \Delta t_A^{\mathrm{I}} = t_1^{n+1} - t_2^{n+1}$ and repeat the algorithm from 2nd step with switched indices $s = 1 \rightarrow 2,\ 2 \rightarrow 1$.

Scenario B: $t_2^{n+2} < t_1^{n+1}$ Initialize the interface time step as $\Delta t^{\mathrm{I}} = \Delta t_B^{\mathrm{I}} = t_2^{n+2} - t_2^{n+1}$ and repeat the algorithm from 2nd step.

The computation ends once each of the subdomains has reached the final time t_{end}.

7.3 General problem of multiple arbitrarily connected subdomains

Thanks to the definition of interfaces, interface regions and appropriate notation in the sections above, the asynchronous integration of a general problem is only matter of the patient implementation. In our opinion, there is no necessity of presenting the algorithm for a general problem, instead, we give the reader several remarks and recommendations that may help with the implementation process.

Generally, we would like to solve the problem of subdomains $\Omega_s \in \mathbf{S}$ $\quad (s = 1 \ldots n_{\mathrm{s}})$ and interfaces $\Gamma_i^{\mathrm{I}} \in \mathbf{I}$ $\quad (i = 1 \ldots n_{\mathrm{i}})$, which are specifically interconnected - each interface Γ_i^{I} is connected to the set of subdomains $\mathbf{S}_i$ and each subdomain Ω_s is connected to the set of interfaces $\mathbf{I}_s$. The sets $\mathbf{S}$, $\mathbf{I}$, $\mathbf{S}_i$ and $\mathbf{I}_s$ are given and remain unchanged during the solution naturally. Besides the existence of these sets, we will further define the *solvable sets* of subdomains and interfaces $\hat{\mathbf{S}}$ and $\hat{\mathbf{I}}$ that change during the simulation according to the actual *state* - the *state* of the problem is given by the value of the *clocks* parameter:

Clocks Keep two time parameters for each subdomain s

- t_s^{act} has the value of the time of the last known solution of the subdomain,
- t_s^{new} has the value of the time of the solution we are looking for.

The time of interest t_s^{new} is given by the computational time step of s-th subdomain as $t_s^{\mathrm{new}} = t_s^{\mathrm{act}} + \Delta t_s$. The computational time step can be chosen arbitrarily, but must be less than the critical one $\Delta t_s < \Delta t_s^C$.

Also, keep the time value of the last known solution of each i-th interface in parameter t_i^I. Together with the two previous parameters, we will further determine the interface time step Δt_i^I.

Solvable set of interfaces $\hat{\mathbf{I}}$

$$\text{definition:} \quad \Gamma_i^{\mathrm{I}} \in \hat{\mathbf{I}} \iff t_s^{\text{new}} > t_i^I \quad \forall\, \Omega_s \in \mathbf{S}_i$$

The interface $\Gamma_i^{\mathrm{I}} \in \hat{\mathbf{I}}$ is always solvable and can be integrated with the interface time step given as $\Delta t_i^I = \min\{t_s^{\text{new}} - t_i^I\} \quad \forall\, \Omega_s \in \mathbf{S}_i$.

Solvable set of subdomains $\hat{\mathbf{S}}$

$$\text{definition:} \quad \Omega_s \in \hat{\mathbf{S}} \iff t_s^{\text{new}} = t_i^I \quad \forall\, \Gamma_i^{\mathrm{I}} \in \mathbf{I}_s$$

The subdomain $\Omega_s \in \hat{\mathbf{S}}$ is always solvable and can be integrated with its computational timestep Δt_s. Note that $t_s^{\text{act}} + \Delta t_s = t_s^{\text{new}} = t_i^I \quad \forall\, \Gamma_i^{\mathrm{I}} \in \mathbf{I}_s$.

Solution process The asynchronous solution may proceed, because there always exists an unempty set of solvable subdomains $\hat{\mathbf{S}}$ or interfaces $\hat{\mathbf{I}}$ at any point of the computation process. In practice, the user only needs to implement a function that can determine these two sets from the defined time parameters and from coupling relationships among all subdomains and interfaces of the system.

8 Numerical validation

Wave propagation through the bimaterial interface and through linearly graded material is investigated (1D formulation). In the case of a bimaterial interface, special attention is given to the energy balance. Furthermore, the impact of 3 thin rods is simulated (2D axisymmetric formulation) and the results are compared with bi-penalty contact specialized verified 3D solver [15] output.

By term *conventional* in the figure's legend below, we mean the computation with a nondecomposed model with a single globally determined time step. By term *asynchronous* we mean the computation with the proposed method.

8.1 Bi-material interface (1D)

It is considered to be a bimaterial bar loaded with a rectangular pulse of 1 Pa (see Figure 6). Each material region is discretized with elements of different sizes ($\Delta h_1 = 0.008$ m, $\Delta h_2 = 0.01$ m) to provide a non-integer critical time step ratio. Each material region is represented by one subdomain, the interface of the subdomains is located exactly at the bimaterial interface.

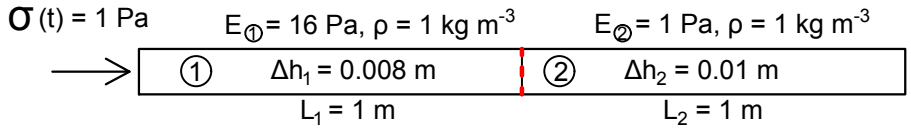


Figure 6: Bar setup with bimaterial interface

The stress at time $t = 0.45$ s is plotted in Figure 7. We can clearly see the match of the proposed method (green) with the conventional (red). Spurious oscillations around the analytical solution are caused by the use of a non-critical computational time step $\Delta t_s = 0.5\Delta t_s^C$.

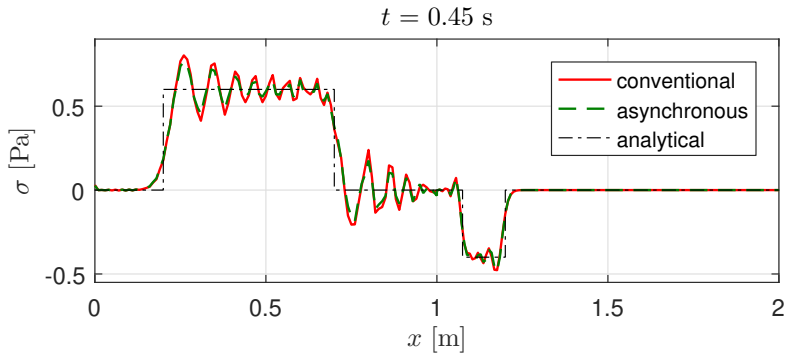
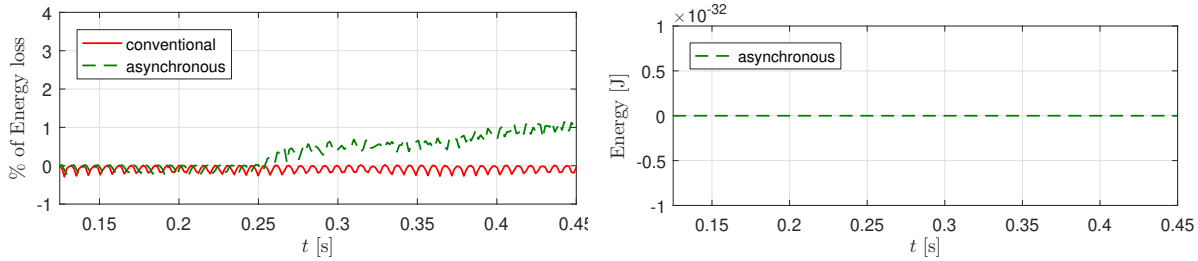


Figure 7: Reflected and transmitted stress pulse at time $t = 0.45$ s

The percentage loss of the total energy (compared to the work done by the load) is shown in the Figure 8a. The interface energy (4) is exactly zero during whole computation.



(a) Total energy loss

(b) Interface energy

Figure 8: Total energy loss (compared to real work done by the load) evolution (left) and interface energy evolution of the asynchronous case (right).

The error of the method is present in the subdomains themselves, where we can see the distortion of the potential and kinetic energy components (see Figure 9). The percentage difference from the conventional solution is plotted.

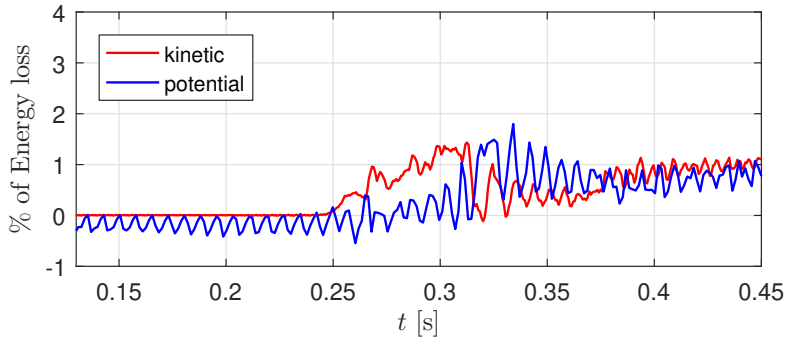


Figure 9: Potential and kinetic energy loss of asynchronously integrated decomposed model compared to the energy conserving single time step scheme

8.2 Graded material (1D)

It is given a bar with a linearly changing Young modulus loaded by a rectangular pulse of 1 Pa (see Figure 10). The bar is divided into 5 equaly sized subdomains. Elements size is set to $\Delta h = 0.01$ m.

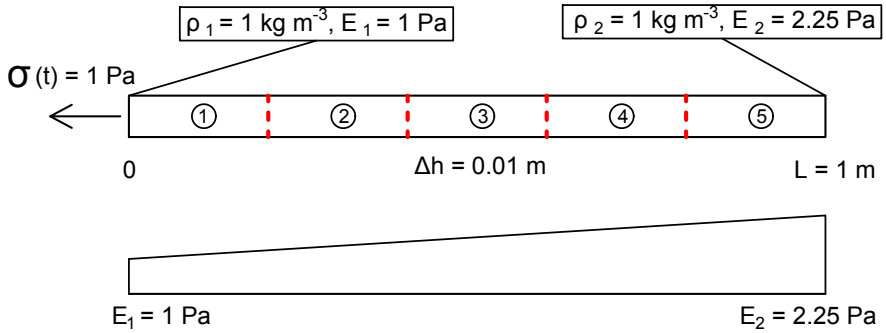


Figure 10: Bar setup with linearly varying Young modulus

The stress at time $t = 0.75$ s is plotted in Figure 11. We can clearly see the match of the proposed method (green) with the conventional (red). We can also observe the spurious oscillation as in the previous bimaterial test.

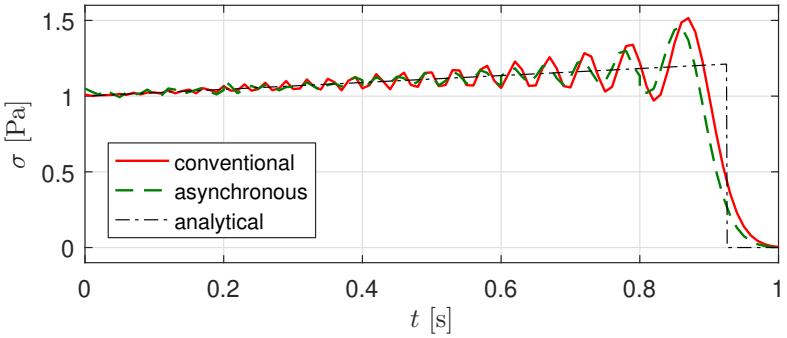


Figure 11: Propagated stress pulse at time $t = 0.75$ s

8.3 Impact of three bars (2D axisymmetry)

The setup is made up of 3 coincident bars of different materials (see Table 1). The first bar made of polymethylmethacrylate (PMMA) has prescribed the initial velocity, which simulates the impact. The second and the third bar are made of steel and EN-AW-7075-T6 aluminum alloy, respectively.

Bar no.	Impact velocity [m/s]	Length [mm]	Diameter [mm]	Material	Density ρ [kg/m^3]	Young modulus E [GPa]	Poisson ratio ν [-]
1.	15.2	1750	20	$PMMA$	1180	5	0.37
2.	0	796	20	$Steel$	7850	210	0.3
3.	0	1600	20	Al	2806	72	0.334

Table 1: Initial velocities and parameters of the bars used in the experimental setup

In case of the asynchronous integrator, each bar refers to one subdomain, that is, 3 subdomains and 2 interfaces in total are defined. The length of the edge of the 2D axisymmetric square element is set to the value of $\sim$ 1.6 mm. The computational time step is set to the half of the critical one.

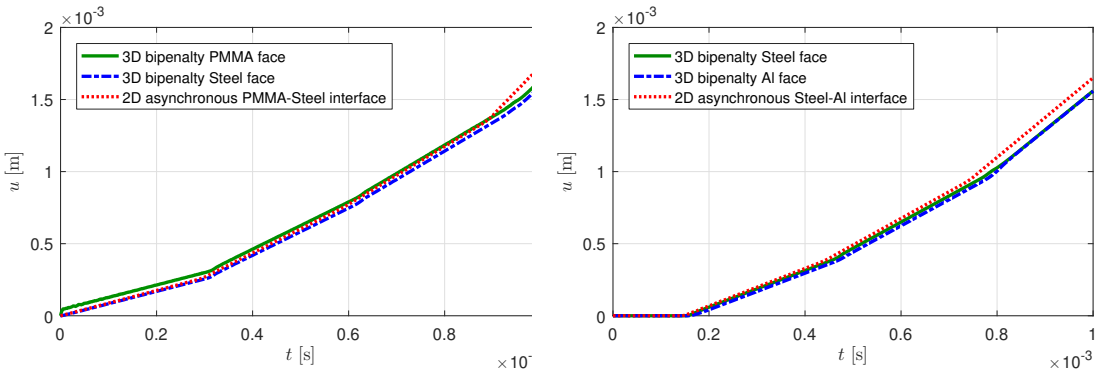
8.3.1 Limitations of the comparison

The numerical results obtained by the 3D solver with the bi-penalty contact method have been used for validation. Details about the bi-penalty method and the solver itself can be found in [15, 19, 20, 21].

The computation is performed only in the time window during which all of the bars are in contact. This eliminates the difference between the contact mechanism of the bipenalty method and the *longitudinally-glued* interfaces of our domain decomposed model. To bring the asynchronous 2D axisymmetric simulation to the 3D contact problem as close as possible, only the longitudinal (perpendicular to the interface) degrees of freedom are permanently coupled (*longitudinally-glued*) at the interfaces. Transverse degrees of freedom are permanently decoupled, i.e., transverse free-end breathing is allowed.

8.3.2 Displacement of the interfaces and strain ε_{11} response

The displacements of the bar fronts for the first PMMA-Steel and the second Steel-Al interfaces are plotted (see Figure 12). The 3D FEA results are described by two curves per each interface to prove that the contact areas remained in contact during the whole computation.



(a) PMMA-Steel interface

(b) Steel-Al interface

Figure 12: Histories of axial displacement u_1 at bar interfaces for time interval $t \in \langle 0, 1 \rangle$ ms

The history of axial strain ε_{11} measured at distance $x = 2.746$ m, at the location of the 2nd interface, is plotted in the Figure 13 . The strain time evolution is acceptably comparable for each method. The lack of an artificial strain oscillation in the 3D FE contact analysis (solid blue curve) is caused by the filter-like behavior of the bipenalty contact formulation [15, 21].

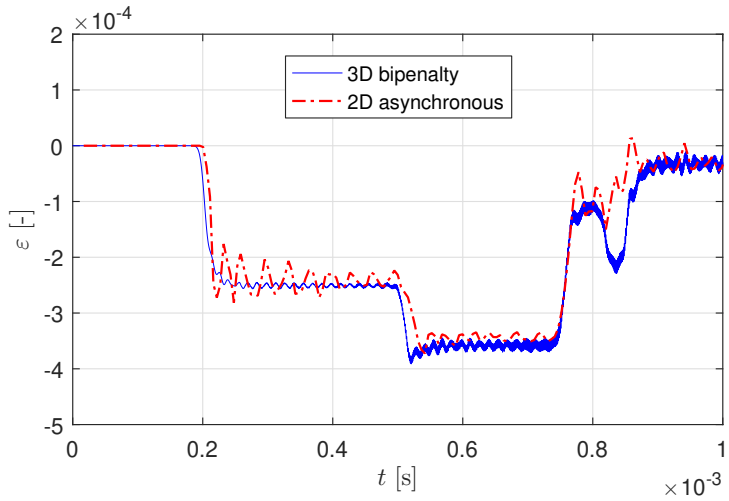
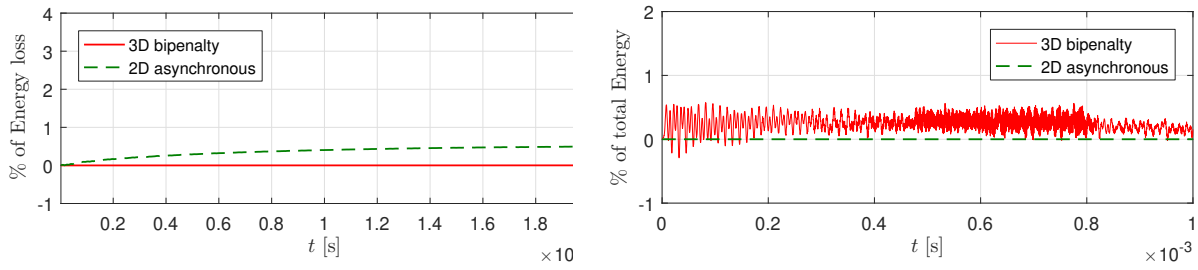


Figure 13: Strain ε_{11} response at the location of the 2nd interface: 3D FEA with contact and proposed asynchronous scheme on 2D axisymmetric model

8.3.3 Energy balance and dissipation

Figure 14a shows the percentage loss of total energy given by the initial velocity of the PMMA bar. For the asynchronous case, the energy dissipation rate is noticeable during the first tenths of milliseconds of the simulation, when the step wavefront propagates through the interfaces. The energy dissipation rate is continually decreasing. The interface energy (4) is zero during the whole simulation (see Figure 14b). Although during the 3D FEA contact analysis no energy is dissipated in total (see Figure 14a), the distortion of the solution is present because of the contact stiffness - see the history of the accumulated contact energy, which should theoretically be zero (see Figure 14b).



(a) Total energy loss

(b) Interface energy

Figure 14: Total energy loss (compared to initial kinetic energy of the PMMA bar) evolution (left) and interface energy evolution (right).

Conclusions

The proposed asynchronous direct time integration scheme is a robust tool for computing decomposed models with different time steps. Although only the case of elastic wave propagation has been demonstrated, the scheme can be applied to non-linear physics problems such as fluid structure interaction [22]. The scheme preserves the continuity of the displacement, velocity, and acceleration field at the interface, and hence its zero energy. The ratios of time steps of

adjacent subdomains can be arbitrary; no restrictions are imposed. The weakness of the method lies in the negligible distortion of the potential and kinetic energy in the interface surroundings and only in the case of discontinuity propagation (rectangular load wave). As a result, we find the scheme to be very suitable for dealing with the propagation of discontinuities such as the investigated step stress pulses.

Acknowledgment

The work of R. D. and R.K. was supported by grant projects with No. 22-00863K of the Czech Science Foundation (CSF) within institutional support RVO:61388998. The work of R. D. was also supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS22/196/OHK2/3T/16. R.K. and R.R. also realized the work under the Czech-Estonian mobility project EstAV-21-02.

REFERENCES

[1] K. C. Park, C. Felippa, and U. Gumaste, "A localized version of the method of lagrange multipliers and its applications," *Computational Mechanics*, vol. 24, pp. 476–490, 01 2000.

[2] K. C. Park, C. A. Felippa, and G. Rebel, "A simple algorithm for localized construction of non-matching structural interfaces," *International Journal for Numerical Methods in Engineering*, vol. 53, no. 9, pp. 2117–2142, Mar. 2002. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/nme.374/abstract

[3] A. Combescure and A. Gravouil, "A numerical scheme to couple subdomains with different time-steps for predominantly linear transient analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, pp. 1129–1157, 01 2002.

[4] W. Subber and K. Matouš, "Asynchronous space–time algorithm based on a domain decomposition method for structural dynamics problems on non-matching meshes," *Computational Mechanics*, vol. 57, no. 2, pp. 211–235, 2016. [Online]. Available: https://link.springer.com/article/10.1007/s00466-015-1228-0#citeas

[5] F.-E. Fekak, M. Brun, A. Gravouil, and B. Depale, "A new heterogeneous asynchronous explicit–implicit time integrator for nonsmooth dynamics," *Computational Mechanics*, vol. 60, no. 1, pp. 1–21, Jul 2017. [Online]. Available: https://doi.org/10.1007/s00466-017-1397-0

[6] B. Seny, J. Lambrechts, V. Legat, and J.-F. Remacle, "Efficient parallel multirate time stepping for accelerating explicit discontinuous galerkin computations," *Fifth International Conference on Advanced Computational Methods in ENgineering*, 08 2011.

[7] A. Prakash and K. D. Hjelmstad, "A feti-based multi-time-step coupling method for newmark schemes in structural dynamics," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 13, pp. 2183–2204, 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1136

[8] S. S. Cho, R. Kolman, J. A. González, and K. C. Park, "Explicit multistep time integration for discontinuous elastic stress wave propagation in heterogeneous solids," *International*

Journal for Numerical Methods in Engineering, vol. 118, no. 5, pp. 276–302, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6027

[9] T. Belytschko and R. Mullen, *Mesh partitions of explicit-implicit time integration, Formulations and computational algorithms in finite element analysis : U.S.-Germany symposium*. Cambridge: Massachusetts Institute of Technology, 1977.

[10] T. Belytschko and M. Robert, "Stability of explicit-implicit mesh partitions in time integration," *International Journal for Numerical Methods in Engineering*, vol. 12, no. 10, pp. 1575–1586, 1978. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620121008

[11] P. Smolinski, "Subcycling integration with non-integer time steps for structural dynamics problems," *Computers & Structures*, vol. 59, pp. 273–281, 04 1996.

[12] W. J. T. Daniel, "A study of the stability of subcycling algorithms in structural dynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 156, no. 1-4, pp. 1–13, 1998.

[13] P. Smolinski, S. Sleith, and T. Belytschko, "Stability of an explicit multi-time step integration algorithm for linear structural dynamics equations," *Computational Mechanics*, vol. 18, pp. 236–244, 07 1996.

[14] T. J. R. Hughes, *The finite element method: linear static and dynamic finite element analysis*. Mineola, NY: Dover Publications, 2000.

[15] J. A. González, J. Kopačka, R. Kolman, and K.-C. Park, "Partitioned formulation of contact-impact problems with stabilized contact constraints and reciprocal mass matrices," *International Journal for Numerical Methods in Engineering*, vol. 122, no. 17, pp. 4609–4636, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6739

[16] K. C. Park and C. A. Felippa, "A Variational Framework for Solution Method Developments in Structural Mechanics," *J. Appl. Mech*, vol. 65, no. 1, pp. 242–249, Mar. 1998. [Online]. Available: http://dx.doi.org/10.1115/1.2789032

[17] N. M. Newmark, "A method of computation for structural dynamics," *Journal of the Engineering Mechanics Division*, vol. 85, no. 3, pp. 67–94, 1959. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/JMCEA3.0000098

[18] S. S. Cho, K. C. Park, and H. Huh, "A method for multidimensional wave propagation analysis via component-wise partition of longitudinal and shear waves," *International Journal for Numerical Methods in Engineering*, vol. 95, no. 3, pp. 212–237, 2013. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4495

[19] R. Kolman, J. Kopačka, J. A. González, S. Cho, and K. Park, "Bi-penalty stabilized technique with predictor–corrector time scheme for contact-impact problems of elastic bars," vol. 189, pp. 305–324. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378475421000987

[20] J. Kopačka, D. Gabriel, J. Plešek, and M. Ulbin, "Assessment of methods for computing the closest point projection, penetration, and gap functions in contact searching problems," *International Journal for Numerical Methods in Engineering*, vol. 105, no. 11, pp. 803–833, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4994

[21] J. Kopačka, A. Tkachuk, D. Gabriel, R. Kolman, M. Bischoff, and J. Plešek, "On stability and reflection-transmission analysis of the bipenalty method in contact-impact problems: A one-dimensional, homogeneous case study," vol. 113, no. 10, pp. 1607–1629. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nme.5712

[22] J. A. González, K. Park, I. Lee, C. Felippa, and R. Ohayon, "Partitioned vibration analysis of internal fluid-structure interaction problems," *International Journal for Numerical Methods in Engineering*, vol. 92, no. 3, pp. 268–300, 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4336