**ACTA IMEKO**
**ISSN: 2221-870X**
*September 2026, Volume 15, Number 3, 1 – 8*

# Performance analysis of an asynchronous time integration scheme for Split Hopkinson Pressure Bar simulations

**Radim Dvořák[1,2], Lukáš Kocian[2], Petr Koudelka[2]**

[1] *Institute of Thermomechanics of the CAS, v. v. i., Dolejškova 1402/5, 182 00 Praha 8, Czech Republic*
[2] *CTU in Prague Faculty of Transportation Sciences, Department of Mechanics and Materials, Na Florenci 25, 110 00 Praha 1, Czech Republic*

ABSTRACT
We present a performance analysis of an asynchronous explicit time integration framework for partitioned elastodynamic problems with strongly heterogeneous temporal scales, motivated by simulations of Split Hopkinson Pressure Bar (SHPB) experiments. The proposed approach allows individual subdomains to advance with different time increments, enabling the small nonlinear specimen to be integrated using the time step required for accuracy while the long elastic bars are advanced asynchronously with substantially larger time steps. Numerical comparisons with conventional synchronous explicit integration, including LS-DYNA, demonstrate that the proposed method preserves accuracy while significantly reducing computational cost. The study further shows that the performance gains increase with the disparity between the characteristic time scales of the coupled subdomains, making the approach particularly attractive for large-scale SHPB simulations involving long elastic bars. These results demonstrate the potential of asynchronous time integration as an efficient alternative to conventional explicit solvers for computationally demanding impact simulations.

**Section:** RESEARCH PAPER

**Keywords:** Split Hopkinson Pressure Bar; finite element method; asynchronous time integration

**Citation:** R. Dvořák, L. Kocian, P. Koudelka, Performance analysis of an asynchronous time integration scheme for Split Hopkinson Pressure Bar simulations, Acta IMEKO, vol. 15 (2026) no. 3, pp. 1–8. DOI: 10.21014/actaimeko.v15i3.2459

**Section Editor:** Daniel Kytyr, Czech Academy of Sciences, Czechia

**Received:** Juli 7, 2026; **In Final Form:** September 7, 2026; **Published:** September, 2026.



**Funding:** The authors acknowledge funding by the Czech Science Foundation in project no. 25-18209S.
Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.
Radim Dvořák also acknowledges institutional support RVO:61388998.

**Corresponding Author:** Radim Dvořák, e-mail: radimd@it.cas.cz

## 1. INTRODUCTION

Elastodynamic simulations are essential in many engineering applications involving transient wave propagation, impact loading, and structural dynamics. A representative example is the Split Hopkinson Pressure Bar (SHPB) experiment, where stress waves propagate through long elastic bars and interact with a small specimen. Direct time integration methods remain the standard approach for solving such problems [1], and numerous schemes have been developed to improve stability, numerical dissipation, and energy conservation properties [2], [3].

A key challenge in SHPB simulations is the coexistence of multiple spatial and temporal scales. The specimen typically requires a fine mesh and a small time step to accurately capture its nonlinear response, whereas the bars can be discretized more coarsely and advanced using significantly larger time steps. However, conventional time integration employs a single global time step dictated by the most restrictive part of the model, resulting in substantial computational inefficiency because the bars usually contain the majority of the system degrees of freedom.

To alleviate this limitation, various asynchronous and multi-time-step integration methods have been proposed. Early approaches introduced non-uniform time stepping without domain decomposition [4]–[8]. More recent formulations combine asynchronous integration with domain decomposition and Lagrange multiplier coupling [9]–[12]. Other developments include explicit–implicit coupling strategies [13], macro–micro formulations [14], global asynchronous approaches [15], [16], acceleration-based coupling [17], and asynchronous variational integrators [18]. Although these methods have demonstrated promising results, many remain restricted to integer time-step ratios, require global solves, introduce artificial dissipation, or suffer from interface drifting. Similar asynchronous concepts have also been successfully applied to other multi-scale dynamical systems [19].

In our previous work [20], we proposed an asynchronous integration framework for partitioned elastodynamic problems that allows each subdomain to advance with its own local time step

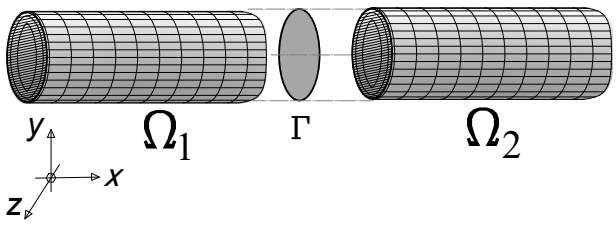



Figure 1. Illustration of a domain $\Omega$ partitioned into two subdomains $\Omega_i$ with a shared interface $\Gamma$.

while preserving a fully explicit formulation. The accuracy and stability of the proposed methodology were thoroughly validated on a range of one-dimensional, two-dimensional, and two-dimensional axisymmetric benchmark problems involving different material combinations and interface configurations. Building upon these validated foundations, the present paper focuses on the application of the method to SHPB simulations, where a small, finely discretized specimen is coupled to long elastic bars containing most of the degrees of freedom. This class of problems is particularly well suited to asynchronous time integration, allowing the computational performance and practical benefits of the proposed approach to be assessed.

## 2. STRONG FORM OF PARTITIONED ELASTODYNAMICS

In this section, the variational formulation of partitioned elastodynamics is presented. To simplify the exposition, the formulation is introduced for a domain composed of two elastic subdomains (see Figure 1), although the subsequent derivations are valid for an arbitrary number of subdomains. The stress tensor, traction vector, and displacement field in the $i$-th subdomain are denoted by $\boldsymbol{\sigma}_i(\boldsymbol{x})$, $\boldsymbol{t}_i(\boldsymbol{x})$, and $\boldsymbol{u}_i(\boldsymbol{x})$, respectively, where $\boldsymbol{x} \in \Omega$ represents the position of a material point. The computational domain $\Omega$ is partitioned into two subdomains $\Omega_i$, $i = 1, 2$. The subdomains are assumed to be non-overlapping, satisfying

$$\Omega_1 \cup \Omega_2 = \Omega\,, \quad \Omega_1 \cap \Omega_2 = \varnothing\,. \tag{1}$$

They are connected through the common interface $\Gamma$, whose outward normal is denoted by $\mathbf{n}_\Gamma$. Continuity across the interface is enforced by the compatibility condition

$$\ddot{\boldsymbol{u}}_i(\boldsymbol{x}) = \ddot{\boldsymbol{u}}_\Gamma(\boldsymbol{x}), \qquad \mathbf{x} \in \Gamma\,, \tag{2}$$

where $\ddot{\boldsymbol{u}}_\Gamma(\boldsymbol{x})$ denotes the interface acceleration field.

The formulation is derived from Hamilton's principle for constrained partitioned elastodynamics using the variational framework introduced by Park and Felippa [21], [22]. Following [23], the corresponding three-field variational functional is expressed as

$$\delta\mathcal{H}\left(\boldsymbol{u}, \boldsymbol{u}_\Gamma, \boldsymbol{\ell}\right) = \int_0^T \delta\Bigg\{ \sum_{i=1}^{2} \left\{\mathcal{T}_i\left(\boldsymbol{u}\right) - \mathcal{U}_i\left(\boldsymbol{u}\right) + \mathcal{W}_i\left(\boldsymbol{u}\right)\right\} + \mathcal{W}_\Gamma\left(\boldsymbol{u}, \boldsymbol{u}_\Gamma, \boldsymbol{\ell}\right) \Bigg\} \,\mathrm{d}t = 0\,, \tag{3}$$

with

$$\delta\mathcal{T}_i = \int_{\Omega_i} \delta\dot{\boldsymbol{u}}_i \cdot \rho\dot{\boldsymbol{u}}_i \,\mathrm{d}V\,, \quad \delta\mathcal{U}_i = \int_{\Omega_i} \delta\boldsymbol{\varepsilon}_i : \boldsymbol{\sigma}_i \,\mathrm{d}V\,, \quad \delta\mathcal{W}_i = \int_{\Omega_i} \delta\boldsymbol{u}_i \cdot \boldsymbol{b}_i \,\mathrm{d}V + \int_{\partial\Omega_i^N} \delta\boldsymbol{u}_i \cdot \boldsymbol{t}_i \,\mathrm{d}S\,, \tag{4}$$

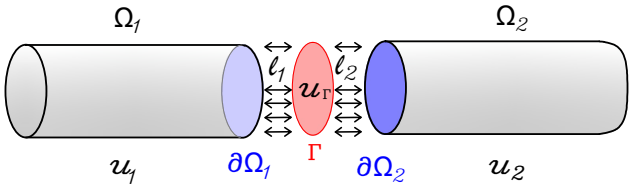



Figure 2. Discretized problem with three unknown fields.

where $\delta\mathcal{T}_i$, $\delta\mathcal{U}_i$, and $\delta\mathcal{W}_i$ denote the virtual kinetic energy, virtual strain energy, and virtual work of the external forces, respectively. Here, $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{\varepsilon}$ is the infinitesimal strain tensor, $\rho$ is the mass density, $\boldsymbol{b}$ denotes the body-force vector, $\boldsymbol{u}$ is the displacement field, $\boldsymbol{t}$ represents the prescribed traction, and $\partial\Omega_i^N$ is the Neumann part of the boundary. The final contribution in (3) accounts for the virtual work associated with the interface constraints,

$$\delta\mathcal{W}_\Gamma\left(\boldsymbol{u}, \boldsymbol{u}_\Gamma, \boldsymbol{\ell}\right) = \sum_{i=1}^{2} \int_\Gamma \delta\left[\boldsymbol{\ell}_i(\boldsymbol{u}_i - \boldsymbol{u}_\Gamma)\right] \,\mathrm{d}S\,, \tag{5}$$

where $\boldsymbol{u}_i(\boldsymbol{x}, t)$ and $\boldsymbol{\ell}_i(\boldsymbol{x}, t)$ denote the displacement field and the localized Lagrange multiplier field associated with the $i$-th subdomain, respectively, while $\boldsymbol{u}_\Gamma(\boldsymbol{x}, t)$ represents the frame displacement (see Figure 2).

## 3. FINITE ELEMENT DISCRETIZATION

A standard finite element discretization is adopted for all subdomains $\Omega_i$ and frame $\Gamma$,

$$\begin{aligned} \boldsymbol{u}_i(\boldsymbol{x}) &= \mathbf{N}_{u_i}(\boldsymbol{x})\,\mathbf{u}_i\,, \\ \boldsymbol{\ell}_i(\boldsymbol{x}) &= \mathbf{N}_{\lambda_i}(\boldsymbol{x})\,\boldsymbol{\lambda}_i\,, \\ \boldsymbol{u}_\Gamma(\boldsymbol{x}) &= \mathbf{N}_{u_\Gamma}(\boldsymbol{x})\,\mathbf{u}_\Gamma\,. \end{aligned} \tag{6}$$

The present formulation employs the Localized Lagrange Multiplier (LLM) method [24]–[26]. Unlike the classical LLM formulation, which uses Dirac-delta interpolation for the multipliers, the multiplier field is interpolated using displacement shape functions,

$$\boldsymbol{\ell}_i(\boldsymbol{x}) = \left[\mathbf{N}_{u_i}\,\mathbf{B}_i\right](\boldsymbol{x})\,\boldsymbol{\lambda}_i\,, \tag{7}$$

where $\mathbf{B}_i$ selects the interface degrees of freedom coupled to the frame.

Substituting the finite element approximation into the variational formulation (3) yields

$$\begin{bmatrix} \bar{\mathbf{K}} & \mathbf{B}\,\mathbf{M}^{\mathrm{B}} & \mathbf{0} \\ \mathbf{M}^{\mathrm{B}\mathsf{T}}\mathbf{B}^{\mathsf{T}} & \mathbf{0} & -\mathbf{C}_{\lambda u_\Gamma} \\ \mathbf{0} & -\mathbf{C}_{\lambda u_\Gamma}^{\mathsf{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \\ \mathbf{u}_\Gamma \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix}, \tag{8}$$

where

$$\bar{\mathbf{K}} = \mathbf{M}\,\mathrm{D}^2 + \mathbf{K}\,, \quad \mathrm{D} = \frac{\mathrm{d}}{\mathrm{d}t}\,.$$

Here, $\mathbf{M}$ and $\mathbf{K}$ in (8) are the block diagonal global mass matrix and the global stiffness matrix, respectively, consisting of standard FEM submatrices $\mathbf{M}_i$ and $\mathbf{K}_i$ and $\mathbf{f}$ is the row block vector of external load consisting of standard FEM subvectors $\mathbf{f}_i$.

The vector $\boldsymbol{\lambda}$ collects the Lagrange multiplier subvectors $\boldsymbol{\lambda}_i$, while $\mathbf{u}_\Gamma$ contains the frame displacement subvectors $\mathbf{u}_\Gamma^j$. Furthermore, $\mathbf{B} = \mathrm{diag}(\mathbf{B}_i)$ is a block-diagonal selection operator.

The matrices $\mathbf{M}^{\mathrm{B}}$ and $\mathbf{C}_{\lambda u_\Gamma}$ describe the coupling between subdomains and frames and are assembled from the block-diagonal

interface mass matrices $\mathbf{M}^{\mathrm{B}}_{ii}$ and the coupling blocks $\mathbf{C}_{\lambda_i u^j_\Gamma}$, respectively, where

$$\mathbf{M}^{\mathrm{B}}_{ii} = \int_\Gamma \left\{ [\mathbf{N}_{u_i}\,\mathbf{B}_i]^{\mathsf{T}}\,\mathbf{N}_{\lambda_i} \right\} \mathrm{d}\Gamma = \\ = \int_\Gamma \left\{ [\mathbf{N}_{u_i}\,\mathbf{B}_i]^{\mathsf{T}}\,[\mathbf{N}_{u_i}\,\mathbf{B}_i] \right\} \mathrm{d}S \tag{9}$$

is the $i$-th subdomain boundary interface mass matrix and

$$\mathbf{C}_{\lambda_i u^j_\Gamma} = \int_\Gamma \left\{ \mathbf{N}^{\mathsf{T}}_{\lambda_i}\,\mathbf{N}_{u^j_\Gamma} \right\} \mathrm{d}S = \\ = \int_\Gamma \left\{ [\mathbf{N}_{u_i}\,\mathbf{B}_i]^{\mathsf{T}}\,\left[\mathbf{N}_{u^j_\Gamma}\right] \right\} \mathrm{d}S \tag{10}$$

is the coupling operator obtained using a standard Mortar-method projection procedure on the common interface. Finally, to facilitate the solution procedure, the displacement constraints are differentiated twice with respect to time, transforming the kinematic compatibility condition into an acceleration continuity constraint. The resulting governing equations are

$$\mathbf{M}\,\ddot{\mathbf{u}} + \mathbf{K}\,\mathbf{u} = \mathbf{f} - \mathbf{B}\,\mathbf{M}^{\mathrm{B}}\,\boldsymbol{\lambda} \quad \text{(as Eqn of motion)}, \tag{11}$$

$$\mathbf{M}^{\mathrm{B}\mathsf{T}}\,\mathbf{B}^{\mathsf{T}}\,\ddot{\mathbf{u}} - \mathbf{C}_{\lambda u_\Gamma}\,\ddot{\mathbf{u}}_\Gamma = \mathbf{0} \quad \text{(as Kin. constraints)}, \tag{12}$$

$$\mathbf{C}^{\mathsf{T}}_{\lambda u_\Gamma}\,\boldsymbol{\lambda} = \mathbf{0} \quad \text{(as Interface equilibrium condition)}, \tag{13}$$

where the unknowns are the nodal displacements $\mathbf{u}$, nodal accelerations $\ddot{\mathbf{u}}$, frame accelerations $\ddot{\mathbf{u}}_\Gamma$, and the Lagrange multipliers $\boldsymbol{\lambda}$.

## 4. EXPLICIT FORMULAS OF ACCELERATION AND MULTIPLIERS

During the asynchronous integration, isolated solution of individual subdomains and subframes is required. Based on governing system of equation, we provide explicit formulas.

### 4.1. The $j$-th subframe problem

If we define the acceleration predictor $\tilde{\ddot{\mathbf{u}}}$ of all $n^j$ adjacent subdomains

$$\tilde{\ddot{\mathbf{u}}} = \mathbf{M}^{-\mathbf{1}}(\mathbf{f} - \mathbf{K}\,\mathbf{u})\,, \tag{14}$$

we can obtain $j$-th subframe acceleration from (11), (12), (13) as

$$\ddot{\mathbf{u}}^j_\Gamma = {\mathbf{M}^j_\Gamma}^{-1}\,\mathbf{f}^j_\Gamma\,, \tag{15}$$

where

$$\begin{gathered} {\mathbf{M}^j_\Gamma}^{-1} = \\ \left[\sum_{i=1}^{n} \left\{ \mathbf{C}_{\lambda_i u^j_\Gamma}{}^{\mathsf{T}} \left( \mathbf{M}^{\mathrm{B}}_{ii}{}^{\mathsf{T}}\,\mathbf{B}^{\mathsf{T}}_i \mathbf{M}^{-1}_i \mathbf{B}_i\,\mathbf{M}^{\mathrm{B}}_{ii} \right)^{-1} \mathbf{C}_{\lambda_i u^j_\Gamma} \right\} \right]^{-1}, \\ \mathbf{f}^j_\Gamma = \\ \sum_{i=1}^{n} \left\{ \mathbf{C}_{\lambda_i u^j_\Gamma}{}^{\mathsf{T}} \left( \mathbf{M}^{\mathrm{B}}_{ii}{}^{\mathsf{T}}\mathbf{B}^{\mathsf{T}}_i \mathbf{M}^{-1}_i \mathbf{B}_i \mathbf{M}^{\mathrm{B}}_{ii} \right)^{-1} \mathbf{M}^{\mathrm{B}}_{ii}{}^{\mathsf{T}}\,\mathbf{B}^{\mathsf{T}}_i\,\tilde{\ddot{\mathbf{u}}}_i \right\} \end{gathered} \tag{16}$$

are the frame mass matrix and frame load vector, respectively.

*Remark:* For large interface problems, the construction of the inverse interface operator (16) may become the computational bottleneck of the proposed method. In such cases, specialized techniques for the direct construction of the inverse matrix, such as the approach presented in Ref. [27], can be employed to significantly reduce the computational cost.

### 4.2. The $i$-th subdomain problem

Once we know the acceleration of all $m$ subframes connected to $i$-th subdomain, we can obtain $i$-th Lagrange multipliers as

$$\boldsymbol{\lambda}_i = \left( \mathbf{M}^{\mathrm{B}}_{ii}{}^{\mathsf{T}}\,\mathbf{B}^{\mathsf{T}}_i \mathbf{M}^{-1}_i \mathbf{B}_i\,\mathbf{M}^{\mathrm{B}}_{ii} \right)^{-1} \cdot \\ \cdot \left( \mathbf{M}^{\mathrm{B}}_{ii}{}^{\mathsf{T}}\,\mathbf{B}^{\mathsf{T}}_i \tilde{\ddot{\mathbf{u}}}_i - \sum_{j=1}^{m} \mathbf{C}_{\lambda_i u^j_\Gamma}\,\ddot{\mathbf{u}}^j_\Gamma \right), \tag{17}$$

and finally,

$$\ddot{\mathbf{u}}_i = \tilde{\ddot{\mathbf{u}}}_i - \mathbf{M}^{-1}_i\,\mathbf{B}_i\,\mathbf{M}^{\mathrm{B}}_{ii}\,\boldsymbol{\lambda}_i \tag{18}$$

is subdomain acceleration.

## 5. TEMPORAL DISCRETIZATION BY DTI METHODS

We use classical central difference method [1]. The solution of the semidiscrete system (11)-(13) in synchronous manner can be then summarized in the following compact algorithm

**Algorithm of synchronous solution:**

**Step 1.** Solve floating subdomains;

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

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

**Step 2.** Evaluate acceleration of the frame and Lagrange multipliers;

$$(\ddot{\mathbf{u}}_\Gamma)^{n+1} = \mathbf{M}^{-1}_\Gamma\,(\mathbf{f}_\Gamma)^{n+1}\,,$$

$$(\boldsymbol{\lambda})^{n+1} = \left( \mathbf{M}^{\mathrm{B}\mathsf{T}}\,\mathbf{B}^{\mathsf{T}}\mathbf{M}^{-\mathbf{1}}\mathbf{B}\,\mathbf{M}^{\mathrm{B}} \right)^{-1} \cdot \\ \cdot \left( \mathbf{M}^{\mathrm{B}\mathsf{T}}\mathbf{B}^{\mathsf{T}}(\tilde{\ddot{\mathbf{u}}})^{n+1} - \mathbf{C}_{\lambda u_\Gamma}(\ddot{\mathbf{u}}_\Gamma)^{n+1} \right)\,.$$

**Step 3.** Evaluate acceleration of the coupled system and velocity;

$$(\ddot{\mathbf{u}})^{n+1} = (\tilde{\ddot{\mathbf{u}}})^{n+1} - \mathbf{M}^{-1}\,\mathbf{B}\,\mathbf{M}^{\mathrm{B}}\,(\boldsymbol{\lambda})^{n+1}\,,$$

$$(\dot{\mathbf{u}})^{n+1} = (\dot{\mathbf{u}})^n + \frac{\Delta t}{2}\left( (\ddot{\mathbf{u}})^n + (\ddot{\mathbf{u}})^{n+1} \right)\,.$$

## 6. ISOLATION OF THE FRAME BY SYSTEM REDUCTION

To solve the frame $\Gamma$ independently of the surrounding subdomains, we define interface regions $\Omega^{\mathbf{r}}_i$ based on the domains of dependence and influence over the interval $\epsilon_i \Delta t^c_i$ (see Figure 3), where

$$\epsilon_i \geq 1 + \Delta t_i / \Delta t^c_i\,. \tag{19}$$

Here, $\Delta t^c_i$ denotes the critical time step of the CDM scheme and

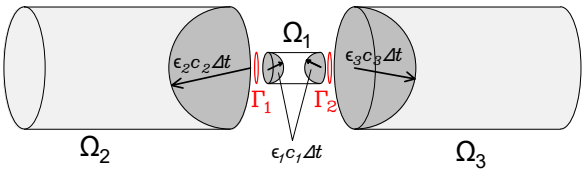



Figure 3. Domain of dependence/influence of the frame, where $c_i$ is wave speed.

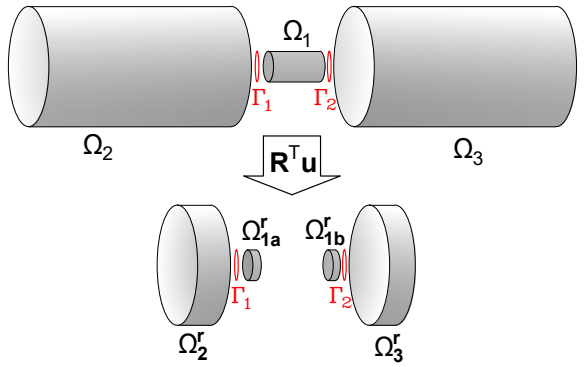



Figure 4. The transformation of subdomains problem to multiple isolated interface regions problem.

$\Delta t_i$ the actual integration step. The interface regions are extracted by the Boolean transformation

$$\mathbf{u}^{\mathbf{r}} = \mathbf{R}^{\mathsf{T}}\mathbf{u}, \tag{20}$$

which selectively retains the degrees of freedom belonging to the interface regions. Substituting this transformation into (8) yields

$$\begin{bmatrix} \bar{\mathbf{K}}^{\mathbf{r}} & \mathbf{B}^{\mathbf{r}}\mathbf{M}^{\mathsf{B}} & \mathbf{0} \\ \mathbf{M}^{\mathsf{B}\mathsf{T}}\mathbf{B}^{\mathbf{r}\mathsf{T}} & \mathbf{0} & -\mathbf{C}_{\lambda u_\Gamma} \\ \mathbf{0} & -\mathbf{C}^{\mathsf{T}}_{\lambda u_\Gamma} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u}^{\mathbf{r}} \\ \boldsymbol{\lambda} \\ \mathbf{u}_\Gamma \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^{\mathbf{r}} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix}, \tag{21}$$

where $\bar{\mathbf{K}}^{\mathbf{r}} = \mathbf{M}^{\mathbf{r}}\,\mathrm{D}^2 + \mathbf{K}^{\mathbf{r}}$, $\mathrm{D} = \dfrac{\mathrm{d}}{\mathrm{d}t}$,

with

$$\begin{aligned} \mathbf{K}^{\mathbf{r}} &= \mathbf{R}\mathbf{K}\mathbf{R}^{\mathsf{T}}, \quad \mathbf{M}^{\mathbf{r}} = \mathbf{R}\mathbf{M}\mathbf{R}^{\mathsf{T}}, \\ \mathbf{f}^{\mathbf{r}} &= \mathbf{R}\mathbf{f}, \quad \mathbf{B}^{\mathbf{r}} = \mathbf{R}\mathbf{B}. \end{aligned} \tag{22}$$

The reduced model consists only of the frame and adjacent interface regions (see Figure 4). Since (21) retains the structure of the original system, the same solution procedure can be employed.

## 7. ASYNCHRONOUS SCHEME

At any stage of the time integration, at least one frame or one subdomain can always be solved independently:

- **Frame** $\Gamma_j$ is solvable if $(t_i^{n+1} > t_\Gamma)\,\forall$ connected subdomains. Then the frame is integrable with step $\Delta t_\Gamma^j = \min\{t_i^{n+1} - t_\Gamma\}$.
- **Subdomain** $\Omega_i$ is solvable if $t_i^{n+1} = t_\Gamma^j\,\forall$ connected frames. Then the subdomain is integrable with step $\Delta t_i$.

The asynchronous time scheme is then implemented as follows:

### 1. INITIALIZATION:

**1.1.** Set the subdomain $\mathbf{t} = \{t_i\} = \mathbf{0}$ and subframe time vector $\mathbf{t}_\Gamma = \{t_\Gamma^j\} = \mathbf{0}$.

**1.2.** Initialize the interface regions: $\{\mathbf{u}^{\mathbf{r}}, \dot{\mathbf{u}}^{\mathbf{r}}, \ddot{\mathbf{u}}^{\mathbf{r}}\}_i = \mathbf{R}_i\{\mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}}\}_i$.

**1.3.** Identify the nearest desired subdomain time levels to solve $\mathbf{t}^{n+1} = \mathbf{t} + \Delta\mathbf{t}$.

### 2.A SOLUTION OF SOLVABLE FRAME:

**2.A.1.** If the subframe $j$ is solvable, proceed with following subroutine. Otherwise proceed with subroutine **2.B**.

**2.A.2.** Determine subframe timestep from connected subdomains $\Delta t_\Gamma = \min\{\mathbf{t}^{n+1} - t_\Gamma\}$.

**2.A.3.** a) for $\Omega_i^{\mathbf{r}}$ at time $t_\Gamma^j + \Delta t_\Gamma^j$ {

i) $(\mathbf{u}_i^{\mathbf{r}})^{n+1} = (\mathbf{u}_i^{\mathbf{r}})^n + \Delta t\,(\dot{\mathbf{u}}_i^{\mathbf{r}})^n + \frac{\Delta t^2}{2}(\ddot{\mathbf{u}}_i^{\mathbf{r}})^n$

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

b) $\ddot{\mathbf{u}}_\Gamma = \mathbf{M}_\Gamma^{-1}\,\mathbf{f}_\Gamma$

c) for $\Omega_i^{\mathbf{r}}$ {

i) $\boldsymbol{\lambda}_i = \left(\mathbf{B}_i^{\mathsf{T}}\mathbf{M}_i^{-1}\mathbf{B}_i\right)^{-1} \cdot \left(\mathbf{B}_i^{\mathbf{r}\mathsf{T}}\tilde{\ddot{\mathbf{u}}}_i^{\mathbf{r}} - \mathbf{C}_{\lambda_i u_\Gamma}\ddot{\mathbf{u}}_\Gamma\right)$.

ii) $\ddot{\mathbf{u}}_i^{\mathbf{r}} = \tilde{\ddot{\mathbf{u}}}_i^{\mathbf{r}} - \mathbf{M}_i^{\mathbf{r}\,-1}\,\mathbf{B}_i^{\mathbf{r}}\,\boldsymbol{\lambda}_i$

iii) $(\dot{\mathbf{u}}^{\mathbf{r}})^{n+1} = (\dot{\mathbf{u}}^{\mathbf{r}})^n + \frac{\Delta t}{2}\left((\ddot{\mathbf{u}}^{\mathbf{r}})^n + (\ddot{\mathbf{u}}^{\mathbf{r}})^{n+1}\right)$ }

d) $t_\Gamma^j = t_\Gamma^j + \Delta t_\Gamma^j$

### 2.B SOLUTION OF SOLVABLE SUBDOMAINS:

**2.B.1** If a subdomain $\Omega_i$ is solvable, proceed with following subroutine.

**2.B.2** for $\Omega_i$ at time $t_i + \Delta t_i$ {

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

b) $\tilde{\ddot{\mathbf{u}}}_i = \hat{\mathbf{M}}_i^{-1}(\mathbf{f}_i - \mathbf{K}_i\,(\mathbf{u}_i))$,

c) $\ddot{\mathbf{u}}_i = \tilde{\ddot{\mathbf{u}}}_i - \mathbf{M}_i^{-1}\,\mathbf{B}_i\,\mathbf{C}_{u\lambda_i}\,\boldsymbol{\lambda}_i$,

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

e) Reset interface region $\Omega_i^{\mathbf{r}}$ : $\{\mathbf{u}_i^{\mathbf{r}}, \dot{\mathbf{u}}_i^{\mathbf{r}}, \ddot{\mathbf{u}}_i^{\mathbf{r}}\} = \mathbf{R}_i\{\mathbf{u}_i, \dot{\mathbf{u}}_i, \ddot{\mathbf{u}}_i\}$.

f) $t_i = t_i + \Delta t_i$, $t_i^{n+1} = t_i^{n+1} + \Delta t_i$ }

### 3. TERMINATION CONDITION:

*if* $\min\{\mathbf{t}\} < T$, repeat phases **2.A** and **2.B**, *else* terminate.

## 8. COMPUTATIONAL ANALYSIS

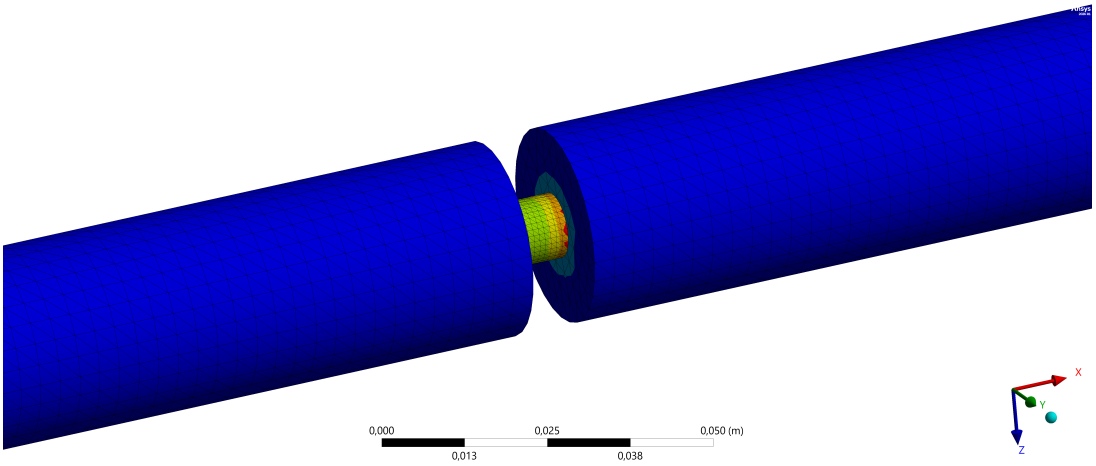
We set up SHPB test represented by problem of coupling of three subdomains: input and output bar and specimen (see Table 1).

| subdomain | Material parameters | DoFs | Critical time step $\Delta t^c$ [s] |
|---|---|---|---|
| Input bar | fixed | 220 000 | 1 |
| Output bar | fixed | 220 000 | 1 |
| Specimen | *varying* | 7 000 | 1 - 32 |

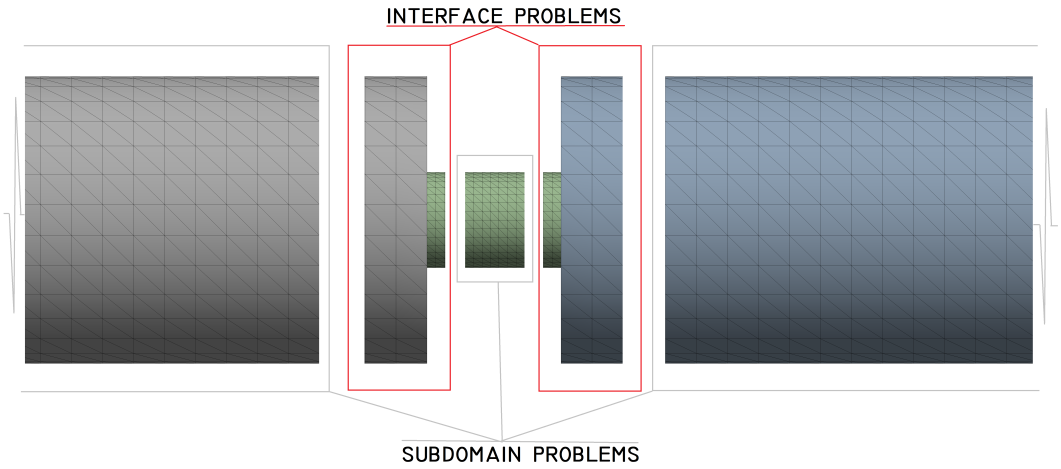
Table 1. Fundamental model parameters

Representative mesh density was chosen. Material parameters of bars were fixed and specimen material parameters were scaled in order to vary the ratio

$$r = \Delta t^c_{\text{bars}} \,/\, \Delta t^c_{\text{specimen}}. \tag{23}$$




(a) The setup




(b) Outline of domain decomposition

Figure 5. SHPB model consisting of a small specimen discretized with a fine mesh and connected to two long bars discretized with a coarser mesh. Despite the coarse discretization, the bars contain the majority of the system degrees of freedom.

This procedure lets us effectively emulate real problem, when specimen undergoes large deformation with multiple nonlinear aspects, therefore

$$\Delta t^c_{\text{specimen}} \ll \Delta t^c_{\text{bars}} \,. \tag{24}$$

On the other hand, bars typically have the majority of degrees of freedom, especially in large SHPB set-ups.

Consequently, a multiscale coupling problem is obtained (see Figure 5), where the subdomains are connected through an interface with a non-matching mesh (see Figure 5a). The interface regions used for asynchronous integration are illustrated in Fig. 5b. In the following analysis, we demonstrate the superior performance of the proposed asynchronous integrator compared to the commercial software LS-DYNA. We also highlight the importance of choosing asynchronous time integration over simply increasing the number of CPUs when aiming to reduce the overall simulation time.

### 8.1. Performance of the asynchronous integrator compared to LS-DYNA

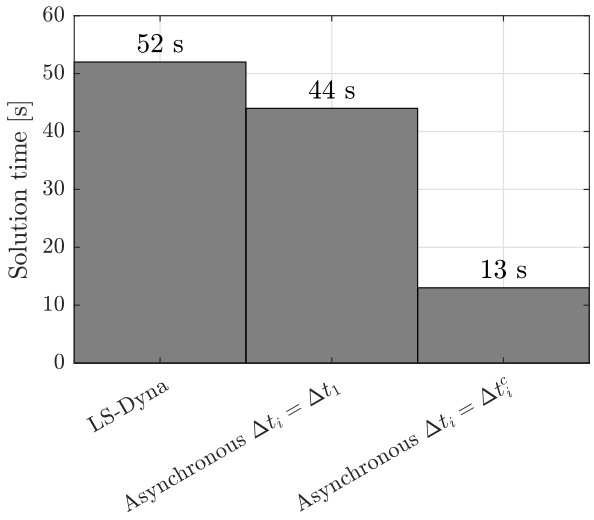
Figure 6a compares the absolute simulation times obtained with LS-DYNA, the proposed solver operating in a synchronous mode using a single global time step, and the proposed asynchronous formulation. It can be seen that LS-DYNA and the synchronous version of the proposed solver exhibit comparable computational performance, while the asynchronous formulation requires substantially less computational time.

The influence of the time-step ratio on computational performance is further examined in Fig. 6b. The proposed asynchronous integrator was executed on a single CPU core and exhibits only a negligible increase in computational cost as the time-step ratio grows. In contrast, the computational cost of LS-DYNA increases noticeably with increasing time-step ratio. Several curves corresponding to different numbers of CPU cores are shown. Although parallel execution reduces the overall solution time, the effect is relatively limited. The best performance is achieved with eight CPU cores, which decreases the solution time by approximately 30-50 % compared to the single-core execution. Nevertheless, the dependence of the computational cost on the time-step ratio remains significant for all investigated numbers of CPU cores.

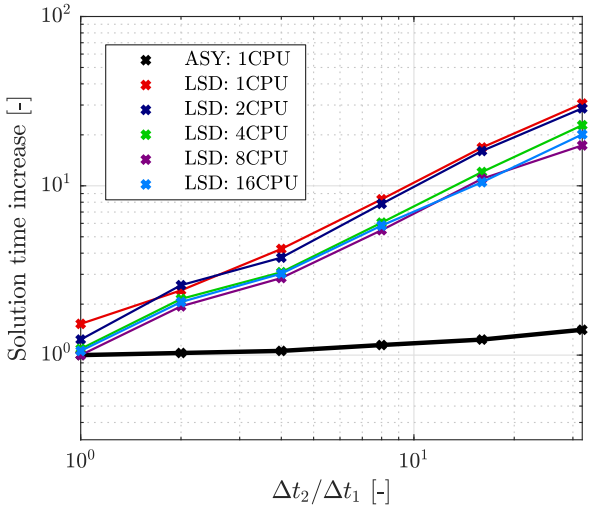
### 8.2. Benefits of asynchronous integration over increasing the number of CPU cores

Figure 7 compares the speedup achieved by increasing the number of CPU cores in LS-DYNA with the speedup obtained using the proposed asynchronous integration strategy. In both cases, the speedup is measured relative to the single-core synchronous LS-DYNA simulation.

The speedup achieved by parallel execution in LS-DYNA remains nearly independent of the time-step ratio (see Figure 7a). For all investigated cases, the acceleration is approximately con-

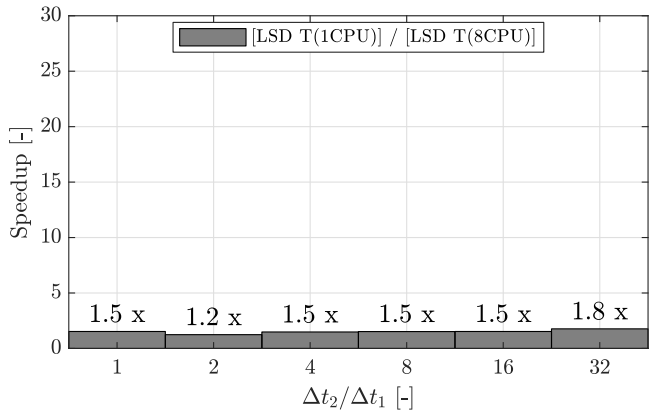



(a) Simulation duration of synchronous computations of both LS-Dyna and Asynchronous scheme and asynchronous computation of Asynchronous scheme

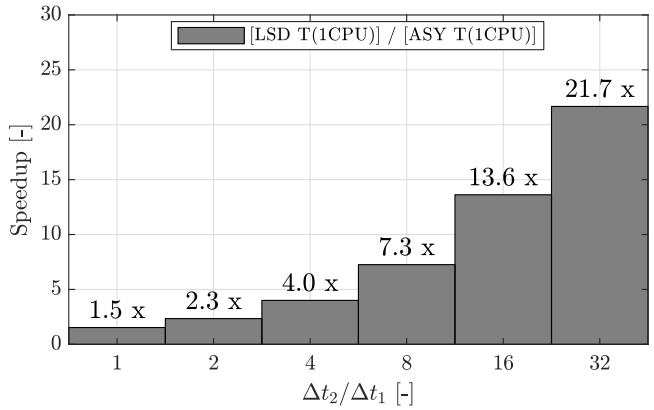



(b) Slowdowns comparison of Asynchronous scheme with 1 CPU core with LS-Dyna with different numbers of used cores

Figure 6. Speedup and slowdown of simulation time for different methods and numbers of CPU cores.




(a) Speedup of 8 core solution vs 1 core solution within LS-Dyna




(b) Speedup of 1 core solution by Asynchronous solver vs 1 core solution by LS-Dyna

Figure 7. Comparison of two approaches for reducing simulation time: increasing the number of CPU cores (top) and adopting the asynchronous integration strategy (bottom).

stant, reaching an average value of about 1.5. This indicates that increasing the number of CPU cores provides only a moderate reduction in computational time.

In contrast, the speedup obtained with the asynchronous formulation strongly depends on the time-step ratio (see Figure 7b). As the ratio between the characteristic time steps of the coupled subdomains increases, the computational advantage of asynchronous integration becomes increasingly pronounced. For the largest investigated ratio of 32, the asynchronous formulation achieves a speedup of approximately 22 relative to the reference synchronous LS-DYNA simulation.

## 9. CONCLUSIONS

The presented results clearly demonstrate that while additional *CPU cores provide only a moderate reduction* in solution time, the *benefits of asynchronous integration grow* with increasing disparity between the characteristic time scales of the coupled subdomains.

This makes asynchronous integration particularly attractive for SHPB simulations, which combine a small nonlinear specimen requiring a short time step with long elastic bars containing a large number of degrees of freedom. In the proposed formulation, the specimen is integrated using the small time step required to accurately capture its response, whereas the bars are advanced asynchronously using substantially larger time steps. As a result, the bars require far fewer time-integration updates than in a synchronous simulation, despite accounting for the majority of the degrees of freedom. Consequently, the overall computational cost becomes dominated by the specimen, while the contribution of the bars remains minimal. Experimental setups with bars tens of meters long can therefore be simulated in nearly the same time as a model containing only the specimen.

The present study considered models with approximately one million degrees of freedom. For substantially larger models, containing tens of millions of degrees of freedom, the influence of parallelization is expected to become more significant. Nevertheless, asynchronous integration and parallel computing are complementary approaches, and their combination offers the potential for further substantial reductions in computational cost for large-scale simulations.

Future work will focus on integrating the proposed asynchronous time-integration framework into existing open-source finite element software, particularly SfePy [28], [29], where it can be combined with advanced multiscale computational frameworks. Such an implementation would enable efficient large-scale simulations involving coupled problems with widely separated temporal and spatial scales, extending the applicability of the proposed methodology to a broader range of engineering and scientific applications.

## AUTHORS' CONTRIBUTION

All authors contributed equally to this work.

## ACKNOWLEDGEMENT

The authors acknowledge funding by the Czech Science Foundation in project no. 25-18209S.

Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

Radim Dvořák also acknowledges institutional support RVO:61388998.

## DATA MANAGEMENT PLAN

The supporting dataset is available at https://doi.org/10.5281/zenodo.21136040.

## REFERENCES

[1] T. J. R. Hughes, The finite element method: linear static and dynamic finite element analysis, Dover Publications, Mineola, NY, 2000, ISBN: 0-486-41181-8.

[2] W.-B. Wen, L. He, T. Liu, S. Deng, S. Duan, A novel hybrid sub-step explicit time integration method with cubic b-spline interpolation and momentum corrector technique for linear and nonlinear dynamics, Nonlinear Dynamics, vol. 110, 2022, pp. 2685–2714. DOI: 10.1007/s11071-022-07740-9

[3] W.-B. Wen, Y. Hong, P. Wang, A generalized three-sub-step composite implicit time integration method with improved accuracy and convergence for linear and nonlinear dynamics, International Journal of Structural Stability and Dynamics, vol. 26, 2026, no. 25. DOI: 10.1142/S021945542650224X

[4] T. Belytschko, R. Mullen, Mesh partitions of explicit-implicit time integration, Formulations and computational algorithms in finite element analysis : U.S.-Germany symposium, Massachusetts Institute of Technology, Cambridge, 1977, ISBN: 9780262021272.

[5] T. Belytschko, R. Mullen, Stability of explicit-implicit mesh partitions in time integration, International Journal for Numerical Methods in Engineering, vol. 12, 1978, no. 10, pp. 1575–1586. DOI: 10.1002/nme.1620121008

[6] P. Smolinski, Subcycling integration with non-integer time steps for structural dynamics problems, Computers & Structures, vol. 59, 1996, pp. 273–281. DOI: 10.1016/0045-7949(95)00256-1

[7] W. J. T. Daniel, A study of the stability of subcycling algorithms in structural dynamics, Computer Methods in Applied Mechanics and Engineering, vol. 156, 1998, no. 1-4, pp. 1–13. DOI: 10.1016/S0045-7825(97)00140-0

[8] P. Smolinski, S. Sleith, T. Belytschko, Stability of an explicit multitime step integration algorithm for linear structural dynamics equations, Computational Mechanics, vol. 18, 1996, pp. 236–244. DOI: 10.1007/BF00369941

[9] A. Combescure, 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, 01 2002, pp. 1129–1157. DOI: 10.1016/S0045-7825(01)00190-6

[10] V. Faucher, H. Bung, A. Combescure, Europlexus: a domain decomposition method in explicit dynamics, International Conference on Supercomputing in Nuclear Applications SNA, The publisher, Paris (France), 2003, p. 12.

[11] A. Prakash, 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, 2004, no. 13, pp. 2183–2204. DOI: 10.1002/nme.1136

[12] F.-E. Fekak, M. Brun, A. Gravouil, B. Depale, A new heterogeneous asynchronous explicit–implicit time integrator for nonsmooth dynamics, Computational Mechanics, vol. 60, Jul 2017, no. 1, pp. 1–21. DOI: 10.1007/s00466-017-1397-0

[13] T. Chantrait, J. Rannou, A. Gravouil, Low intrusive coupling of implicit and explicit time integration schemes for structural dynamics: Application to low energy impacts on composite structures, Finite Elements in Analysis and Design, vol. 86, 2014, pp. 23–33. DOI: 10.1016/j.finel.2014.03.011

[14] A. Gravouil, A. Combescure, M. Brun, Heterogeneous asynchronous time integrators for computational structural dynamics, International Journal for Numerical Methods in Engineering, vol. 102, 2015, no. 3-4, pp. 202–232. DOI: 10.1002/nme.4818

[15] W. Subber, K. Matouš, Asynchronous space–time algorithm based on a domain decomposition method for structural dynamics problems on non-matching meshes, Computational Mechanics, vol. 57, 2016, no. 2, pp. 211–235. DOI: 10.1007/s00466-015-1228-0

[16] A. S. Seibold, D. J. Rixen, Asynchronous time integration in structural mechanics, PAMM, vol. 19, 2019, no. 1, p. e201900495. DOI: 10.1002/pamm.201900495

[17] S. S. Cho, R. Kolman, J. A. González, 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, 2019, no. 5, pp. 276–302. DOI: 10.1002/nme.6027

[18] D. B. Jadhav, D. Phansalkar, K. Weinberg, M. Ortiz, S. Leyendecker, A spatio-temporally adaptive asynchronous variational integrator for a phase field model of dynamic fracture, Mechanics of Materials, vol. 215, 2026, p. 105604. DOI: 10.1016/j.mechmat.2026.105604

[19] K. M. Zhang, A. S. Wexler, An asynchronous time-stepping (ats) integrator for atmospheric applications: Aerosol dynamics, Atmospheric Environment, vol. 40, 2006, no. 24, pp. 4574–4588. DOI: 10.1016/j.atmosenv.2006.03.048

[20] R. Dvořák, R. Kolman, M. Mračko, J. Kopačka, T. Fíla, O. Jiroušek, J. Falta, M. Neuhäuserová, V. Rada, V. Adámek, J. A. González, Energy-conserving interface dynamics with asynchronous direct time integration employing arbitrary time steps, Computer Methods in Applied Mechanics and Engineering, vol. 413, Aug. 2023, p. 116110. DOI: 10.1016/j.cma.2023.116110

[21] K. C. Park, C. A. Felippa, A variational framework for solution method developments in structural mechanics, Journal of Applied Mechanics, vol. 65, 1998, no. 1, pp. 242–249.

[22] K. C. Park, C. A. Felippa, A variational principle for the formulation of partitioned structural systems, International Journal for Numerical Methods in Engineering, vol. 47, 2000, pp. 395–418. DOI:10.1002/(SICI)1097-0207(20000110/30)47:1/3<395::AID-NME777>3.0.CO;2-9

[23] J. A. González, J. Kopačka, R. Kolman, 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, 2021, no. 17, pp. 4609–4636. DOI: 10.1002/nme.6739

[24] K. C. Park, C. A. Felippa, U. A. Gumaste, A localized version of the metod of Lagrange multipliers and its applications, Computational Mechanics, vol. 24, 2000, pp. 476–490. DOI: 10.1007/s004660050007

[25] K. C. Park, C. A. Felippa, G. Rebel, A simple algorithm for localized construction of non-matching structural interfaces, International Journal for Numerical Methods in Engineering, vol. 53, 2002, no. 9, pp. 2117–2142. DOI: 10.1002/nme.374

[26] R. Dvořák, J. A. González, On the automatic construction of interface coupling operators for non-matching meshes by optimization methods, Computer Methods in Applied Mechanics and

Engineering, vol. 432, 2024, p. 117336.
DOI: 10.1016/j.cma.2024.117336

[27] R. Cimrman, R. Kolman, J. A. González, K. Park, Higher-order inverse mass matrices for the explicit transient analysis of heterogeneous solids, International Journal for Numerical Methods in Engineering, vol. 125, 2024, no. 11, p. e7457.
DOI: 10.1002/nme.7457

[28] R. Cimrman, Fast evaluation of finite element weak forms using python tensor contraction packages, Advances in Engineering Software, vol. 159, 2021, p. 103033.
DOI: 10.1016/j.advengsoft.2021.103033

[29] R. Cimrman, V. Lukeš, E. Rohan, Multiscale finite element calculations in python using SfePy, 2018.
DOI: 10.1007/s10444-019-09666-0