

# Asynchronous space–time algorithm based on a domain decomposition method for structural dynamics problems on non-matching meshes

Waad Subber<sup>1</sup> · Karel Matouš<sup>2</sup>

Received: 18 May 2015 / Accepted: 18 November 2015 / Published online: 7 December 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** Large-scale practical engineering problems featuring localized phenomena often benefit from local control of mesh and time resolutions to efficiently capture the spatial and temporal scales of interest. To this end, we propose an asynchronous space–time algorithm based on a domain decomposition method for structural dynamics problems on non-matching meshes. The three-field algorithm is based on the dual-primal like domain decomposition approach utilizing the localized Lagrange multipliers along the space and time common-refinement-based interface. The proposed algorithm is parallel in nature and well suited for a heterogeneous computing environment. Moreover, two-levels of parallelism are embedded in this novel scheme. For linear dynamical problems, the algorithm is unconditionally stable, shows an optimal order of convergence with respect to space and time discretizations as well as ensures conservation of mass, momentum and energy across the non-matching grid interfaces. The method of manufactured solutions is used to verify the implementation, and an engineering application is considered, where a sandwich plate is impacted by a projectile.

**Keywords** Asynchronous time integration · Multi-time-step methods · Local Lagrange multipliers · Domain decomposition methods · Non-matching grids · High-performance computing

## 1 Introduction

Large-scale multi-physics simulations on parallel computers often require a localized treatment of the space and time resolutions. Multiple spatial and temporal scales in structural dynamics problems usually arise from material heterogeneities, e.g. composite panels with stiff skin and soft core [35], complex loading conditions, such as in impact [62] or shock [57], and many other scenarios. To effectively capture the multiscale behavior of such problems, it is necessary to locally control the spatial and temporal resolutions of the numerical scheme. High spatial and temporal resolutions may be only required in a small part of the computational domain where fast dynamics or localized phenomena are exhibited, whereas a coarser discretization is sufficient in the majority of the structure. Therefore, computing power should be directed toward the high resolution regions of the computational domain.

Modern advances in high-performance computing facilitate parallel simulations of large-scale scientific and engineering problems using domain decomposition methods [41,53,58,63]. Domain decomposition may arise naturally from partitioning the computational domain into subregions based on physical considerations, such as in particulate composites [36], partitioning to accommodate parallel computing [41,58], or due to temporal and spatial scales of interest [1,52]. To accommodate these local spatial and temporal scales, the computational subdomains should be independently refined in the spatial and temporal dimen-

---

✉ Karel Matouš  
kmatous@nd.edu

Waad Subber  
wsubber@nd.edu

<sup>1</sup> Center for Shock Wave-processing of Advanced Reactive Materials (C-SWARM), University of Notre Dame, 117 Cushing Hall, Notre Dame, IN 46556, USA

<sup>2</sup> Department of Aerospace and Mechanical Engineering, Center for Shock Wave-processing of Advanced Reactive Materials (C-SWARM), University of Notre Dame, 367 Fitzpatrick Hall of Engineering, Notre Dame, IN 46556, USA

sions. However, this localized resolution treatment leads to subdomains with non-matching meshes along the interfaces and time integration with unequal time steps. Therefore, we present a novel algorithm based on the three-field domain decomposition technique to effectively address the issues of non-matching localized meshes and asynchronous time increments.

One of the popular domain decomposition methods for parallel computing, FETI, was proposed initially for elliptic boundary value problems by Farhat and Roux [17], and it was later extended to transient simulations [13]. Typically, in the dual domain decomposition methods, the continuity is enforced by global Lagrange multipliers (the dual-variables). Park et al. [45–47] proposed an alternative approach that departs from the direct subdomain-to-subdomain constraint. In particular, local Lagrange multipliers are used to constrain domains to an intermediate interface or a frame. Such an intermediate interface is employed by Brezzi and Marini [8] to handle domains with non-matching meshes. A crucial component of these schemes is the selection of the frame (interface) between the subdomains. A zero moment rule which preserves a constant state of stress along the interface was proposed in [45,56]. An alternative procedure for the frame or a “common” interface is the common-refinement-based approach proposed by Jiao and coworkers [29,30]. The common refinement technique minimizes a certain  $L_2$  norm between the source and target functions and preserves physical quantities along the interface [29,30]. Another popular class of domain decomposition methods for non-matching meshes is the mortar method. The mortar method allows nonconforming decomposition of the global domain and provides an optimal coupling between the subdomains [2,3,41]. A discussion on the mortar based domain decomposition for fluid-structure interaction problems and its computational cost is provided in [16,56]. For a comparison among different methods for data transfer between non-matching meshes, see [11,29].

The non-matching mesh resolution may necessitate asynchronous time integration. In other words, in order to satisfy the stability and accuracy requirements of the localized time integration scheme, stiff subdomains (i.e. subdomains with a small mesh size) may require much smaller time steps than those occupying the rest of the structure, where larger time steps might be permitted. Mixed implicit-explicit time integration also may be desirable. To this end, Belytschko and Mullen [4,5] proposed a nodal partitioning mixed explicit-implicit method while Hughes and Liu [26,27] introduced an element-based partitioning scheme. Moreover, several sub-cycling methods for both first- and second-order problems were proposed in [6,43,59,60]. Recognizing the potential of domain decomposition methods, the FETI-like transient integrator was developed by Combescure and Gravouil [10,21] for subdomains with matching interfaces. This algorithm

is unconditionally stable for synchronous time integration, but can be energy dissipative in the case of asynchronous time stepping. Prakash and Hjelmstad [51] and Prakash et al. [52] proposed a stable and non-dissipative sub-cycling scheme based on binary trees for parallelism. Unfortunately, the sequential time flow makes this method less desirable for high-performance computing. Very recently, Karimi and Nakshatrala [34] presented an approach for multi-time stepping on many subdomains combining the works of Combescure and Gravouil [21] and Prakash and Hjelmstad [51]. Gates et al. [20], and Beneš and Matouš [7] proposed a computational scheme for non-linear problems with both synchronous and asynchronous time stepping utilizing the variational integrators and localized Lagrange multipliers. While this method is highly parallelizable, it is only conditionally stable for asynchronous time steps and requires matching spatial discretizations. A general methodology to couple different classes of time integration schemes with different time steps is described in [22,39], and this methodology is extended to the case of non-matching meshes marching in time with synchronous [24] and asynchronous [18] time integration. Such a framework is based on the mortar method while discretizing the Lagrange multipliers directly on an interface mesh. Since the kinematic constraint is interpolated between the adjacent subdomains, a dissipation of the interface energy may occur when using asynchronous time stepping. Furthermore, the method is based on the tree/linked decoupling approach, which is a serial algorithm where one subdomain needs to be solved first followed by the solution to the adjacent one.

In this work, we develop the parallel asynchronous space-time algorithm based on the domain decomposition method (PASTA-DDM) for structural dynamics problems on non-matching meshes. This three-field algorithm is based on the dual-primal like domain decomposition approach, whereby we define a global primal variable and use a local dual variable to enforce the continuity requirements. Our approach circumvents the need to interpolate the kinematic constraints between the adjacent subdomains with coarse and fine timescales (such as in [7,10,20,21,51,52]). Instead following the localized Lagrange multipliers methodology, we interpose a common interface among the subdomains and define a second global primal variable (the frame velocity). This technique enables a local coupling along the interface and leads to high-degree of parallelism in spatial dimension. For accuracy and fulfillment of the conservation requirements, we discretize the common interface (the frame) using the common-refinement-based technique. For asynchronous time integration, we extend the concept of the common refinement to the temporal direction, and employ a generalized  $\alpha$  like method for the local Lagrange multiplier field. In other works [7,20,51,52], the frame velocity is interpolated between the coarse and fine timescales which may limit the

stability and parallelism of the asynchronous time integration algorithm. In our approach however, the temporal discretization leads to favorable conservation properties and enables a fine grain time-based parallelism within each subdomain. In particular, the synchronization time step in PASTA-DDM can be viewed as the coarse time grid correction in the *Parareal* algorithm [19,37]. Therefore, the PASTA-DDM algorithm offers two-levels of inherent parallelism. For linear problems, we prove by energy method that the PASTA-DDM is an unconditionally stable and energy preserving scheme in discrete sense. Moreover, we verify the jump conditions along the non-matching interfaces and show that PASTA-DDM ensures conservation of mass, momentum and energy along the common interface. For the engineering application, we consider a sandwich plate impacted by a projectile. In this problem, we employ the mixed implicit-explicit integration with heterogeneous time steps and non-matching meshes. Based on the physical properties and local elastic wave speeds of the problem, the computational domain is split into multiple non-overlapping subdomains (i.e. 9 subdomains with cross points) leading to a large degree of spatial and temporal asynchrony.

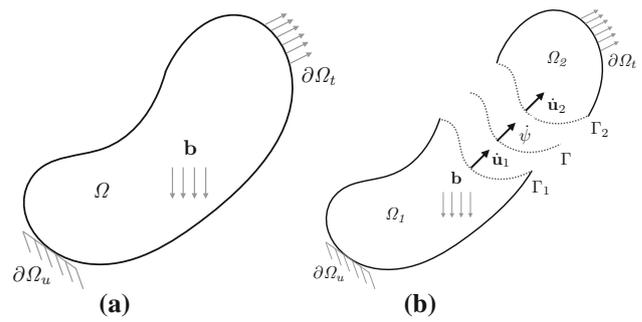
We organize our paper as follows. In Sect. 2, we present the formulation of the three-field dual-primal like domain decomposition approach and introduce the semi-discrete as well as fully-discrete forms. In Sect. 3, we discuss the interface problem of the PASTA-DDM scheme and propose it as a nested solve. The stability analysis of PASTA-DDM is presented in Sect. 4. Numerical verification studies and application to a real engineering problem are presented in Sect. 5. Finally, some conclusions are drawn in Sect. 6.

## 2 Mathematical formulation

Consider an elastic body occupying a bounded domain  $\Omega \subset \mathbb{R}^d$  ( $d = 1, 2, 3$ ) with boundary  $\partial\Omega$  satisfying the Lipschitz condition. Further, let  $\partial\Omega_u$  and  $\partial\Omega_t$ , denote a portion of the boundary where the displacement and surface traction are prescribed, and  $\partial\Omega_u \cap \partial\Omega_t = \emptyset$  (see Fig. 1a). The transient behavior of the elastic body during the time of interest  $t \in [0, T_f]$  is governed by the following equation of motion together with boundary and initial conditions:

$$\begin{aligned} \rho \ddot{\mathbf{u}}(\mathbf{x}, t) &= \nabla \cdot \boldsymbol{\sigma} + \mathbf{b} && \text{in } \Omega \quad \times [0, T_f], \\ \mathbf{u}(\mathbf{x}, t) &= \bar{\mathbf{u}} && \text{on } \partial\Omega_u \quad \times [0, T_f], \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= \bar{\mathbf{t}} && \text{on } \partial\Omega_t \quad \times [0, T_f], \\ \mathbf{u}(\mathbf{x}, 0) &= \mathbf{u}_0 && \text{in } \Omega, \\ \dot{\mathbf{u}}(\mathbf{x}, 0) &= \dot{\mathbf{u}}_0 && \text{in } \Omega, \end{aligned} \tag{1}$$

where  $\rho$  is the mass density,  $\boldsymbol{\sigma}$  is the stress tensor,  $\mathbf{u}$  is the displacement field,  $\mathbf{b}$  is the body force,  $\bar{\mathbf{u}}$  is the prescribed displacement on  $\partial\Omega_u$ ,  $\bar{\mathbf{t}}$  is the prescribed traction on  $\partial\Omega_t$ ,  $\mathbf{n}$  is



**Fig. 1** A diagram showing an arbitrary computational domain  $\Omega$  and its partitioning into two non-overlapping subdomains  $\Omega_1$  and  $\Omega_2$ . The local interfaces of  $\Omega_1$  and  $\Omega_2$  are denoted by  $\Gamma_1$  and  $\Gamma_2$ , while the global interface is denoted by  $\Gamma$ . The velocities on the boundaries  $\Gamma_1$ ,  $\Gamma_2$  and  $\Gamma$  are denoted by  $\dot{\mathbf{u}}_1$ ,  $\dot{\mathbf{u}}_2$  and  $\dot{\boldsymbol{\psi}}$ , respectively. **a** Spatial domain,  $\Omega$ . **b** Domain decomposition,  $\Omega = \Omega_1 \cup \Omega_2$

a unit normal to the surface, and  $\mathbf{u}_0$  and  $\dot{\mathbf{u}}_0$  are the initial displacement and velocity, respectively. For small deformation, the infinitesimal strain  $\boldsymbol{\epsilon}$  tensor is given by

$$\boldsymbol{\epsilon} = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T), \tag{2}$$

and for elastic behavior, the constitutive equation is expressed as

$$\boldsymbol{\sigma} = \mathbf{D} : \boldsymbol{\epsilon}, \tag{3}$$

where  $\mathbf{D}$  is the linear elasticity tensor. The Lagrangian of the system can be defined as

$$\mathcal{L}(\mathbf{u}, \dot{\mathbf{u}}) = T(\dot{\mathbf{u}}) - V(\mathbf{u}), \tag{4}$$

where the kinetic  $T(\dot{\mathbf{u}})$  and elastic potential  $V(\mathbf{u})$  energies are given by:

$$\begin{aligned} T(\dot{\mathbf{u}}) &= \frac{1}{2} \int_{\Omega} \rho \dot{\mathbf{u}} \cdot \dot{\mathbf{u}} \, d\Omega, \\ V(\mathbf{u}) &= \frac{1}{2} \int_{\Omega} \boldsymbol{\epsilon} : \mathbf{D} : \boldsymbol{\epsilon} \, d\Omega - \int_{\Omega} \mathbf{u} \cdot \mathbf{b} \, d\Omega - \int_{\partial\Omega_t} \mathbf{u} \cdot \bar{\mathbf{t}} \, d\Gamma. \end{aligned} \tag{5}$$

It is common for large-scale dynamical problems with complex localized features to solve the equation of motion, Eq. (1), on parallel computers using domain decomposition methods. In domain decomposition techniques, the physical domain  $\Omega$  is partitioned into  $n_s$  non-overlapping subdomains ( $\Omega_s, 1 \leq s \leq n_s$ ) (see Fig. 1b, e.g. for the case of two subdomains) such that [41,58,63]:

$$\Omega = \bigcup_{s=1}^{n_s} \Omega_s, \quad \Omega_s \cap \Omega_r = \emptyset, \tag{7}$$

$$\Gamma = \bigcup_{s=1}^{n_s} \Gamma_s, \quad \Gamma_s = \partial\Omega_s \setminus \partial\Omega. \tag{8}$$

To obtain the weak form, we transform the original problem, Eq. (1), into the following constrained minimization form. Find  $\mathbf{u}_s$  such that:

$$\begin{aligned} \mathcal{L}(\mathbf{u}_s, \dot{\mathbf{u}}_s) = & \sum_{s=1}^{n_s} \left( \frac{1}{2} \int_{\Omega_s} \rho_s \dot{\mathbf{u}}_s \cdot \dot{\mathbf{u}}_s \, d\Omega \right. \\ & - \frac{1}{2} \int_{\Omega_s} \boldsymbol{\varepsilon}_s : \mathbf{D} : \boldsymbol{\varepsilon}_s \, d\Omega + \int_{\Omega_s} \mathbf{u}_s \cdot \mathbf{b}_s \, d\Omega \\ & \left. + \int_{\partial\Omega_t} \mathbf{u}_s \cdot \bar{\mathbf{t}}_s \, d\Gamma \right) \rightarrow \min, \end{aligned} \tag{9}$$

subject to:

$$\int_{\Gamma} (\dot{\mathbf{u}}_s - \dot{\boldsymbol{\psi}}) \, d\Gamma = \mathbf{0}, \quad s = 1, \dots, n_s, \tag{10}$$

where  $\dot{\mathbf{u}}_s$  is the local subdomain velocity and  $\dot{\boldsymbol{\psi}}$  is the velocity of the common (frame) interface (see Fig. 1b).

Note that a continuity constraint in the aforementioned minimization problem can be imposed on the displacement, velocity or acceleration fields [7,10,14,20]. However, to guarantee energy conservation and for flexibility in coupling different types of time integration schemes, enforcing the continuity of the velocity across the interface seems most appropriate [10,14,15]. Thus, we impose a kinematic constraint on the velocity field among the portioned subdomains as defined in Eq. (10).

In the kinematic constraint, Eq. (10), the common interface variable  $\dot{\boldsymbol{\psi}}$  is global, while the subdomain interface variable  $\dot{\mathbf{u}}_s$  is local. To enforce such a constraint, we utilize local Lagrange multipliers [48,50,56]. The variable on the common interface is the primal kinematic variable (defined globally) and the local Lagrange multipliers are the dual variable (defined locally), and thus the subdomains do not share information directly, but only communicate through the common interface [48,50,56]. Accordingly, the augmented Lagrangian associated with the constrained minimization problem can be expressed as

$$\begin{aligned} \bar{\mathcal{L}}(\mathbf{u}_s, \dot{\mathbf{u}}_s, \boldsymbol{\lambda}_s, \dot{\boldsymbol{\psi}}) = & \sum_{s=1}^{n_s} \left( \frac{1}{2} \int_{\Omega_s} \rho_s \dot{\mathbf{u}}_s \cdot \dot{\mathbf{u}}_s \, d\Omega \right. \\ & - \frac{1}{2} \int_{\Omega_s} \boldsymbol{\varepsilon}_s : \mathbf{D} : \boldsymbol{\varepsilon}_s \, d\Omega + \int_{\Omega_s} \mathbf{u}_s \cdot \mathbf{b}_s \, d\Omega \\ & + \int_{\partial\Omega_t} \mathbf{u}_s \cdot \bar{\mathbf{t}}_s \, d\Gamma \\ & \left. + \int_{\Gamma} \boldsymbol{\lambda}_s \cdot (\dot{\mathbf{u}}_s - \dot{\boldsymbol{\psi}}) \, d\Gamma \right), \end{aligned} \tag{11}$$

where we introduce the local Lagrange multiplier  $\boldsymbol{\lambda}_s$  to enforce the kinematic constraint. Note that for the case of

a velocity constraint (Eq. 10), the local Lagrange multiplier,  $\boldsymbol{\lambda}_s$ , represents a generalized momentum [7].

The first variation of the augmented Lagrangian with respect to the field variables  $(\mathbf{u}_s, \dot{\mathbf{u}}_s, \boldsymbol{\lambda}_s, \dot{\boldsymbol{\psi}})$  gives

$$\begin{aligned} \delta \bar{\mathcal{L}} = & \sum_{s=1}^{n_s} \left( \int_{\Omega_s} \rho_s \delta \dot{\mathbf{u}}_s \cdot \dot{\mathbf{u}}_s \, d\Omega - \int_{\Omega_s} \delta \boldsymbol{\varepsilon}_s : \mathbf{D} : \boldsymbol{\varepsilon}_s \, d\Omega \right. \\ & + \int_{\Omega_s} \delta \mathbf{u}_s \cdot \mathbf{b}_s \, d\Omega + \int_{\partial\Omega_t} \delta \mathbf{u}_s \cdot \bar{\mathbf{t}}_s \, d\Gamma \\ & + \int_{\Gamma} \delta \boldsymbol{\lambda}_s \cdot (\dot{\mathbf{u}}_s - \dot{\boldsymbol{\psi}}) \, d\Gamma + \int_{\Gamma} \delta \dot{\mathbf{u}}_s \cdot \boldsymbol{\lambda}_s \, d\Gamma \\ & \left. - \int_{\Gamma} \delta \dot{\boldsymbol{\psi}} \cdot \boldsymbol{\lambda}_s \, d\Gamma \right). \end{aligned} \tag{12}$$

Hamilton’s principle requires that the action integral (the integral of the Lagrangian from initial to final times  $[0, T_f]$ ) is stationary for all possible paths:

$$\begin{aligned} \int_0^{T_f} \left( \sum_{s=1}^{n_s} \left( \int_{\Omega_s} \rho_s \delta \dot{\mathbf{u}}_s \cdot \dot{\mathbf{u}}_s \, d\Omega + \int_{\Gamma} \delta \dot{\mathbf{u}}_s \cdot \boldsymbol{\lambda}_s \, d\Gamma \right. \right. \\ \left. \left. - \int_{\Gamma} \delta \dot{\boldsymbol{\psi}} \cdot \boldsymbol{\lambda}_s \, d\Gamma + \int_{\Omega_s} \delta \mathbf{u}_s \cdot \mathbf{b}_s \, d\Omega \right. \right. \\ \left. \left. - \int_{\Omega_s} \delta \boldsymbol{\varepsilon}_s : \mathbf{D} : \boldsymbol{\varepsilon}_s \, d\Omega + \int_{\partial\Omega_t} \delta \mathbf{u}_s \cdot \bar{\mathbf{t}}_s \, d\Gamma \right. \right. \\ \left. \left. + \int_{\Gamma} \delta \boldsymbol{\lambda}_s \cdot (\dot{\mathbf{u}}_s - \dot{\boldsymbol{\psi}}) \, d\Gamma \right) \right) dt = 0. \end{aligned} \tag{13}$$

Next, we integrate by parts the first three terms in Eq. (13) and apply the boundary conditions  $\delta \mathbf{u}(0) = \delta \boldsymbol{\psi}(0) = \delta \mathbf{u}(T_f) = \delta \boldsymbol{\psi}(T_f) = \mathbf{0}$ , which leads to the following three equations for a typical subdomain  $\Omega_s$ .

The constrained equation of motion:

$$\begin{aligned} \int_{\Omega_s} \rho_s \ddot{\mathbf{u}}_s \cdot \delta \mathbf{u}_s \, d\Omega + \int_{\Omega_s} \boldsymbol{\sigma}_s : \delta \boldsymbol{\varepsilon}_s \, d\Omega + \int_{\Gamma_s} \dot{\boldsymbol{\lambda}}_s \cdot \delta \mathbf{u}_s \, d\Gamma \\ = \int_{\Omega_s} \delta \mathbf{u}_s \cdot \mathbf{b}_s \, d\Omega + \int_{\partial\Omega_t} \delta \mathbf{u}_s \cdot \bar{\mathbf{t}}_s \, d\Gamma \end{aligned} \tag{14}$$

The kinematic constraint coupling all the subdomains:

$$\int_{\Gamma} \delta \boldsymbol{\lambda}_s \cdot (\dot{\mathbf{u}}_s - \dot{\boldsymbol{\psi}}) \, d\Gamma = 0. \tag{15}$$

The equilibrium equation among the subdomains:

$$\sum_{s=1}^{n_s} - \int_{\Gamma} \delta \dot{\boldsymbol{\psi}} \cdot \dot{\boldsymbol{\lambda}}_s \, d\Gamma = 0. \tag{16}$$

### 2.1 Spatial discretization

In this subsection, we seek an approximate solution to the weak form in the spatial dimension using the finite element

method. Thus, let each subdomain  $\Omega_s$  be discretized by a cell with a maximum element size  $h$ , and let  $\mathcal{X}_h, \mathcal{Y}_h$  and  $\mathcal{Z}_h$  be the finite element subspaces for the displacement field, Lagrange multipliers and the common interface variable defined as

$$\mathcal{X}_h = \text{span}\{\mathbf{N}_u^1(\mathbf{x}), \mathbf{N}_u^2(\mathbf{x}), \dots, \mathbf{N}_u^{n_u}(\mathbf{x})\}, \tag{17}$$

$$\mathcal{Y}_h = \text{span}\{\mathbf{N}_\lambda^1(\mathbf{x}), \mathbf{N}_\lambda^2(\mathbf{x}), \dots, \mathbf{N}_\lambda^{n_\lambda}(\mathbf{x})\}, \tag{18}$$

$$\mathcal{Z}_h = \text{span}\{\mathbf{N}_\psi^1(\mathbf{x}), \mathbf{N}_\psi^2(\mathbf{x}), \dots, \mathbf{N}_\psi^{n_\psi}(\mathbf{x})\}, \tag{19}$$

where  $\{\mathbf{N}_u^i(\mathbf{x})\}_{i=1}^{n_u}, \{\mathbf{N}_\lambda^i(\mathbf{x})\}_{i=1}^{n_\lambda}$  and  $\{\mathbf{N}_\psi^i(\mathbf{x})\}_{i=1}^{n_\psi}$  are piecewise linear finite element basis functions. Then the finite element approximate solutions  $\mathbf{u}_s^h \in \mathcal{X}_h, \lambda_s^h \in \mathcal{Y}_h$  and  $\psi_s^h \in \mathcal{Z}_h$  can be written as

$$\begin{aligned} \mathbf{u}_s^h &= \sum_{i=1}^{n_u} \mathbf{N}_u^i \tilde{\mathbf{u}}_s^i(t), & \lambda_s^h &= \sum_{i=1}^{n_\lambda} \mathbf{N}_\lambda^i \tilde{\lambda}_s^i(t), \\ \psi_s^h &= \sum_{i=1}^{n_\psi} \mathbf{N}_\psi^i \tilde{\psi}_s^i(t). \end{aligned} \tag{20}$$

For stability analysis of the finite element spaces and optimal error estimates for the three-field formulation, the multiscale mortar finite elements, and the discontinuous stabilized mortar formulation, in the context of domain decomposition, we refer to [2, 8, 23].

Substituting the discrete representations of the field variables, Eq. (20), in the weak form, Eqs. (14), (15) and (16), leads to the following semi-discretized dual-primal linear system

$$\begin{aligned} \mathbf{M}_s \ddot{\mathbf{u}}_s(t) + \mathbf{K}_s \mathbf{u}_s(t) + \mathbf{E}_s^T \dot{\lambda}_s(t) &= \mathbf{f}_s(t), \\ \mathbf{E}_s \dot{\mathbf{u}}_s(t) - \mathbf{B}_s \dot{\psi}(t) &= \mathbf{0}, \\ \sum_{s=1}^{n_s} -\mathbf{B}_s^T \dot{\lambda}_s(t) &= \mathbf{0}, \end{aligned} \tag{21}$$

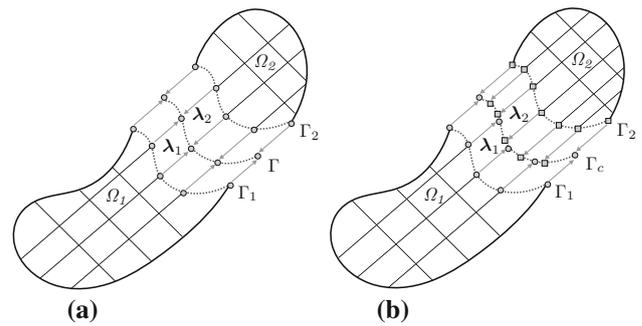
where we drop the nodal finite element association marking (tilde) for brevity of the representation. Here,  $\mathbf{M}_s$  and  $\mathbf{K}_s$  are the traditional subdomain mass and stiffness matrices, respectively. The subdomain vector  $\mathbf{f}_s(t)$  represents the prescribed forces. The mass like projection matrices  $\mathbf{E}_s$  and  $\mathbf{B}_s$  are defined as

$$\mathbf{E}_s = \int_{\Gamma_s} \mathbf{N}_\lambda^T \mathbf{N}_u \, d\Gamma, \tag{22}$$

$$\mathbf{B}_s = \int_{\Gamma} \mathbf{N}_\lambda^T \mathbf{N}_\psi \, d\Gamma. \tag{23}$$

### 2.1.1 Non-matching grids

In principle, each subdomain can be discretized in space independently using a suitable mesh size based on the multi-scale features of the problem. However, this local treatment



**Fig. 2** Matching and non-matching subdomains meshes at the interface.  $\Gamma_c$  is the common-refinement-based interface for data transfer between non-matching spatial discretizations. **a** Matching subdomains meshes. **b** Non-matching subdomains meshes

of the spatial resolution can lead to non-matching meshes along the interface  $\Gamma$ . For optimal accuracy and to fulfill the conservation requirement of physical quantities, a common-refinement-based interface [29, 30] is utilized for non-matching meshes. The common-refinement-based interface denoted by  $\Gamma_c$  is a surface (in 3D) or a line (in 2D) that provides a discretization for the interface such that the  $L_2$  norm of the error between the source and target field is minimized. Details of an efficient and robust algorithm for constructing a common-refinement-based interface are provided in the work of Jiao and Heath [31, 32]. Thus, this approach leads to an accurate integration of the interface mass-like coupling matrix (see Fig. 2b). The local Lagrange multipliers are discretized on a low dimensional mesh that is a restriction of the subdomain discretization on the local interface boundary (i.e.,  $\mathbf{N}_\lambda(\mathbf{x}) = \mathbf{N}_u(\mathbf{x}), \forall \mathbf{x} \in \Gamma_s$ ). It follows that Eq. (22) and Eq. (23) become:

$$\mathbf{E}_s = \int_{\Gamma_s} \mathbf{N}_u^T \mathbf{N}_u \, d\Gamma, \tag{24}$$

$$\mathbf{B}_s = \int_{\Gamma_c} \mathbf{N}_u^T \mathbf{N}_\psi \, d\Gamma. \tag{25}$$

It is worth mentioning that the zero-moment rule is another approach available in the literature that can be used to provide a discretization of the common interface [45, 56]. A review of different techniques for data transfer between non-matching meshes is provided in [11].

As discussed briefly in the previous section, the selection of the finite element spaces, Eqs. (17)–(19), is crucial especially for elliptic problems. For transient analysis, however, the problem is regularized by the positive definite mass matrix. Nevertheless, the ratio of continuity over coercivity (i.e. ellipticity) constants may be very large (for example, when using a small time step). In such a case, better regularization needs to be considered.

In the context of a domain decomposition with matching grids, it has been shown that the continuity condition, Eq. (21-

b), is not required to be such that the discrete hybrid system of Eqs. (21-a) and (21-c) satisfies the Ladyzhenskaya-Babuška-Brezzi (*inf-sup*) condition [38]. In the case of matching meshes the discrete trace operators,  $\mathbf{E}_s = \mathbf{B}_s$ , can be interpreted as restriction matrices [38]. This is not the case if different trace operators are used as in Eq. (24) and Eq. (25), and we provide the *inf-sup* analysis of our scheme in the elliptic setting in Appendix 1.

### 2.2 Temporal discretization

To obtain the fully discrete dual-primal transition linear system, a time discretization scheme is required. Thus, in the following subsections, we address the time discretization of the dual primal system (see Eq. (21)).

#### 2.2.1 Synchronous time integration

For time discretization with a unified time step for all subdomains (i.e. synchronous time integration), let the integration time of interest  $[T_0, T_f]$  be divided into  $n_f$  time steps. Next, we define the synchronization time step (the global time step)  $\Delta T$  as

$$\Delta T = \frac{T_f - T_0}{n_f}. \tag{26}$$

Consequently, the fully discrete representation of the dual primal system, Eq. (21), at a discrete instant of time  $n + 1$  can be expressed as

$$\begin{aligned} \mathbf{M}_s \ddot{\mathbf{u}}_s^{n+1} + \mathbf{K}_s \mathbf{u}_s^{n+1} + \mathbf{E}_s^T \boldsymbol{\Lambda}_s^{n+1} &= \mathbf{f}_s^{n+1}, \\ \mathbf{E}_s \dot{\mathbf{u}}_s^{n+1} - \mathbf{B}_s \boldsymbol{\Psi}^{n+1} &= \mathbf{0}, \quad n = 0, 1, \dots, n_f - 1 \\ \sum_{s=1}^{n_s} -\mathbf{B}_s^T \boldsymbol{\Lambda}_s^{n+1} &= \mathbf{0}, \end{aligned} \tag{27}$$

where for simplicity of notation, we denote  $\boldsymbol{\Lambda}_s(t) = \dot{\boldsymbol{\lambda}}_s(t)$  and  $\boldsymbol{\Psi} = \dot{\boldsymbol{\psi}}(t)$ .

To advance the system one time step, we use the Newmark algorithm (potentially with local integration parameters  $\gamma_s$  and  $\beta_s$ ), that is

$$\dot{\mathbf{u}}_s^{n+1} = \dot{\mathbf{u}}_s^n + (1 - \gamma) \Delta T \ddot{\mathbf{u}}_s^n + \gamma \Delta T \ddot{\mathbf{u}}_s^{n+1}, \tag{28}$$

$$\mathbf{u}_s^{n+1} = \mathbf{u}_s^n + \Delta T \dot{\mathbf{u}}_s^n + \left(\frac{1}{2} - \beta\right) \Delta T^2 \ddot{\mathbf{u}}_s^n + \beta \Delta T^2 \ddot{\mathbf{u}}_s^{n+1}. \tag{29}$$

At this stage, it is worth emphasizing that other time integration schemes (e.g. geometric integrators [33]) can be used within the PASTA-DDM framework provided that the subdomains local stability criteria are satisfied.

Combining the fully discrete dual primal system, Eq. (27), with the Newmark scheme, Eqs. (28) and (29), gives the following transition linear problem

$$\begin{bmatrix} \mathbf{A}_s & \mathbf{R}_s^T & \mathbf{0} \\ \mathbf{L}_s & \mathbf{0} & -\mathbf{B}_s \\ \mathbf{0} & \sum_{s=1}^{n_s} -\mathbf{B}_s^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{U}_s \\ \boldsymbol{\Lambda}_s \\ \boldsymbol{\Psi} \end{Bmatrix}^{n+1} = \begin{Bmatrix} \mathbf{F}_s^{n+1} - \mathbf{C}_s \mathbf{U}_s^n \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix}, \tag{30}$$

where for compact representation, we define

$$\begin{aligned} \mathbf{A}_s &= \begin{bmatrix} \mathbf{M}_s & \mathbf{0} & \mathbf{K}_s \\ -\gamma \Delta T \mathbf{I} & \mathbf{I} & \mathbf{0} \\ -\beta \Delta T^2 \mathbf{I} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \\ \mathbf{C}_s &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -(1 - \gamma) \Delta T \mathbf{I} & -\mathbf{I} & \mathbf{0} \\ -\left(\frac{1}{2} - \beta\right) \Delta T^2 \mathbf{I} & -\Delta T \mathbf{I} & -\mathbf{I} \end{bmatrix}, \\ \mathbf{U}_s &= \begin{Bmatrix} \ddot{\mathbf{u}}_s \\ \dot{\mathbf{u}}_s \\ \mathbf{u}_s \end{Bmatrix}, \quad \mathbf{F}_s = \begin{Bmatrix} \mathbf{f}_s \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix}, \\ \mathbf{R}_s^T &= \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \otimes \mathbf{E}_s^T, \quad \mathbf{L}_s = [\mathbf{0} \ \mathbf{I} \ \mathbf{0}] \otimes \mathbf{E}_s. \end{aligned}$$

Here  $\mathbf{I}$  is the identity matrix.

#### 2.2.2 Asynchronous time integration

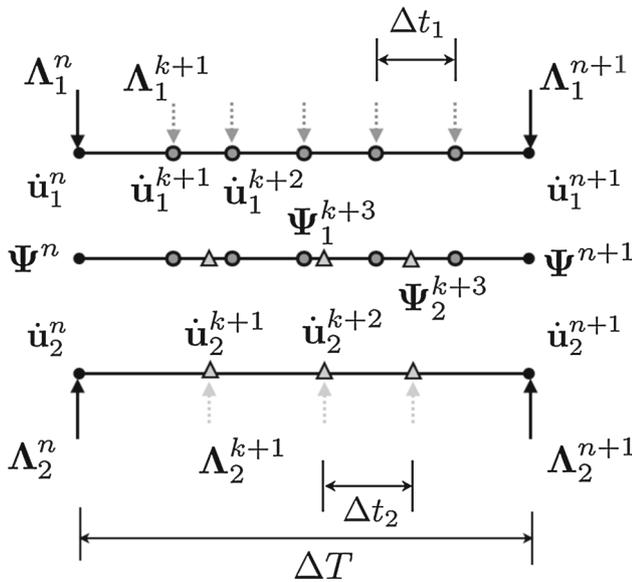
For time discretization using different time steps in each subdomain (i.e. multi-time stepping, asynchronous time integration), we extend the idea of spatial common-refinement-based interface into the temporal dimension. In particular, we define a temporal common refinement as a union of the two different time scales of adjacent subdomains (see Fig. 3). The common time scale serves as a bridge that links the asynchronous time steps of the subdomains.

Thus for asynchronous time discretization, let the global time step  $\Delta T$  be decomposed into multiple sub steps as

$$\Delta t_s = \frac{\Delta T}{n_{t_s}}, \tag{31}$$

where  $n_{t_s}$  is the local number of time steps for subdomain  $\Omega_s$ . The system time step  $\Delta T$  can be viewed as a synchronization time step whereby all the subdomains reach this directly, but with different number of time steps.

Next, we employ the Newmark scheme, Eqs. (28) and (29), in terms of local integration parameters ( $\gamma_s$  and  $\beta_s$ ) and subdomain time increment  $\Delta t_s$  as



**Fig. 3** Temporal common refinement time integration scheme. Here  $\Delta T$  represents the synchronization time step, while  $\Delta t_1$  and  $\Delta t_2$  are the subdomain time increments. The superscript  $k$  and  $n$  indicate the subdomain ( $\Omega_1, \Omega_2$ ) and the system (synchronization) time step indices, respectively

$$\dot{\mathbf{u}}_s^{k+1} = \dot{\mathbf{u}}_s^k + (1 - \gamma_s)\Delta t_s \ddot{\mathbf{u}}_s^k + \gamma_s \Delta t_s \ddot{\mathbf{u}}_s^{k+1}, \tag{32}$$

$$\mathbf{u}_s^{k+1} = \mathbf{u}_s^k + \Delta t_s \dot{\mathbf{u}}_s^k + \left(\frac{1}{2} - \beta_s\right) \Delta t_s^2 \ddot{\mathbf{u}}_s^k + \beta_s \Delta t_s^2 \ddot{\mathbf{u}}_s^{k+1}, \tag{33}$$

where  $k$  is the local time stepping index for a typical subdomain  $\Omega_s$ .

**Proposition 1** For the local Lagrange multipliers, we use a generalized  $\alpha$  like method for the temporal discretization

$$\Lambda_s^k = S^k \Lambda_s^n + T^k \Lambda_s^{n+1}, \tag{34}$$

where  $\Lambda_s^n$  and  $\Lambda_s^{n+1}$  are the Lagrange multipliers at the synchronization time step, and the linear functions  $S^k$  and  $T^k$  are given as

$$S^k = 1 - \frac{t_k}{\Delta T} = 1 - \frac{k}{n_{t_s}}, \tag{35}$$

$$T^k = \frac{t_k}{\Delta T} = \frac{k}{n_{t_s}}, \tag{36}$$

since  $t_k = k\Delta t_s$  and  $\Delta T = n_{t_s}\Delta t_s$ . Note that  $S^k + T^k = 1$ .

Proposition (1) is formulated based on the fact that the Lagrange multipliers can be viewed as lumped quantities at the beginning and end of one synchronization time step [51]. Furthermore, we notice that for synchronous time integration, the Lagrange multiplier varies linearly over one synchronization time step.

Combining the local time discretization scheme, Eqs. (32), (33) and (34), and the constrained discrete dual-primal equations, Eq. (27), gives the following transition linear system

$$\mathbf{A}_s \mathbf{U}_s^{k+1} + T^{k+1} \mathbf{R}_s^T \Lambda_s^{n+1} = \mathbf{F}_s^{k+1} - \mathbf{C}_s \mathbf{U}_s^k - S^{k+1} \mathbf{R}_s^T \Lambda_s^n, \tag{37}$$

$$\mathbf{L}_s \mathbf{U}_s^{k+1} - \mathbf{E}_s \Psi_s^{k+1} = \mathbf{0}, \quad \text{if } k \neq n_{t_s} - 1 \tag{38}$$

$$\mathbf{L}_s \mathbf{U}_s^{k+1} - \mathbf{B}_s \Psi^{n+1} = \mathbf{0}, \quad \text{if } k = n_{t_s} - 1 \tag{39}$$

$$\sum_{s=1}^{n_s} -\mathbf{B}_s^T \Lambda_s^{n+1} = \mathbf{0}, \tag{40}$$

where we redefine the amplification and the right-hand matrices in terms of the local parameters  $\gamma_s, \beta_s$  and  $\Delta t_s$  as

$$\mathbf{A}_s = \begin{bmatrix} \mathbf{M}_s & \mathbf{0} & \mathbf{K}_s \\ -\gamma_s \Delta t_s \mathbf{I} & \mathbf{I} & \mathbf{0} \\ -\beta_s \Delta t_s^2 \mathbf{I} & \mathbf{0} & \mathbf{I} \end{bmatrix},$$

$$\mathbf{C}_s = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -(1 - \gamma_s) \Delta t_s \mathbf{I} & -\mathbf{I} & \mathbf{0} \\ -\left(\frac{1}{2} - \beta_s\right) \Delta t_s^2 \mathbf{I} & -\Delta t_s \mathbf{I} & -\mathbf{I} \end{bmatrix}.$$

Note that  $\Psi_s^{k+1}$  represents an intermediate frame velocity, while  $\Psi^{n+1}$  is the synchronization frame velocity shared among the subdomains (see Fig. 3). The intermediate frame velocity,  $\Psi_s^{k+1}$ , is local and defined at the subdomain time increments. Since the intermediate frame velocity is local, we define a local constraint in Eq. (38) that can be viewed as a mechanism to transfer information between the local time step and the global synchronization step. On the other hand, the global frame velocity  $\Psi^{n+1}$  is defined at the synchronization time step, where the spatial common-refinement-based interface is described. At the synchronization time step, a global constraint Eq. (39) is introduced to transfer information among the subdomains. In other works [7, 10, 20, 21, 51], the intermediate frame velocity,  $\Psi_s^{k+1}$ , is interpolated between the coarse and fine timescales which may lead to undesirable energy conservation properties.

*Remark 1* The fully discrete dual-primal transition linear system, Eqs. (37) to (40), is full rank and consistent, and thus it has a unique solution.

As an illustrative example, for the case of two subdomains with time steps  $n_{t_1} = 2$  and  $n_{t_2} = 3$ , the global dual-primal transition system can be written as



$$\begin{bmatrix} \mathcal{A}_1 & & & \mathcal{B}_1 \\ & \mathcal{A}_2 & & \mathcal{B}_2 \\ & & \ddots & \vdots \\ & & & \mathcal{A}_{n_s} & \mathcal{B}_{n_s} \\ \mathcal{C}_1^T & \mathcal{C}_2^T & \dots & \mathcal{C}_{n_s}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathcal{U}_1 \\ \mathcal{U}_2 \\ \vdots \\ \mathcal{U}_{n_s} \\ \Psi \end{bmatrix}^{n+1} = \begin{bmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \\ \vdots \\ \mathcal{F}_{n_s} \\ \mathbf{0} \end{bmatrix}^n. \quad (47)$$

Clearly, the linear system in Eq. (47) is highly parallelizable. Once the global interface variable  $\Psi$  is available, the subdomain unknowns  $\mathcal{U}_s$  can be obtained independently and hence in parallel. Thus, since  $\mathcal{A}_s^{-1}$  is invertable, a concurrent block Gaussian elimination reduces the system in Eq. (47) to the following interface problem for the global variable  $\Psi$

$$\mathcal{A}_\Psi \Psi = \mathcal{F}_\Psi, \quad (48)$$

where the condensed interface matrix  $\mathcal{A}_\Psi$  and the corresponding right-hand side  $\mathcal{F}_\Psi$  are defined as

$$\mathcal{A}_\Psi = \sum_{s=1}^{n_s} \mathcal{C}_s^T \mathcal{A}_s^{-1} \mathcal{B}_s, \quad (49)$$

$$\mathcal{F}_\Psi = \sum_{s=1}^{n_s} \mathcal{C}_s^T \mathcal{A}_s^{-1} \mathcal{F}_s. \quad (50)$$

Having the interface variable  $\Psi$ , the interior unknowns  $\mathcal{U}_s$  can be obtained concurrently by solving the following local problem on each subdomain

$$\mathcal{U}_s = \mathcal{A}_s^{-1}(\mathcal{F}_s - \mathcal{B}_s \Psi). \quad (51)$$

The primary task at hand in our PASTA-DDM is to solve the interface problem Eq. (48). To simplify the condensed interface matrix  $\mathcal{A}_\Psi$ , we notice that for a typical subdomain, the local contribution  $\mathcal{C}_s^T \mathcal{A}_s^{-1} \mathcal{B}_s$  to  $\mathcal{A}_\Psi$  can be written as

$$\mathcal{C}_s^T \mathcal{A}_s^{-1} \mathcal{B}_s = \begin{bmatrix} \mathbf{0} & \mathcal{C}_s^T \end{bmatrix} \begin{bmatrix} \mathbf{I}_{n_s} \otimes \mathbf{A}_s + \mathbf{I}_{n_s-1} \otimes \mathbf{C}_s & \mathbf{I}_c \otimes \mathbb{R}_s^T \\ \mathbf{I}_{n_s} \otimes \mathbf{L}_s & \mathbf{I}_{n_s+1} \otimes \mathbf{E}_s \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbb{B}_s \end{bmatrix}, \quad (52)$$

where we define the following generalized permutation matrices,

$$\mathbf{I}_{n_s} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \in \mathcal{R}^{n_s \times n_s},$$

$$\mathbf{I}_{n_s+1} = \begin{bmatrix} 0 & 1 & & \\ & 0 & 1 & \\ & & \ddots & \ddots \\ & & & 0 \end{bmatrix} \in \mathcal{R}^{n_s \times n_s}, \quad (53)$$

$$\mathbf{I}_{n_s-1} = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & & 1 & 0 \end{bmatrix} \in \mathcal{R}^{n_s \times n_s},$$

$$\mathbf{I}_c = \begin{bmatrix} \frac{1}{n_s} \\ \frac{2}{n_s} \\ \vdots \\ \frac{n_s}{n_s} \end{bmatrix} \in \mathcal{R}^{n_s \times 1}, \quad (54)$$

and restriction matrices

$$\mathbb{R}_s^T = [\mathbf{R}_s^T \ \mathbf{0}]. \quad (55)$$

Using the Schur complement approach, Eq. (52) can be factorized as

$$\mathcal{C}_s^T \mathcal{A}_s^{-1} \mathcal{B}_s = \mathcal{C}_s^T [(\mathbf{I}_{n_s+1} \otimes \mathbf{E}_s) - (\mathbf{I}_{n_s} \otimes \mathbf{L}_s) [\mathbf{I}_{n_s} \otimes \mathbf{A}_s + \mathbf{I}_{n_s-1} \otimes \mathbf{C}_s]^{-1} (\mathbf{I}_c \otimes \mathbb{R}_s^T)] \mathbb{B}_s. \quad (56)$$

We can show that for a linear problem the term  $[\mathbf{I}_{n_s} \otimes \mathbf{A}_s + \mathbf{I}_{n_s-1} \otimes \mathbf{C}_s]^{-1}$  can be calculated using one factorization of the subdomain amplification matrix  $\mathbf{A}_s$  as

$$[\mathbf{I}_{n_s} \otimes \mathbf{A}_s + \mathbf{I}_{n_s-1} \otimes \mathbf{C}_s]^{-1} = \sum_{i=0}^{n_s-1} (-1)^i \mathbf{I}_{n_s-i} \otimes \mathbf{A}_s^{-1} (\mathbf{B}_s \mathbf{A}_s^{-1})^i, \quad (57)$$

where  $(\mathbf{B}_s \mathbf{A}_s^{-1})^i$  represents a regressive solve.

After substituting Eq. (57) back into Eq. (56) and manipulating the terms, we obtain

$$\mathcal{C}_s^T \mathcal{A}_s^{-1} \mathcal{B}_s = \mathcal{C}_s^T \left[ \mathbf{I}_{n_s+1} \otimes \mathbf{E}_s - \sum_{i=0}^{n_s-1} (-1)^i \mathbf{I}_{n_s-i} \mathbf{I}_c \otimes \mathbf{L}_s \mathbf{A}_s^{-1} (\mathbf{B}_s \mathbf{A}_s^{-1})^i \mathbb{R}_s^T \right] \mathbb{B}_s, \quad (58)$$

$$= \mathcal{C}_s^T \mathcal{A}_\Psi^s \mathbb{B}_s.$$

Similarly, we can obtain the subdomain condensed right-hand side as

$$\mathcal{C}_s^T \mathcal{A}_s^{-1} \mathcal{F}_s = \mathcal{C}_s^T \left[ \mathbf{I}_{n_s+1} \otimes \mathbf{E}_s - \sum_{i=0}^{n_s-1} (-1)^i \mathbf{I}_{n_s-i} \mathbf{I}_c \otimes \mathbf{L}_s \mathbf{A}_s^{-1} (\mathbf{B}_s \mathbf{A}_s^{-1})^i \mathbb{R}_s^T \right] \mathcal{F}_s, \quad (59)$$

$$= \mathcal{C}_s^T \mathcal{A}_\Psi^s \mathcal{F}_s,$$

where the local condensed interface matrix is defined as

$$\mathcal{A}_\Psi^s = \left[ \begin{array}{c} \mathbf{I}_{n_{t_s}+1} \otimes \mathbf{E}_s - \sum_{i=0}^{n_{t_s}-1} (-1)^i \mathbf{I}_{n_{t_s}-i} \mathbf{I}_c \\ \otimes \mathbf{L}_s \mathbf{A}_s^{-1} (\mathbf{B}_s \mathbf{A}_s^{-1})^i \mathbb{R}_s^T \end{array} \right]. \quad (60)$$

Finally, the condensed interface problem for the global variable  $\Psi$ , Eq. (48), becomes:

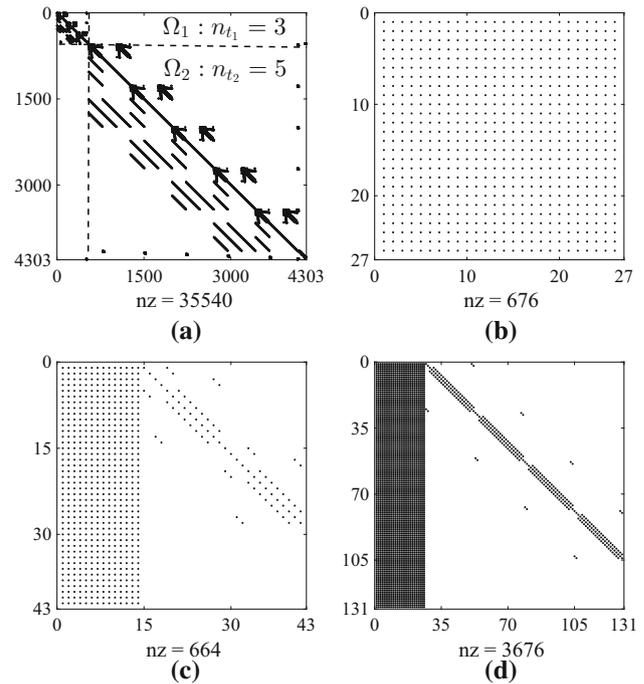
$$\sum_{s=1}^{n_s} \mathbb{C}_s^T \mathcal{A}_\Psi^s \mathbb{B}_s \Psi = \sum_{s=1}^{n_s} \mathbb{C}_s^T \mathcal{A}_\Psi^s \mathcal{F}_s. \quad (61)$$

*Remark 2* The condensed interface problem, Eq. (61), can be solved either by using factorization of the subdomain amplification matrix or more efficiently by a recycling Krylov subspace method for a sequence of linear systems with multiple right-hand sides.

For a particular case of two subdomains with non-matching grids and asynchronous time stepping, we show in Figs. 4 and 5 the sparsity structure and spectrum of the eigenvalues of the global system, Eq. (47), the condensed interface system, Eq. (61), and the local subdomain contributions, Eq. (60). Clearly, the condensed interface system is dense, but much smaller in size and better conditioned than its global counterpart.

Note that in practice, the condensed interface primal system, Eq. (61), itself does not need to be constructed explicitly, and only a procedure for matrix-vector product is required. Specifically, a preconditioned Krylov subspace iterative method is often used to solve the interface problem. At each iteration of the iterative solver loop, only the action of the local condensed interface matrix, Eq. (60), on a vector is needed. This matrix-vector product can be obtained in parallel by solving local problems on each subdomain and gathering the resulting vectors. Moreover, the rate of convergence of the iterative method can be generally improved by a suitable preconditioner [41, 53, 58, 63]. Devising a two-level scalable preconditioner that can exploit the block structure of the subdomain transition system and nested solve is essential for efficient parallel implementation of PASTA-DDM. For example, the Balancing Domain Decomposition by Constraints (BDDC) technique [40, 61] can be considered in this aspect. Since the primary goal of this article is formulation of the asynchronous space and time algorithm, at this stage we construct the interface problem, Eq. (61), and solve it using a direct solver.

Considering the mathematical structure of the solver, (Eqs. (48) to (51)), and the vast amount of parallel implementation strategies [8, 38, 41, 53, 63], we note the parallel nature of this scheme. Moreover, it is worth mentioning that



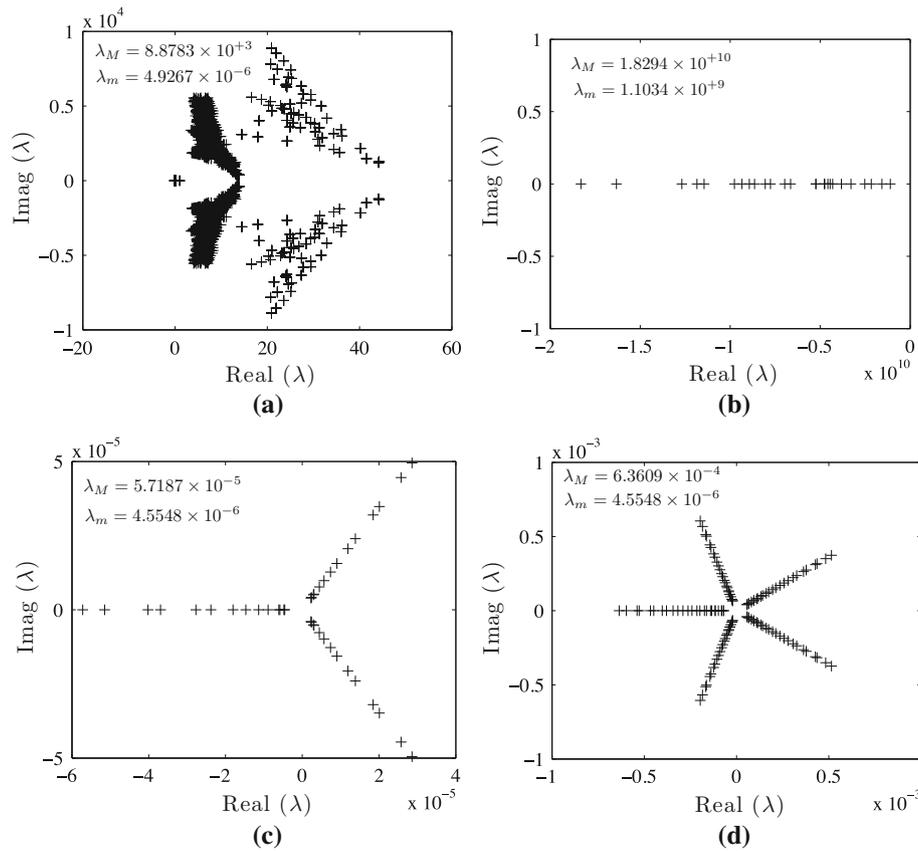
**Fig. 4** The sparsity structure of (a) the global dual-primal linear system given in Eq. (47), (b) the condensed interface primal system defined in Eq. (61), and (c), (d) the local contribution Eq. (60) for the case of two subdomains with non-matching grids ( $h_1 = 0.1768$  m and  $h_2 = 0.0884$  m) and asynchronous time stepping ( $\Delta t_1 : \Delta t_2 = 3:5$ ). The geometry and physical properties used are from the first example in Sect. 5.1. The variable  $nz$  represents the number of non-zero entries. **a** The global system, Eq. (47). **b** The condensed interface system, Eq. (61). **c** The local contribution from  $\Omega_1$ , Eq. (60). **d** The local contribution from  $\Omega_2$ , Eq. (60)

we implemented our PASTA-DDM algorithm in a virtual parallel environment. That is, all the subdomains construct their local mass, stiffness, amplification and restriction matrices independently. Furthermore, the intermediate temporal states of the subdomains are computed independently after solving the interface problem.

Finally, it is noteworthy that PASTA-DDM offers two-levels of parallelization (see Fig. 4). A global coarse grain parallelism due to the spatial decomposition and a fine grain time-based parallelism within each subdomain. The local fine grain parallelism can be achieved efficiently utilizing the *Parareal* algorithm [19, 37], for example. The synchronization time step in PASTA-DDM can be viewed as the coarse time grid correction in the *Parareal* algorithm.

### 4 Stability analysis of PASTA-DDM

In this section, the stability analysis of PASTA-DDM based on the energy method is presented [10, 25, 51]. Specifically, we prove that for the linear problem considered in this work, PASTA-DDM is an unconditionally stable scheme.



**Fig. 5** The spectrum of the eigenvalues of (a) the global dual-primal linear system given in Eq. (47), (b) the condensed interface primal interface system defined in Eq. (61) and (c), (d) the local contribution, Eq. (60), for the case of two subdomains with non-matching grids ( $h_1 = 0.1768$  m and  $h_2 = 0.0884$  m) and asynchronous time stepping ( $\Delta t_1:\Delta t_2 = 3:5$ ). The geometry and physical properties used are from

the first example in Sect. 5.1. In this figure, the variables  $\lambda_M$  and  $\lambda_m$  are the maximum and minimum eigenvalues, respectively. **a** The global system, Eq. (47). **b** The condensed interface system, Eq. (61). **c** The local contribution from  $\Omega_1$ , Eq. (60). **d** The local contribution from  $\Omega_2$ , Eq. (60)

**Lemma 1** *The increment of the discrete mechanical energy of a dynamical system (which is decomposed into  $n_s$  subdomains each of which has  $n_{t_s}$  time steps and without external force) over the time interval from  $t_n$  to  $t_{n+1}$  can be expressed as*

$$\begin{aligned} & \left[ \frac{1}{2} \ddot{\mathbf{u}}^T \mathbf{Q} \ddot{\mathbf{u}} + \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{K} \dot{\mathbf{u}} \right]^n \\ &= - \sum_{s=1}^{n_s} \sum_{k=1}^{n_{t_s}} \left( \gamma_s - \frac{1}{2} \right) \left[ \ddot{\mathbf{u}}_s^T \right]^k \mathbf{Q}_s \left[ \ddot{\mathbf{u}}_s \right]^k \\ & \quad - \sum_{s=1}^{n_s} \sum_{k=1}^{n_{t_s}} \frac{1}{\Delta t_s} \left[ \dot{\mathbf{u}}_s^T \right]^k \mathbf{E}_s^T \left[ \Lambda_s \right]^k. \end{aligned} \tag{62}$$

Here we denote

$$\mathbf{Q}_s = \mathbf{M}_s + \Delta t_s^2 (\beta_s - \frac{1}{2} \gamma_s) \mathbf{K}_s, \tag{63}$$

and the mean  $\langle \bullet \rangle$  and difference  $\llbracket \bullet \rrbracket$  operators are defined as

$$\begin{aligned} \langle \bullet \rangle &= \frac{1}{2} (\bullet^{k+1} + \bullet^k), \\ \llbracket \bullet \rrbracket^k &= (\bullet^{k+1} - \bullet^k), \quad \llbracket \bullet \rrbracket^n = (\bullet^{n+1} - \bullet^n), \end{aligned} \tag{64}$$

where  $k$  is the local and  $n$  is the global (synchronization) time step (see Fig. 3).

*Proof* For a typical subdomain  $\Omega_s$ , without external force, the equation of motion in terms of the difference operator can be written as

$$\mathbf{M}_s \llbracket \ddot{\mathbf{u}}_s \rrbracket^k + \mathbf{K}_s \llbracket \mathbf{u}_s \rrbracket^k + \mathbf{E}_s^T \llbracket \Lambda_s \rrbracket^k = \mathbf{0}. \tag{65}$$

Note that Newmark’s equation in terms of the mean and difference operators can also be expressed as

$$\llbracket \dot{\mathbf{u}}_s \rrbracket^k = \Delta t_s \langle \ddot{\mathbf{u}}_s \rangle + \left( \gamma_s - \frac{1}{2} \right) \Delta t_s \llbracket \ddot{\mathbf{u}}_s \rrbracket^k, \tag{66}$$

$$\llbracket \mathbf{u}_s \rrbracket^k = \Delta t_s \langle \dot{\mathbf{u}}_s \rangle + \left( \beta_s - \frac{1}{2} \gamma \right) \Delta t_s^2 \llbracket \ddot{\mathbf{u}}_s \rrbracket^k. \tag{67}$$

Premultiplying Eq. (65) by  $[\dot{\mathbf{u}}_s^T]^k$  and then substituting Newmark's Eqs. (66) and (67) in Eq. (65) and rearranging the terms give

$$\begin{aligned} \frac{1}{2} [\dot{\mathbf{u}}_s^T \mathbf{Q}_s \ddot{\mathbf{u}}_s]^k + \frac{1}{2} [\dot{\mathbf{u}}_s^T \mathbf{K}_s \dot{\mathbf{u}}_s]^k &= - \left( \gamma_s - \frac{1}{2} \right) [\dot{\mathbf{u}}_s^T]^k \mathbf{Q}_s [\ddot{\mathbf{u}}_s]^k \\ &\quad - \frac{1}{\Delta t_s} [\dot{\mathbf{u}}_s^T]^k \mathbf{E}_s^T [\mathbf{\Lambda}_s]^k, \end{aligned} \quad (68)$$

where we use the identity  $(\ddot{\mathbf{u}}_s)^T \mathbf{Q}_s [\ddot{\mathbf{u}}_s]^k = \frac{1}{2} [\dot{\mathbf{u}}_s^T \mathbf{Q}_s \ddot{\mathbf{u}}_s]^k$ . Summing up Eq. (68) over all subdomains yields

$$\begin{aligned} \left[ \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{Q} \ddot{\mathbf{u}} + \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{K} \dot{\mathbf{u}} \right]^n &= - \sum_{s=1}^{n_s} \sum_{k=1}^{n_{t_s}} \left( \gamma_s - \frac{1}{2} \right) [\dot{\mathbf{u}}_s^T]^k \mathbf{Q}_s [\ddot{\mathbf{u}}_s]^k \\ &\quad - \sum_{s=1}^{n_s} \sum_{k=1}^{n_{t_s}} \frac{1}{\Delta t_s} [\dot{\mathbf{u}}_s^T]^k \mathbf{E}_s^T [\mathbf{\Lambda}_s]^k. \end{aligned} \quad (69)$$

□

**Lemma 2** *The total energy increment of a system decomposed into  $n_s$  subdomains with  $n_{t_s}$  time steps is given as*

$$\begin{aligned} \left[ \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{Q} \ddot{\mathbf{u}} + \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{K} \dot{\mathbf{u}} \right]^n &= - \sum_{s=1}^{n_s} \sum_{k=1}^{n_{t_s}} \left( \gamma_s - \frac{1}{2} \right) [\dot{\mathbf{u}}_s^T]^k \mathbf{Q}_s [\ddot{\mathbf{u}}_s]^k \\ &\quad - \frac{1}{\Delta T} \sum_{s=1}^{n_s} [\dot{\mathbf{u}}_s^T]^n \mathbf{E}_s^T [\mathbf{\Lambda}_s]^n. \end{aligned} \quad (70)$$

*Proof* Recall that in Proposition (1) (see Eq. (34)), the intermediate local Lagrange multipliers are computed as

$$\mathbf{\Lambda}_s^k = \left( 1 - \frac{k}{n_{t_s}} \right) \mathbf{\Lambda}_s^n + \left( \frac{k}{n_{t_s}} \right) \mathbf{\Lambda}_s^{n+1}, \quad (71)$$

and thus

$$[\mathbf{\Lambda}_s]^k = \mathbf{\Lambda}_s^{k+1} - \mathbf{\Lambda}_s^k = \frac{1}{n_{t_s}} (\mathbf{\Lambda}_s^{n+1} - \mathbf{\Lambda}_s^n) = \frac{1}{n_{t_s}} [\mathbf{\Lambda}_s]^n, \quad (72)$$

where the local and global jump operators are given in Eq. (64). Utilizing Eq. (72), the last right-hand side term of Eq. (69) reads

$$\begin{aligned} \sum_{k=1}^{n_{t_s}} [\dot{\mathbf{u}}_s^T]^k \mathbf{E}_s^T [\mathbf{\Lambda}_s]^k &= \sum_{k=1}^{n_{t_s}} \frac{1}{n_{t_s}} \left( (\dot{\mathbf{u}}_s^{k+1})^T - (\dot{\mathbf{u}}_s^k)^T \right) \mathbf{E}_s^T [\mathbf{\Lambda}_s]^n \\ &= \frac{1}{n_{t_s}} [\dot{\mathbf{u}}_s^T]^n \mathbf{E}_s^T [\mathbf{\Lambda}_s]^n. \end{aligned} \quad (73)$$

Substituting Eq. (73) into Lemma (1) and noting that  $\Delta T = n_{t_s} \Delta t_s$  give

$$\begin{aligned} \left[ \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{Q} \ddot{\mathbf{u}} + \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{K} \dot{\mathbf{u}} \right]^n &= - \sum_{s=1}^{n_s} \sum_{k=1}^{n_{t_s}} \left( \gamma_s - \frac{1}{2} \right) [\dot{\mathbf{u}}_s^T]^k \mathbf{Q}_s [\ddot{\mathbf{u}}_s]^k \\ &\quad - \frac{1}{\Delta T} \sum_{s=1}^{n_s} [\dot{\mathbf{u}}_s^T]^n \mathbf{E}_s^T [\mathbf{\Lambda}_s]^n. \end{aligned} \quad (74)$$

□

**Theorem 1** *PASTA-DDM is energy conservative scheme, provided that the underlying integration methods for the subdomains are energy preserving (i.e., for Newmark's integration parameter  $\gamma_s = 1/2$ ). The total increment of the mechanical energy of the system is zero:*

$$\left[ \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{M} \ddot{\mathbf{u}} + \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{K} \dot{\mathbf{u}} \right]^n = 0. \quad (75)$$

*Proof* From the compatibility and equilibrium equations (see Eq. (27)), we can write the following expressions

$$\mathbf{E}_s [\dot{\mathbf{u}}_s]^n = \mathbf{B}_s [\Psi]^n, \quad (76)$$

$$\sum_{s=1}^{n_s} -\mathbf{B}_s^T [\mathbf{\Lambda}_s]^n = \mathbf{0}. \quad (77)$$

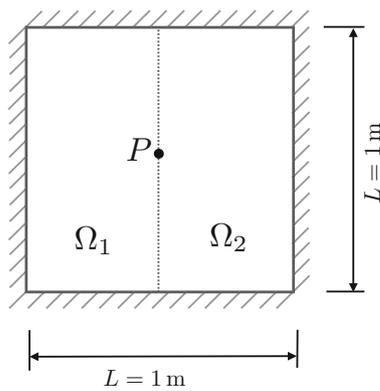
Substituting Eqs. (76) and (77) into Lemma (2) and using Newmark's integration parameter without numerical dissipation ( $\gamma_s = 1/2$ ) lead to

$$\left[ \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{M} \ddot{\mathbf{u}} + \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{K} \dot{\mathbf{u}} \right]^n = 0. \quad (78)$$

Therefore, PASTA-DDM conserves the total discrete energy and is an unconditionally stable scheme. □

## 5 Numerical results

In this section, we present verification studies and an engineering application of a sandwich plate impact problem. First, we consider the Method of Manufactured Solutions (MMS) to evaluate the spatial and temporal order of convergence of PASTA-DDM (see [55]). Second, we verify the fulfillment of the jump conditions along the common refinement interface (i.e., to illustrate mass, momentum and energy conservation properties of PASTA-DDM across the non-matching meshes). Next, to demonstrate the applicability of PASTA-DDM to practical engineering problems, we consider a sandwich plate impacted by a projectile. In such simulations, local treatment of both the mesh and time resolutions is crucial to preserve accuracy and efficiency in a heterogeneous parallel computing environment.



**Fig. 6** A diagram of the computational domain for mesh size and time increment convergence studies.  $P$  is a point on the interface having coordinates  $x = 0.5$  m and  $y = 0.5$  m

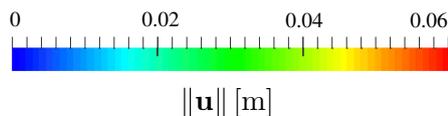
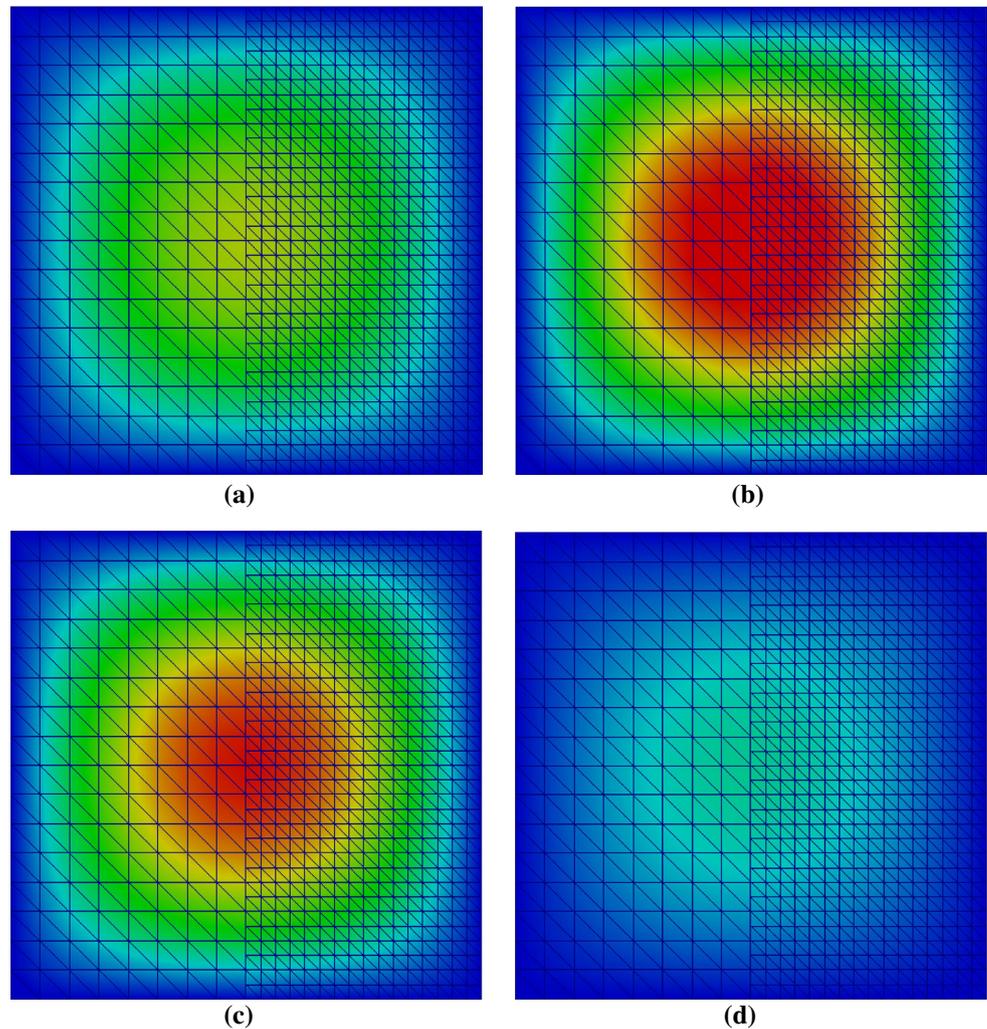
### 5.1 Spatial and temporal order of convergence

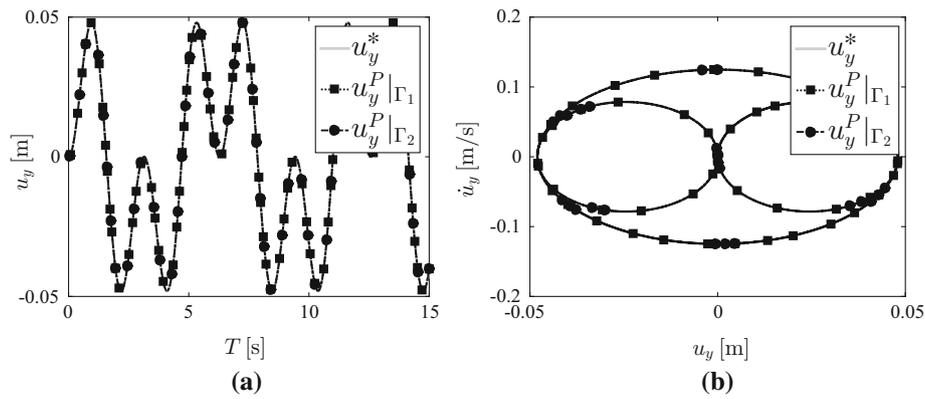
As shown in Fig. 6, a unit square computational domain that is decomposed into two non-overlapping subdomains at  $x = 0.5$  m is considered for the convergence studies. The material properties for both subdomains are: Young’s modulus  $E = 210$  GPa, Poisson’s ratio  $\nu = 0.3$  and density  $\rho = 7800$  kg/m<sup>3</sup>. The implicit Newmark algorithm is used for both subdomains. Furthermore, the following manufactured displacement field is used:

$$\mathbf{u}^*(\mathbf{x}, t) = \begin{Bmatrix} x(L-x)y(L-y) \\ x(L-x)y(L-y) \end{Bmatrix} \sin(\omega t) \sin(2\omega t). \quad (79)$$

For non-matching meshes ( $h_1/h_2 = 2$ ) and asynchronous time stepping ( $\Delta t_1:\Delta t_2 = 1:3$ ), Fig. 7 shows snapshots of

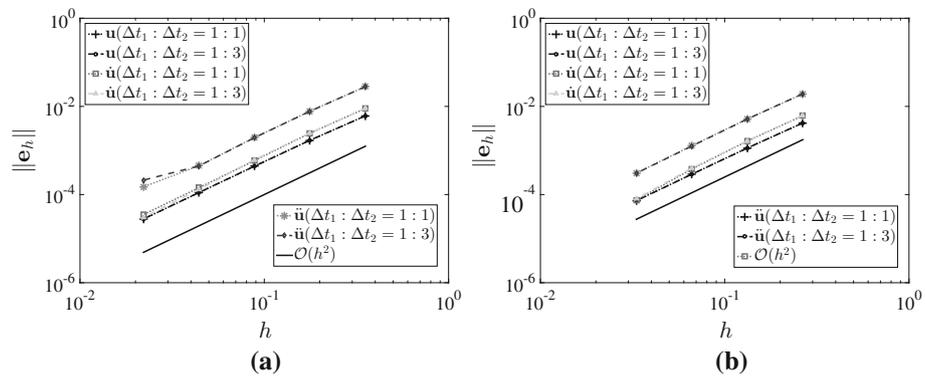
**Fig. 7** Snapshots of the displacement magnitude at different synchronization time steps. In this figure,  $h_1 = 0.0884$  m,  $h_2 = 0.0442$  m,  $\Delta t_1:\Delta t_2 = 1:3$  and  $\Delta T = 0.0375$  s. **a**  $T = 0.6$  s. **b**  $T = 0.9$  s. **c**  $T = 1.2$  s. **d**  $T = 1.5$  s





**Fig. 8** Time history of the vertical component of the displacement field and phase space diagram at point  $P$ . The manufactured solution at point  $P$  is denoted by  $u_y^*$ , while the PASTA-DDM solution from subdomain

$\Omega_1$  and  $\Omega_2$  on the interface is denoted as  $u_y^P|_{\Gamma_1}$  and  $u_y^P|_{\Gamma_2}$ . **a** Time history. **b** Phase space diagram



**Fig. 9** Rate of convergence of PASTA-DDM for displacement, velocity and acceleration fields with respect to mesh size  $h$  for matching and non-matching meshes with synchronous  $\Delta t_1:\Delta t_2 = 1:1$  and asyn-

chronous  $\Delta t_1:\Delta t_2 = 1:3$  time stepping. The synchronization time is  $\Delta T = 0.0375$  s. **a** Matching meshes. **b** Non-matching meshes

the displacement magnitude at different synchronization time steps. Note that the fields are continuous across the interface in the weak sense.

For the case of non-matching grids ( $h_1/h_2 = 2$ ) and asynchronous time stepping ( $\Delta t_1:\Delta t_2 = 1:3$ ), in Fig. 8 we show the time history of the vertical component of the displacement field and phase space diagram at point  $P$  (see Fig. 6). The results from PASTA-DDM are compared to the manufactured solution in Eq. (79).

For both cases of matching and non-matching grids, the convergence properties of PASTA-DDM with respect to the mesh resolution are illustrated in Fig. 9. The spatial error at a given instant of time is computed as

$$\|e_h\| = \sqrt{\sum_{s=1}^{n_s} \int_{\Omega_s} (\mathbf{u}^*(\mathbf{x}, T_f) - \mathbf{u}_s(\mathbf{x}, T_f))^2 d\Omega}, \quad (80)$$

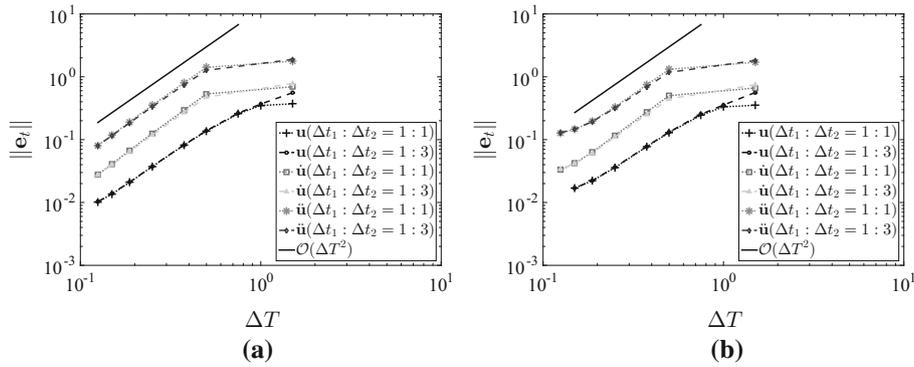
where  $T_f = 15$  s. Note that for transient analysis, measuring the spatial error at a given time is typical [12,42,44,54], and the time integration of Eq. (80) only adds a constant shift. For the error in velocity, the same error measure, Eq. (80),

is used with the appropriate manufactured velocity field,  $\dot{\mathbf{u}}^*$ , obtained from Eq. (79) and the corresponding velocity field  $\dot{\mathbf{u}}_s$  define in Eq. (32). For the Newmark time integration method (a constant average acceleration scheme) to achieve a second order accuracy for the acceleration, a post-processing step is required [9,22,64,65]. Therefore, the average acceleration within a synchronization time step is calculated as

$$\ddot{\mathbf{u}}_s^{n+1/2}(\Delta T) = \frac{\Delta T}{\Delta t_s} \sum_{k=1}^{n_s} \left( \frac{\ddot{\mathbf{u}}_s^{k+1} - \ddot{\mathbf{u}}_s^k}{2} \right), \quad (81)$$

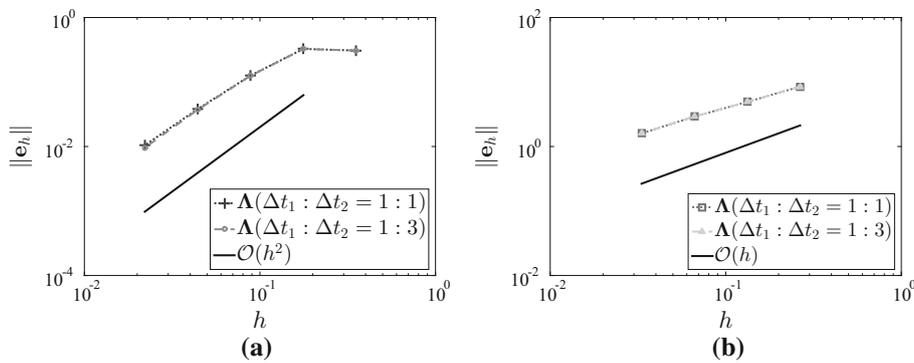
where  $k$  is the local and  $n$  is the global (synchronization) time step (see Fig. 3). We use Eq. (81) together with the appropriate manufactured acceleration field,  $\ddot{\mathbf{u}}^*$  (Eq. (79)), to compute the spatial error given by Eq. (80).

In Fig. 9a, we refine both subdomain meshes simultaneously, while in Fig. 9b the ratio of mesh size ( $h_1/h_2 = 2$ ) is kept constant and both subdomains are refined at the same time. For clarity of the presentation, in Fig. 9b we plot the norm of the error against the average mesh size. For the linear



**Fig. 10** Rate of convergence of PASTA-DDM for displacement, velocity and acceleration with respect to the synchronization time increment  $\Delta T$  for matching and non-matching meshes with synchronous  $\Delta t_1:\Delta t_2 = 1:1$  and asynchronous  $\Delta t_1:\Delta t_2 = 1:3$  time stepping.

For matching grids (a),  $h_1 = h_2 = 0.1768$  m and for non-matching grids (b),  $h_1 = 0.2750$  m and  $h_2 = 0.1768$  m. **a** Matching meshes. **b** Non-matching meshes



**Fig. 11** Rate of convergence of PASTA-DDM for the local Lagrange multipliers with respect to mesh size  $h$  for matching and non-matching meshes with synchronous  $\Delta t_1:\Delta t_2 = 1:1$  and asynchronous  $\Delta t_1:\Delta t_2 =$

1:3 time stepping. The synchronization time is  $\Delta T = 0.0375$  s. **a** Matching meshes. **b** Non-matching meshes

finite elements used, PASTA-DDM achieves the optimal second order convergence properties (in displacement, velocity and average acceleration) with respect to the spatial mesh size for both matching and non-matching grids as well as synchronous and asynchronous time integration.

In Fig. 10, we demonstrate the rate of convergence of PASTA-DDM (in displacement, velocity and acceleration) with respect to the time increment for both matching and non-matching meshes, and synchronous and asynchronous time stepping. The temporal error is computed as

$$\|e_t\| = \sqrt{\sum_{s=1}^{n_s} \int_{\Omega_s} \int_T (\mathbf{u}^*(\mathbf{x}, t) - \mathbf{u}_s(\mathbf{x}, t))^2 dt d\Omega}. \quad (82)$$

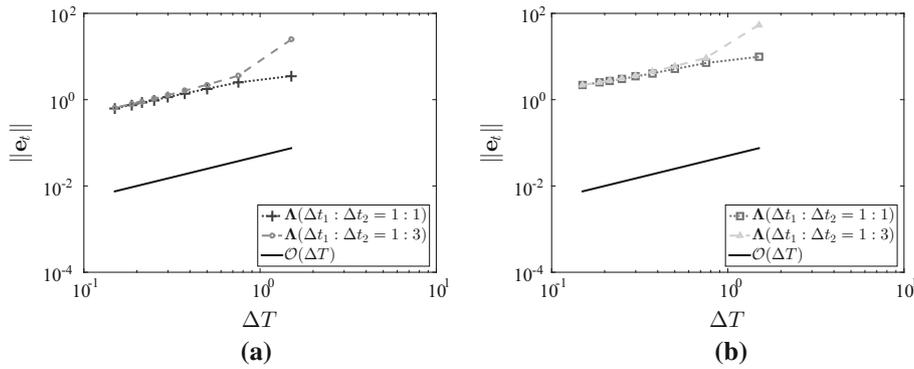
For consistency, the error norms are plotted against the synchronization time step  $\Delta T$ . The time integration interval is  $T \in [0, 150]$  s. Once again for the error in velocity, the same error measure, Eq. (82), is used with the appropriate manufactured velocity field,  $\dot{\mathbf{u}}^*$ , obtained from Eq. (79), and the corresponding velocity field,  $\dot{\mathbf{u}}_s$ , defined in Eq. (32). Moreover, the error in the acceleration is computed using Eq. (82), whereby

the manufactured acceleration field,  $\ddot{\mathbf{u}}^*$ , is obtained from Eq. (79), and the corresponding averaged acceleration field,  $\ddot{\mathbf{u}}_s$ , is calculated using Eq. (81). As shown in Fig. 10, PASTA-DDM maintains the second order convergence properties in displacement, velocity and average acceleration of the underlying implicit Newmark scheme with the conservative integration parameters ( $\gamma_s = 1/2, \beta_s = 1/4$ ) [25]. Note that the convergence results in Fig. 10b are tailing off slightly for the last time step increment due to the spatial error pollution.

Figure 11 shows the convergence rate of the local Lagrange multipliers in PASTA-DDM with respect to mesh size for matching and non-matching meshes with synchronous and asynchronous time stepping. The spatial error of the Lagrange multipliers at a given instant of time is calculated as

$$\|e_h\| = \sqrt{\sum_{s=1}^{n_s} \int_{\Gamma_s} (\Lambda^*(\mathbf{x}, T_f) - \Lambda_s(\mathbf{x}, T_f))^2 d\Gamma_s}, \quad (83)$$

where  $T_f = 15$  s. The manufactured solution, Eq. (79), is used to compute  $\Lambda^*$  and  $\Lambda_s$  is the vector of sub-domain Lagrange multipliers. For matching meshes with



**Fig. 12** Rate of convergence of PASTA-DDM for the Lagrange multipliers with respect to the synchronization time increment  $\Delta T$  for matching and non-matching meshes with synchronous  $\Delta t_1:\Delta t_2 = 1:1$  and asynchronous  $\Delta t_1:\Delta t_2 = 1:3$  time stepping. For matching

meshes (a),  $h_1 = h_2 = 0.1768$  m and for non-matching meshes (b),  $h_1 = 0.1768$  m and  $h_2 = 0.0884$  m. **a** Matching meshes. **b** Non-matching meshes

synchronous and asynchronous time stepping, the Lagrange multipliers exhibit the quadratic convergence rate with respect to mesh size (Fig. 11a). However, for non-matching discretization the Lagrange multipliers show a first order convergence rate (Fig. 11b). Note that the stress field is piecewise constant in our analysis and also first order convergent.

The time convergence of the Lagrange multipliers with respect to the synchronization time step is shown in Fig. 12. The temporal error of the local Lagrange multipliers is computed as

$$\|e_t\| = \sqrt{\sum_{s=1}^{n_s} \int_{\Gamma_s} \int_T (\Lambda^*(\mathbf{x}, t) - \Lambda_s(\mathbf{x}, t))^2 dt d\Gamma_s}. \quad (84)$$

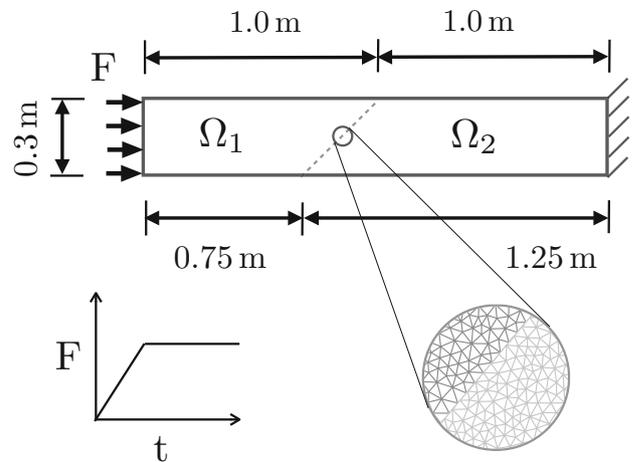
As shown in Fig. 12, for matching and non-matching meshes with synchronous and asynchronous time stepping, the Lagrange multipliers show a linear convergence rate with respect to the synchronization time step.

### 5.2 Verification of the jump conditions

Due to non-matching meshes on both sides of the interface, the global field variables do not satisfy the continuity conditions at the interface, and thus suffer finite jump discontinuities. In this case, the global balance laws imply that the jump conditions must hold on the interface, where the field variables do not maintain the classical degree of smoothness [28]. Accordingly, the mass balance, linear momentum balance and the first law of thermodynamics require that the following jump conditions hold on the interface (material singular surface):

$$\llbracket \dot{\mathbf{u}} \rrbracket_{\Gamma_c} = \int_{\Gamma_c} (\dot{\mathbf{u}}|_{\Gamma_1} - \dot{\mathbf{u}}|_{\Gamma_2}) d\Gamma_c = 0, \quad (85)$$

$$\llbracket \Lambda \rrbracket_{\Gamma_c} = \int_{\Gamma_c} (\Lambda|_{\Gamma_1} - \Lambda|_{\Gamma_2}) d\Gamma_c = 0, \quad (86)$$



**Fig. 13** A cantilever beam decomposed into two subdomains with non-matching meshes at the interface. The interface is inclined by  $75.964^\circ$

$$\llbracket \dot{\mathbf{u}}^T \Lambda \rrbracket_{\Gamma_c} = \int_{\Gamma_c} (\dot{\mathbf{u}}|_{\Gamma_1}^T \Lambda|_{\Gamma_1} - \dot{\mathbf{u}}|_{\Gamma_2}^T \Lambda|_{\Gamma_2}) d\Gamma_c = 0, \quad (87)$$

where  $\Gamma_1, \Gamma_2$  and  $\Gamma_c$  are the left, right and common interfaces, respectively. Analogously, the kinematic continuity reads:

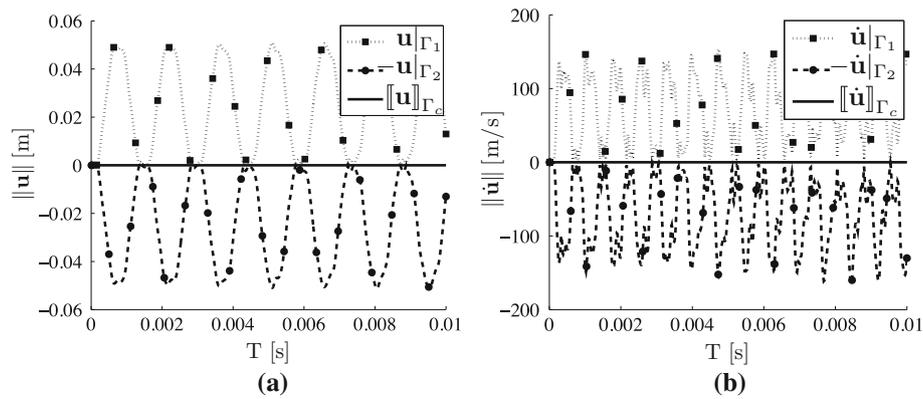
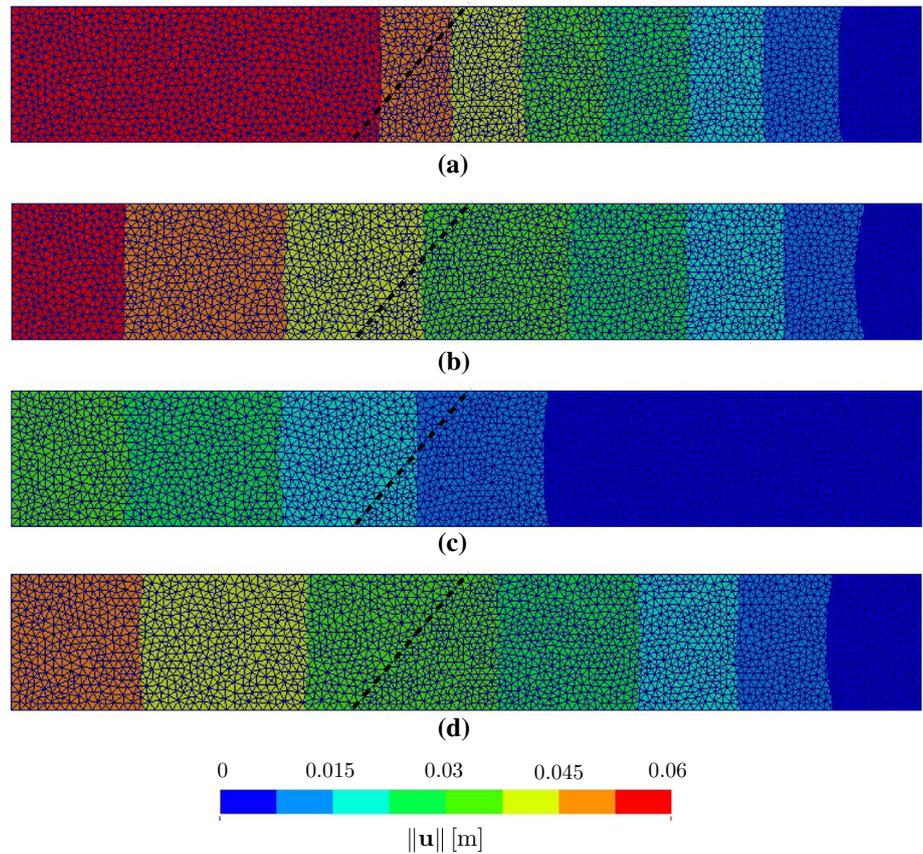
$$\llbracket \mathbf{u} \rrbracket_{\Gamma_c} = \int_{\Gamma_c} (\mathbf{u}|_{\Gamma_1} - \mathbf{u}|_{\Gamma_2}) d\Gamma_c = 0. \quad (88)$$

To verify the fulfillment of the jump conditions and energy conservation properties of PASTA-DDM across the interface, we consider a cantilever beam clamped at the right-hand side, while a time-dependent horizontal traction is applied on the left end. The beam is decomposed into two non-overlapping subdomains glued together by an inclined surface creating non-matching meshes across the interface as shown in Fig. 13.

For the numerical implementation, we consider the following material properties for both subdomains: Young’s

**Fig. 14** Snapshots of the magnitude of the displacement field at different synchronization time steps.

- a**  $T = 0.675 \times 10^{-3}$  s.
- b**  $T = 1.962 \times 10^{-3}$  s.
- c**  $T = 3.235 \times 10^{-3}$  s.
- d**  $T = 3.975 \times 10^{-3}$  s

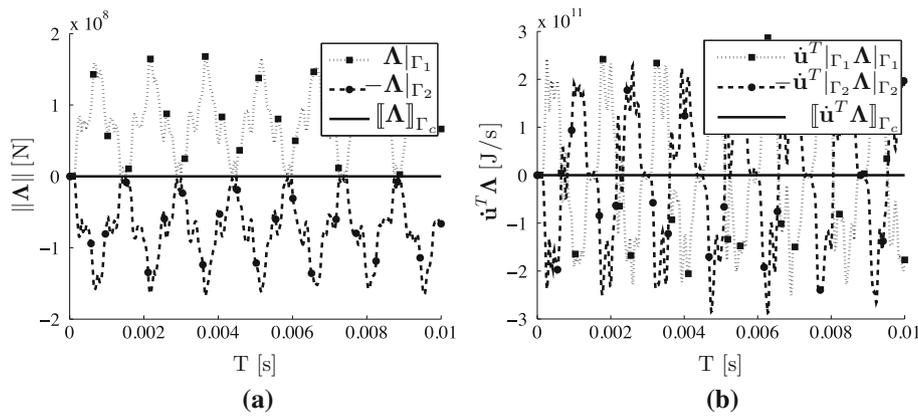


**Fig. 15** Verification of the jump conditions in PASTA-DDM: **(a)** displacement and **(b)** velocity along the interface,  $\Delta t_1 : \Delta t_2 = 1:3$  and  $h_1 = 0.0301$  m,  $h_2 = 0.0255$  m. **a** Jump in displacement. **b** Jump in velocity

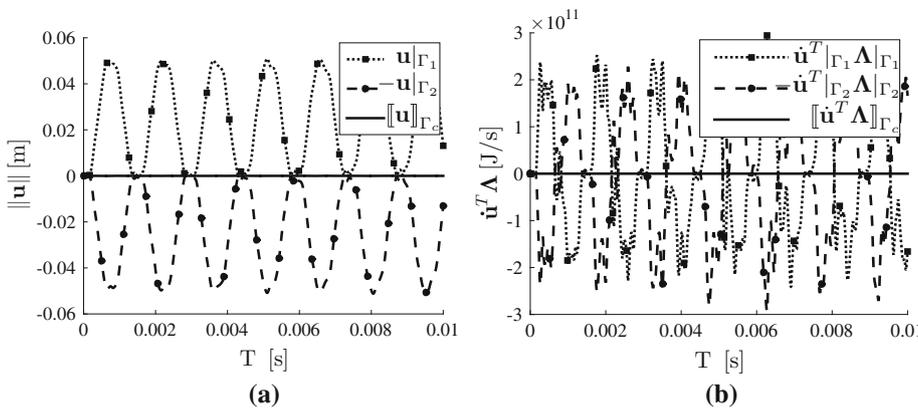
modulus  $E = 210$  GPa, Poisson’s ratio  $\nu = 0.3$  and density  $\rho = 7800$  kg/m<sup>3</sup>. The traction reaches its full value of  $5.0 \times 10^9$  kN/m in  $2.5 \times 10^{-5}$  s. The finite element discretizations for subdomain  $\Omega_1$  and subdomain  $\Omega_2$  consist of 1759 and 2775 elements, respectively. The system is integrated up to  $T_f = 0.01$  s with time steps  $\Delta t_1 = 1.25 \times 10^{-5}$  s and  $\Delta t_2 = 3.75 \times 10^{-5}$  s for  $\Omega_1$  and  $\Omega_2$ , respectively. The implicit Newmark integrator is used for both subdomains.

Figure 14 shows the displacement magnitude at different synchronization time steps. Figures 15 and 16 show the jump in the displacement, velocity, local Lagrange multipliers and

power across the interface (see Eqs. (85) to (88)). For clarity of presentation, the quantities related to the second local interface,  $\Gamma_2$ , are plotted with a negative sign. The maximum absolute values in displacement, velocity, Lagrange multiples and power jumps across the common-refinement interface are  $[\mathbf{u}]_{\Gamma_c} = 3.136 \times 10^{-5}$  m,  $[\dot{\mathbf{u}}]_{\Gamma_c} = 3.835 \times 10^{-7}$  m/s,  $[\mathbf{\Lambda}]_{\Gamma_c} = 1.601 \times 10^{-6}$  N, and  $[\dot{\mathbf{u}}^T \mathbf{\Lambda}]_{\Gamma_c} = 1.156 \times 10^{-1}$  J/s. As shown in Figs. 15 and 16, PASTA-DDM satisfies the jump conditions across the interface. Thus, PASTA-DDM preserves the mass, momentum and energy balance across the interface.



**Fig. 16** Verification of the jump conditions in PASTA-DDM: **(a)** Lagrange multipliers and **(b)** power along the interface,  $\Delta t_1:\Delta t_2 = 1:3$  and  $h_1 = 0.0301$  m,  $h_2 = 0.0255$  m. **a** Jump in Lagrange multipliers. **b** Jump in power



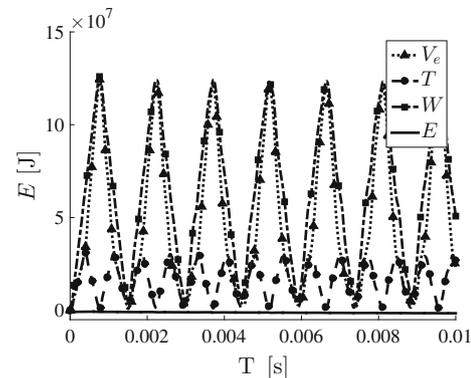
**Fig. 17** Verification of the jump conditions in PASTA-DDM: **(a)** displacement and **(b)** power on the interface,  $\Delta t_1:\Delta t_2 = 1:10$  and  $h_1 = 0.0301$  m,  $h_2 = 0.0255$  m. **a** Jump in displacement. **b** Jump in power

To verify the jump conditions with respect to a larger time asynchrony ratio  $\Delta t_1:\Delta t_2$ , in Fig. 17, we show the jump in the displacement and power along the interface using one order of magnitude of the asynchrony ratio  $\Delta t_1:\Delta t_2 = 1:10$ . Although not shown (for the brevity of the article), similar results are obtained for the jump in velocity and the local Lagrange multipliers, and no distinction can be made with the results previously shown for  $\Delta t_1:\Delta t_2 = 1:3$  in Figs. 15 and 16.

Figure 18 displays the total mechanical energy balance of the system. As can be seen, PASTA-DDM conserves the total energy of the system. Again, we verified the total mechanical energy balance using the asynchrony ratio  $\Delta t_1:\Delta t_2 = 1:10$ , and no discrepancy can be seen between the results of  $\Delta t_1:\Delta t_2 = 1:3$ .

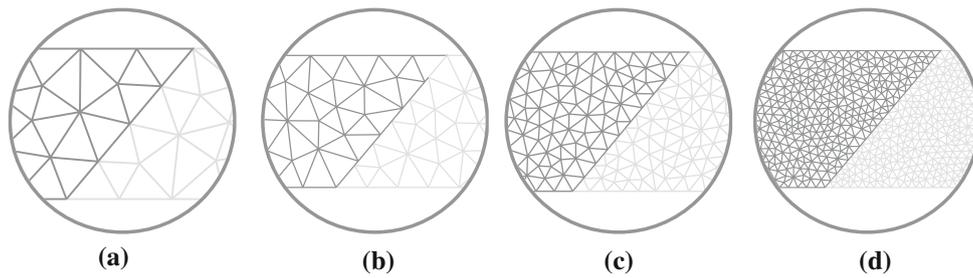
### 5.3 Error assessment of the interface quantities

In this subsection, we continue the analysis of the example introduced in Subsect. 5.2. In particular, we quantify the error in jump conditions due to the non-matching meshes and asynchronous time stepping. Four different meshes with



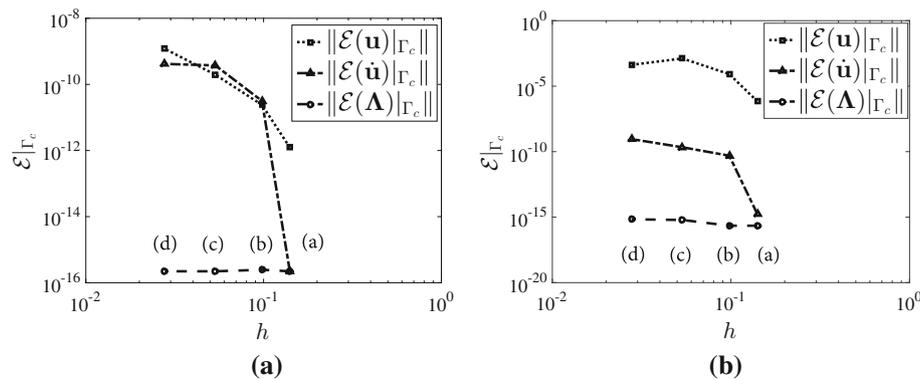
**Fig. 18** The total energy balance of PASTA-DDM,  $\Delta t_1:\Delta t_2 = 1:3$  and  $h_1 = 0.0301$  m,  $h_2 = 0.0255$  m. Here,  $V_e = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u}$  is the elastic energy,  $T = \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{M} \dot{\mathbf{u}}$  is the kinetic energy,  $W = \mathbf{u}^T \mathbf{f}$  represents work due to external forces, and  $E = T + V_e - W$  denotes the total energy of the system

increasing amount of non-matching discretizations along the interface are considered as shown in Fig. 19. The mesh in Fig. 19a represents the case of matching grids, while the subsequent meshes carry larger and larger degree of mesh



**Fig. 19** Magnified insets of the subdomain meshes around the interface showing the amount of mesh incompatibility. The mesh density in subdomains  $\Omega_1$  and  $\Omega_2$  is: **a**  $h_1 = 0.1515$  m and  $h_2 = 0.1294$  m,

**b**  $h_1 = 0.1008$  m and  $h_2 = 0.0949$  m, **c**  $h_1 = 0.0551$  m and  $h_2 = 0.0530$  m and **d**  $h_1 = 0.0301$  m and  $h_2 = 0.0255$  m



**Fig. 20** The relative error convergence with respect to the mean mesh size for both **(a)** synchronous  $\Delta t_1:\Delta t_2 = 1:1$  and **(b)** asynchronous  $\Delta t_1:\Delta t_2 = 1:3$  time integration. The synchronization time step is

$\Delta T = 1.25 \times 10^{-5}$  s. The markers **(a)** to **(d)** indicate the meshes used in the analysis (see Fig. 19). **a** Synchronous time integration. **b** Asynchronous time integration

incompatibility (see Fig. 19b–d). Thus, as we refine the meshes, the degree of mesh incompatibility increases along the common interface (i.e., several small interface segments are created along the common-refinement). The relative error due to mesh incompatibility at a given time is defined as

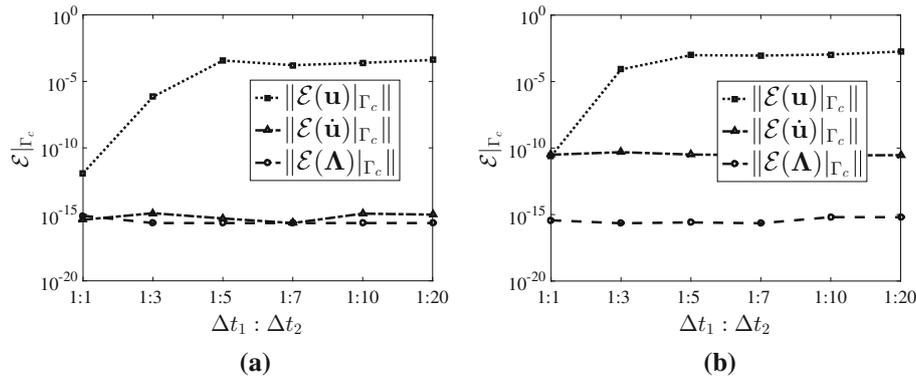
$$\mathcal{E}(\bullet)|_{\Gamma_c} = \frac{\int_{\Gamma_c} (\bullet|_{\Gamma_1}(\mathbf{x}, T_f) - \bullet|_{\Gamma_2}(\mathbf{x}, T_f)) d\Gamma_c}{\frac{1}{2} \int_{\Gamma_c} (\bullet|_{\Gamma_1}(\mathbf{x}, T_f) + \bullet|_{\Gamma_2}(\mathbf{x}, T_f)) d\Gamma_c}, \quad (89)$$

where  $T_f = 0.01$  s is the end of time integration period.

Convergence of the relative error with respect to the amount of non-matching meshes for both synchronous and asynchronous integration is shown in Fig. 20. For consistency, the relative error is plotted with respect to the average mesh size. In both cases of synchronous and asynchronous time integration, the results indicate that the equilibrium between the subdomains is satisfied strongly (up to the machine precision) regardless of the amount of non-matching discretization. As expected, the relative error in the kinematic constraint,  $\mathbf{u}$  and  $\dot{\mathbf{u}}$ , (which is enforced weakly) grows as the degree of mesh incompatibility increases. Moreover, the displacement error is more pronounced, since the common interface is solved in terms of velocities. However,

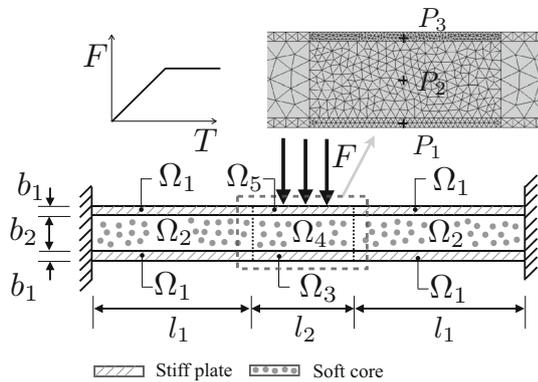
the error growth saturates as meshes are refined, which is a consequence of the common-refinement-based projection properties [29,30]. In addition, Fig. 20b displays the consequence of the asynchronous time stepping. Error in velocities and the local Lagrange multipliers are largely unaffected (error stays around  $10^{-10}$ ). However, the displacement errors suffer a constant shift in value, since they are being integrated using a global (i.e. common refinement) time increment  $\Delta T$ .

Next in Fig. 21, we quantify the error due to the time-step asynchrony ratio  $\Delta t_1:\Delta t_2$  for both matching and non-matching grids. Here, we study asynchrony ratio up to 1:20. Note that a large time asynchrony ratio  $\Delta t_1:\Delta t_2$  should be considered in conjunction with accuracy requirement. This is analogous to mesh incompatibility (i.e.  $h_1:h_2$ ). Large time step differences (e.g.  $\Delta t_1:\Delta t_2 = 1:100$ ), should be bridged with care using several nested transition zones ( $5 \times (\Delta t_1:\Delta t_2 = 1:20)$ ). As shown in Fig. 21, the relative error in the local Lagrange multipliers, for both cases, is zero (up to the machine precision). Increasing the ratio of asynchrony does not change the magnitude of the relative error in the velocity. However, the non-matching meshes introduce a constant error shift. In both cases, the relative error in displacements shows a mild growth when increasing the time



**Fig. 21** The relative error convergence with respect to time-step asynchrony ratio  $\Delta t_1:\Delta t_2$ , for both matching and non-matching meshes. For case of matching interface, we use the mesh shown in Fig. 19a, while

for non-matching case we use the mesh shown in Fig. 19b. The synchronization time step is  $\Delta T = 1.25 \times 10^{-5}$  s. **a** Matching meshes. **b** Non-matching meshes



**Fig. 22** A schematic diagram of a projectile impacting a sandwich plate. The points of interest have coordinates  $P_1 = (0.1, 0.001)$  m,  $P_2 = (0.1, 0.01)$  m and  $P_3 = (0.1, 0.019)$  m

asynchrony ratio. Once more, this is expected due to the fact that the overall accuracy of the solution is bounded by the global synchronization time step  $\Delta T$ . Note again that the error in all fields saturates as  $\Delta t_1:\Delta t_2$  grows.

**5.4 Engineering impact problem with multiple subdomains and implicit/explicit time integration**

As an illustration of PASTA-DDM to practical engineering problems, we consider a sandwich plate composed of stiff and soft materials impacted by a projectile. This example is inspired by Prakash et al. [52]. In this multi-scale problem, the subdomain meshes and time steps are selected based on the local dynamics. This leads to non-matching meshes at the interfaces and subdomains advancing in time asynchronously. Moreover, a mixed time integration scheme (i.e., implicit and explicit Newmark’s algorithms) is utilized.

A diagram detailing the impact problem is shown in Fig. 22. The composite plate is made from a stiff skin and

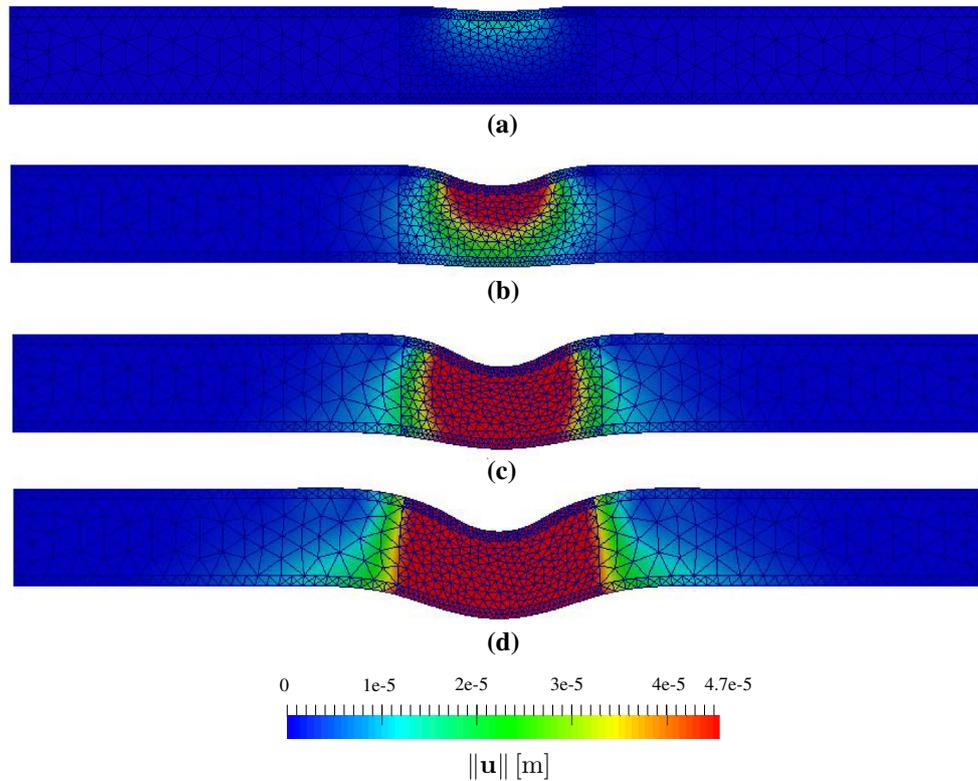
a soft core. The material properties of the stiff skin are  $E = 70$  GPa,  $\mu = 0.3$ ,  $\rho = 2700$  kg/m<sup>3</sup> and of the core are  $E = 7.78$  GPa,  $\mu = 0.3$ ,  $\rho = 900$  kg/m<sup>3</sup>. The physical dimension are as follows  $b_1 = 2$  mm,  $b_2 = 16$  mm,  $l_1 = 80$  mm and  $l_2 = 60$  mm. The projectile effect is modeled as a traction with a ramp function that reaches its peak value 100 kN/m in  $1 \times 10^{-5}$  s. The final time for the simulation is  $T_f = 2 \times 10^{-5}$  s.

The impact problem contains two time scales. Fast dynamics below the impact zone and slow dynamics away from the region around the projectile [52]. While considering the physical properties of the problem, we partition the computational domain into 9 non-overlapping subdomains (due to symmetry, we only mark 5 subdomains in Fig. 22 for clarity). Note that the cross points (a node shared among more than two subdomains), as shown in the mesh insert of Fig. 22, do not require any special treatment in the framework of local Lagrange multipliers, because the non-uniqueness of the constraint is eliminated [47–49]. Fine mesh resolution is used for the subdomains near the impact zone (subdomains  $\Omega_3$ ,  $\Omega_4$  and  $\Omega_5$ ). Moreover, as the dynamics are slow in the regions far from the impact zone, subdomains with coarser meshes are used away from the region of interest (subdomains  $\Omega_1$  and  $\Omega_2$ ). To estimate the required time step that captures the dynamics of the problem, we use the material wave propagation speed  $c_s = \sqrt{E_s/\rho_s}$ . Given the wave speed in each subdomain, we employ the Courant-Friedrichs-Lewy (CFL) condition to select the time step  $\Delta t_s = h_s/c_s$  for each subdomains.

Since dynamics in the region under the projectile are faster than in regions away from the impact zone, the time resolution under the impact zone is governed by accuracy rather than stability. Thus, we use the explicit Newmark scheme (i.e.  $\gamma_s = 1/2$ ,  $\beta_s = 0$ ) in  $\Omega_5$ , whereas the implicit Newmark scheme (i.e.  $\gamma_s = 1/2$ ,  $\beta_s = 1/4$ ) is employed for the rest of the subdomains.

**Table 1** Physical, spatial and temporal parameters for a composite plate impact problem

Subdomain	No. nodes	No. elem	$\Delta t_s$ [s]	$E_s$ [GPa]	$\rho_s$ [kg/m <sup>3</sup> ]	$c_s$ [m/s]
$\Omega_1$	78	87	$\Delta T/2$	70	2700	$5.0918 \times 10^3$
$\Omega_2$	90	136	$\Delta T$	7.78	900	$2.9401 \times 10^3$
$\Omega_3$	126	166	$\Delta T/4$	70	2700	$5.0918 \times 10^3$
$\Omega_4$	239	420	$\Delta T$	7.78	900	$2.9401 \times 10^3$
$\Omega_5$	225	341	$\Delta T/5$	70	2700	$5.0918 \times 10^3$



**Fig. 23** Snapshots of the displacement field magnitude at different synchronization time steps for PASTA-DDM solution. The mesh displacements are 50× magnified. **a**  $T = 6 \times 10^{-6}$  s. **b**  $T = 1.2 \times 10^{-5}$  s. **c**  $T = 1.6 \times 10^{-5}$  s. **d**  $T = 2 \times 10^{-5}$  s

For the sake of comparison, first the problem is solved using the smallest mesh size for all subdomains (i.e.  $h_{min} = 0.5$  mm), which leads to 6, 058 nodes and 10, 691 elements. Note that the CFL condition for this case yields a critical time step  $T_{cr} = 9.4 \times 10^{-8}$  s. Therefore, to obtain an accurate solution we use  $\Delta T = 4 \times 10^{-8}$  s for all the subdomains. We identify this approach as Direct Numerical Modeling (DNM).

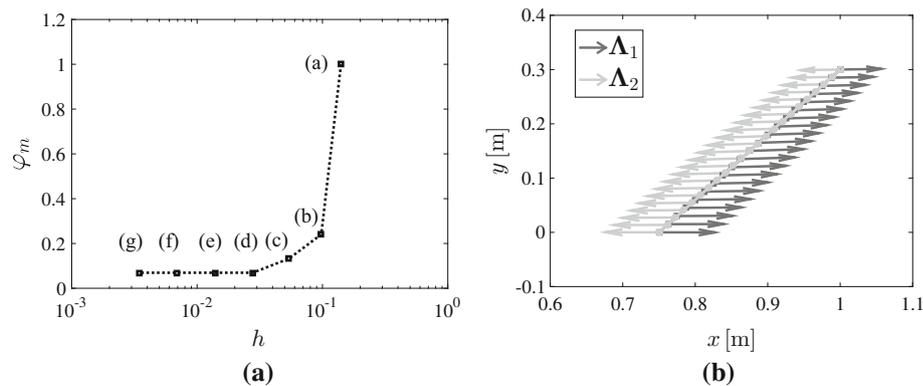
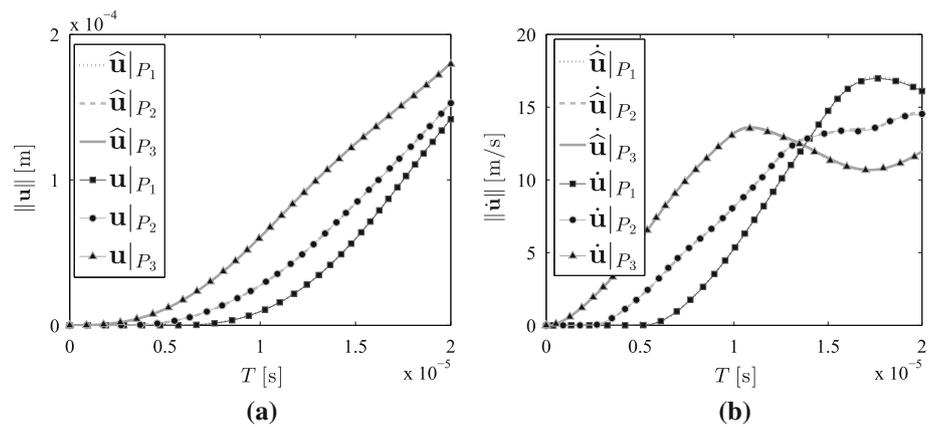
Next, we employ PASTA-DDM and use localized mesh size and asynchronous time stepping. The number of time steps to synchronize is calculated based on the CFL condition of the impact zone,  $\Omega_5$ , where the mesh size is kept as  $h = 0.5$  mm, while coarser meshes are used for the rest of the subdomains as shown in Table 1. Note that in this case, we use a synchronization time step  $\Delta T$  such that  $\Delta t_5 = \Delta T/5 = 4 \times 10^{-8}$  s (i.e.  $\Delta T = 2 \times 10^{-7}$  s), where the number of time steps  $n_{t_s} = 5$  for subdomain  $\Omega_5$ . This approach gives 1, 082 nodes and 1, 547 elements.

In Fig. 23, snapshots of the displacement field magnitude are shown at different synchronization time steps. Figure 24 displays the response at points  $P_1, P_2$  and  $P_3$  along the center line  $x = 0.1$  m (as shown in Fig. 22). In the plot, we compare the displacement and velocity trajectories at points ( $P_1, P_2$  and  $P_3$ ) obtained from DNM and PASTA-DDM. As clearly shown in the figure, PASTA-DDM achieves the same accuracy of the high resolution model, but with smaller number of elements and faster solution time.

### 6 Conclusions

We develop the Asynchronous Space-Time Algorithm based on the Domain Decomposition Method for structural dynamics problems on non-matching meshes. The methodology

**Fig. 24** Displacement and velocity at points along the center line  $x = 0.1$  m. The DNM results are identified as  $\hat{\mathbf{u}}|_{P_i}$  and  $\hat{\mathbf{u}}|_{P_i}$ , while PASTA-DDM results are denoted by  $\mathbf{u}|_{P_i}$  and  $\dot{\mathbf{u}}|_{P_i}$ . **a** Displacement. **b** Velocity



**Fig. 25** The smallest eigenvalue  $\varphi_m = \min(\varphi)$  and the local Lagrange multipliers. The mesh characteristics for discretizations (a), (b), (c) and (d) are given in Fig. 19. The mesh densities in (e) are  $h_1 = 0.0150$  m and  $h_2 = 0.0128$  m; (f) are  $h_1 = 0.0075$  m and  $h_2 = 0.0064$  m; (g) are

$h_1 = 0.0038$  m and  $h_2 = 0.0032$  m. For clarity of presentation, we plot  $\varphi_m$  against the mean mesh size  $h = (h_1 + h_2)/2$ . The local Lagrange multiplier fields are plotted for mesh density (d). **a** Smallest eigenvalue. **b** Local Lagrange multipliers

is based on the dual-primal like domain decomposition technique utilizing the localized Lagrange multipliers. For optimal accuracy and preserving physical quantities, the interface between the non-matching meshes is discretized using the common-refinement-based technique. Moreover, we extend the idea of common refinement to the temporal dimension and introduce a generalized  $\alpha$  method for the local Lagrange multipliers. The algorithm offers two-levels of parallelism and is well suited for a heterogeneous computing environment. For linear dynamical problems, PASTA-DDM is an unconditionally stable scheme and preserves mass, momentum and energy along the common interface. Furthermore, PASTA-DDM maintains the optimal rate of convergence with respect to mesh size and time increment for displacement, velocity and averaged acceleration. The computer implementation is verified using the method of manufactured solutions, and rigorous assessment of mass, momentum and energy jump conditions across the common refinement is performed. A projectile impact problem shows potential of PASTA-DDM for a variety of engineering applications.

**Acknowledgments** This work has been supported by the Department of Energy, National Nuclear Security Administration, under Award No. DE-NA0002377.

## Appendix 1: Inf-sup test

In what follows, we give a brief description of the numerical test for checking the *inf-sup* conditions of the three-field discretization given by Eq (27). This *inf-sup* analysis is taken from work of Brezzi and Marini [8]. The numerical test consists of two generalized eigenvalue problems. The first *inf-sup* problem corresponds to the local Lagrange multipliers and the displacement field discretizations and is local to each subdomain. The second *inf-sup* problem is a global one, and is associated with the local Lagrange multipliers and the common-interface variable discretization. In this section, we neglect the inertia forces and consider only an elliptic system.

Since the first generalized eigenvalue problem is associated with the local Lagrange multipliers and local displace-

ment field, the corresponding trace matrix,  $\mathbf{E}_s$  (see Eq. (24)), is a restriction operator [38]. This is a consequence of the common-refinement-based interface discretization with  $\mathbf{N}_u = \mathbf{N}_\lambda, \forall \mathbf{x} \in \Gamma_s$ . Therefore, this *inf-sup* problem is trivially satisfied and not reported here.

The second *inf-sup* problem is related to the discretization spaces between the common-interface variable,  $\Psi$ , and the local Lagrange multipliers,  $\Lambda_s$ . This generalized eigenvalue problem reads [8]:

$$\mathbf{W}\mathbf{Q}^{-1}\mathbf{W}^T\mathbf{m} = \varphi\mathbf{H}\mathbf{m}, \tag{90}$$

where

$$\mathbf{W} = \int_{\Gamma} \mathbf{N}_{\psi}^T \mathbf{N}_{\lambda}^{\Gamma_1+\Gamma_2} d\Gamma, \tag{91}$$

$$\mathbf{Q} = \frac{l_{\Omega}}{\mu} \int_{\Gamma} \left(\mathbf{N}_{\lambda}^{\Gamma_1+\Gamma_2}\right)^T \mathbf{N}_{\lambda}^{\Gamma_1+\Gamma_2} d\Gamma, \tag{92}$$

$$\mathbf{H} = \frac{\mu}{l_{\Omega}} \int_{\Gamma} \mathbf{N}_{\psi}^T \mathbf{N}_{\psi} d\Gamma. \tag{93}$$

Here  $\mu$  is the Lamé parameter and  $l_{\Omega}$  is a typical length of the domain, for instance the diameter of  $\Omega$ . When constructing the matrix  $\mathbf{N}_{\lambda}^{\Gamma_1+\Gamma_2}$ , we take a basis for the whole space  $\mathcal{Z}_h$  (see Eq. (19)), which means taking into account both  $\mathcal{Z}_h^{\Gamma_1}$  and  $\mathcal{Z}_h^{\Gamma_2}$ .

To address the global *inf-sup* condition, we select the example presented in Sect. 5.3 with an inclined common interface. In particular, we compute the generalized eigenvalue problem, Eq. (90), for the sequence of meshes (a) to (d) from Fig. 19. Moreover, we add three more finer discretizations that are generated by repeatedly halving the mesh in (d). Figure 25a shows the convergence of the smallest eigenvalue for meshes (a) to (g). The smallest eigenvalue is  $\varphi_m(g) = 6.932 \times 10^{-2}$  and has been obtained for mesh (g). As clearly shown in Fig. 25a, the smallest eigenvalue is positive and converges to a constant as we refine the meshes. Thus, the *inf-sup* test is satisfied. Note that for mesh density (a), which represents the case of matching grids, we obtain the smallest eigenvalue of  $\varphi_m(a) = 1.0$ . This is a consequence of matching discretization for which the *inf-sup* condition is trivially satisfied [38] and the  $\mathbf{E}_s = \mathbf{B}_s$  matrices are the restriction operators. Finally, we note that the results presented in Sect. 5.2 are on the converged mesh (d). This verifies the analysis in Sect. 5.2.

Next using the mesh density (d) in Fig. 19, we plot the local Lagrange multipliers on the common interface in Fig. 25b. The local Lagrange multipliers are smooth and well converged indicating the stable solution of the common-based discretization using our three-field method.

## References

1. Abedi R, Hawker MA, Haber RB, Matouš K (2010) An adaptive spacetime discontinuous Galerkin method for cohesive models of elastodynamic fracture. *Int J Numer Methods Eng* 81(10):1207–1241
2. Arbogast T, Pencheva G, Wheeler MF, Yotov I (2007) A multi-scale mortar mixed finite element method. *Multiscale Model Simul* 6(1):319–346
3. Becker R, Hansbo P, Stenberg R (2003) A finite element method for domain decomposition with non-matching grids. *ESAIM. Math Model Numer Anal* 37(02):209–225
4. Belytschko T, Mullen R (1978) Mesh partitions of explicit-implicit time integration. In: Bath K, Oden J, Wunderlich W (eds) *Formulations and computational algorithms in finite element analysis*. MIT Press, Cambridge, pp 673–690
5. Belytschko T, Mullen R (1978) Stability of explicit-implicit mesh partitions in time integration. *Int J Numer Methods Eng* 12(10):1575–1586
6. Belytschko T, Yen HJ, Mullen R (1979) Mixed methods for time integration. *Comput Methods Appl Mech Eng* 17:259–275
7. Beneš M, Matouš K (2010) Asynchronous multi-domain variational integrators for nonlinear hyperelastic solids. *Comput Methods Appl Mech Eng* 199(2932):1992–2013
8. Brezzi F, Marini LD (2005) The three-field formulation for elasticity problems. *GAMM-Mitteilungen* 28(2):124–153
9. Brun M, Gravouil A, Combescure A, Limam A (2015) Two feti-based heterogeneous time step coupling methods for newmark and  $\alpha$ -schemes derived from the energy method. *Comput Methods Appl Mech Eng* 283:130–176
10. Combescure A (2002) G.A.: a numerical scheme to couple subdomains with different time-steps for predominantly linear transient analysis. *Comput Methods Appl Mech Eng* 191:1129–1157
11. De Boer A, Van Zuijlen A, Bijl H (2007) Review of coupling methods for non-matching meshes. *Comput Methods Appl Mech Eng* 196(8):1515–1525
12. Falcone M, Ferretti R (1998) Convergence analysis for a class of high-order semi-lagrangian advection schemes. *SIAM J Numer Anal* 35(3):909–940
13. Farhat C, Chen PS, Mandel J (1995) A scalable lagrange multiplier based domain decomposition method for time-dependent problems. *Int J Numer Methods Eng* 38(22):3831–3853
14. Farhat C, Crivelli L, Gérardin M (1995) Implicit time integration of a class of constrained hybrid formulations part i: Spectral stability theory. *Computer methods in applied mechanics and engineering* 125(1):71–107
15. Farhat C, Crivelli L, Roux FX (1994) A transient FETI methodology for large-scale parallel implicit computations in structural mechanics. *Int J Numer Methods Eng* 37(11):1945–1975
16. Farhat C, Lesoinne M, Le Tallec P (1998) Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Comput Methods Appl Mech Eng* 157(1):95–114
17. Farhat C, Roux FX (1991) A method of finite element tearing and interconnecting and its parallel solution algorithm. *Int J Numer Methods Eng* 32(6):1205–1227
18. Faucher V, Combescure A (2003) A time and space mortar method for coupling linear modal subdomains and non-linear subdomains in explicit structural dynamics. *Comput Methods Appl Mech Eng* 192(5):509–533
19. Gander MJ, Vandewalle S (2007) Analysis of the parareal timeparallel timeintegration method. *SIAM J Sci Comput* 29(2):556–578

20. Gates M, Matouš K, Heath MT (2008) Asynchronous multi-domain variational integrators for non-linear problems. *Int J Numer Methods Eng* 76(9):1353–1378
21. Gravouil A, Combescure A (2001) Multi-time-step explicit-implicit method for non-linear structural dynamics. *Int J Numer Methods Eng* 50(1):199–225
22. Gravouil A, Combescure A, Brun M. (2014) Heterogeneous asynchronous time integrators for computational structural dynamics. *Int J Numer Methods Eng*
23. Hauret P, Le Tallec P (2007) A discontinuous stabilized mortar method for general 3d elastic problems. *Comput Methods Appl Mech Eng* 196(49):4881–4900
24. Herry B, Di Valentin L, Combescure A (2002) An approach to the connection between subdomains with non-matching meshes for transient mechanical analysis. *Int J Numer Methods Eng* 55(8):973–1003
25. Hughes T (2000) *The finite element method: linear static and dynamic finite element analysis.*, Dover civil and mechanical engineering, Dover Publications, Mineola
26. Hughes T, Liu W (1978) Implicit-explicit finite elements in transient analysis: implementation and numerical examples. *J Appl Mech* 45:375–378
27. Hughes T, Liu W (1978) Implicit-explicit finite elements in transient analysis: stability theory. *J Appl Mech* 45(2):371–374
28. Hutter K, Johnk K (2004) *Continuum methods of physical modeling.* Springer, New York
29. Jaiman RK, Jiao X, Geubelle PH, Loth E (2005) Assessment of conservative load transfer for fluid-solid interface with non-matching meshes. *Int J Numer Methods Eng* 64(15):2014–2038
30. Jiao X, Heath MT (2004) Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. *Int J Numer Methods Eng* 61(14):2402–2427
31. Jiao X, Heath MT (2004) Overlaying surface meshes, part I: algorithms. *Int J Comput Geom Appl* 14(06):379–402
32. Jiao X, Heath MT (2004) Overlaying surface meshes, part II: Topology preservation and feature matching. *Int J Comput Geom Appl* 14(06):403–419
33. Kane C, Marsden JE, Ortiz M, West M (2000) Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems. *Int J Numer Methods Eng* 49(10):1295–1325
34. Karimi S, Nakshatrala K (2014) On multi-time-step monolithic coupling algorithms for elastodynamics. *J Comput Phys* 273:671–705
35. Kim JS, Arronche L, Farrugia A, Muliana A, Saponara VL (2011) Multi-scale modeling of time-dependent response of smart sandwich constructions. *Compos Struct* 93(9):2196–2207
36. Kruijs J, Zeman J, Gruber P (2013) Model of imperfect interfaces in composite materials and its numerical solution by FETI method. In: Bank R, Holst M, Widlund O, Xu J (eds) *Domain decomposition methods in science and engineering XX.*, Lecture notes in computational science and engineering, Springer, Berlin
37. Maday Y, Turinici G (2005) The parareal in time iterative solver: a further direction to parallel implementation. In: Barth T, Griebel M, Keyes D, Nieminen R, Roose D, Schlick T, Kornhuber R, Hoppe R, Piaux J, Pironneau O, Xu J (eds) *Domain decomposition methods in science and engineering*, vol 40., Lecture notes in computational science and engineering, Springer, Berlin, pp 441–448
38. Magouls F, Roux FX (2006) Lagrangian formulation of domain decomposition methods: a unified theory. *Appl Math Model* 30(7):593–615 (Parallel and vector processing in science and engineering)
39. Mahjoubi N, Gravouil A, Combescure A (2009) Coupling subdomains with heterogeneous time integrators and incompatible time steps. *Comput Mech* 44(6):825–843
40. Mandel J, Dohrmann CR (2003) Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer Linear Algebra Appl* 10(7):639–659
41. Mathew T (2008) *Domain decomposition methods for the numerical solution of partial differential equations*, vol 61., Lecture notes in computational science and engineering, Springer, Berlin
42. Nakshatrala K, Prakash A, Hjelmstad K (2009) On dual Schur domain decomposition method for linear first-order transient problems. *J Comput Phys* 228(21):7957–7985
43. Neal MO, Belytschko T (1989) Explicit-explicit subcycling with non-integer time step ratios for structural dynamic systems. *Comput Struct* 31(6):871–880
44. Oberkampf WL, Trucano TG (2002) Verification and validation in computational fluid dynamics. *Prog Aerosp Sci* 38(3):209–272
45. Park K, Felippa C, Rebel G (2002) A simple algorithm for localized construction of non-matching structural interfaces. *Int J Numer Methods Eng* 53(9):2117–2142
46. Park K, Felippa CA (1998) A variational framework for solution method developments in structural mechanics. *J Appl Mech* 65(1):242–249
47. Park K, Felippa CA (2000) A variational principle for the formulation of partitioned structural systems. *Int J Numer Methods Eng* 47(1–3):395–418
48. Park K, Felippa CA, Ohayon R (2001) Partitioned formulation of internal fluid-structure interaction problems by localized lagrange multipliers. *Comput Methods Appl Mech Eng* 190(2425):2989–3007 (Advances in computational methods for fluid–structure interaction)
49. Park KC, Felippa CA, Gumaste UA (2000) A localized version of the method of lagrange multipliers and its applications. *Comput Mech* 24(6):476–490
50. Park KC, Felippa CA, Rebel G (2002) A simple algorithm for localized construction of non-matching structural interfaces. *Int J Numer Methods Eng* 53(9):2117–2142
51. Prakash A, Hjelmstad KD (2004) A FETI-based multi-time-step coupling method for Newmark schemes in structural dynamics. *Int J Numer Methods Eng* 61(13):2183–2204
52. Prakash A, Tacioglu E, Hjelmstad KD (2014) Computationally efficient multi-time-step method for partitioned time integration of highly nonlinear structural dynamics. *Comput Struct* 133:51–63
53. Quarteroni A, Valli A (1999) *Domain decomposition methods for partial differential equations.* Numerical mathematics and scientific computation. Oxford University Press, New York
54. Radu F, Pop IS, Knabner P (2004) Order of convergence estimates for an euler implicit, mixed finite element discretization of richards' equation. *SIAM J Numer Anal* 42(4):1452–1478
55. Roache PJ (1998) *Verification and validation in computational science and engineering.* Hermosa, Albuquerque
56. Ross MR, Sprague MA, Felippa CA, Park K (2009) Treatment of acoustic fluid-structure interaction by localized lagrange multipliers and comparison to alternative interface-coupling methods. *Comput Methods Appl Mech Eng* 198(9):986–1005
57. Scovazzi G, Love E, Shashkov M (2008) Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: theoretical framework and two-dimensional computations. *Comput Methods Appl Mech Eng* 197(912):1056–1079
58. Smith B, Bjorstad P, Gropp W (1996) *Domain decomposition: parallel multilevel methods for elliptic partial differential equations.* Cambridge University Press, New York
59. Smolinski P, Sleith S (1992) Explicit multi-time step methods for structural dynamics. In: *New methods in transient analysis*, pp. 1–4. ASME. PVP-Vol. 246 /AMD-Vol. 143
60. Smolinski P, Sleith S, Belytschko T (1996) Stability of an explicit multi-time step integration algorithm for linear structural dynamics equations. *Comput Mech* 18(3):236–244

61. Sousedik B, Sístek J, Mandel J (2013) Adaptive-multilevel BDDC and its parallel implementation. *Computing* 95(12):1087–1119
62. Souza F, Allen D, Kim YR (2008) Multiscale model for predicting damage evolution in composites due to impact loading. *Compos Sci Technol* 68(13):2624–2634
63. Toselli A, Widlund O (2005) Domain decomposition methods: algorithms and theory, vol 34., Computational mathematicsSpringer, Berlin
64. Wiberg NE, Li X (1993) A post-processing technique and an a posteriori error estimate for the newmark method in dynamic analysis. *Earthq Eng Struct Dyn* 22(6):465–489
65. Zeng LF, Wiberg NE, Li X, Xie Y (1992) Posteriori local error estimation and adaptive time-stepping for newmark integration in dynamic analysis. *Earthq Eng Struct Dyn* 21(7):555–571