Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.023071

ARTICLE

Towards a Unified Single Analysis Framework Embedded with Multiple Spatial and Time Discretized Methods for Linear Structural Dynamics

David Tae and Kumar K. Tamma*

Department of Mechanical Engineering, University of Minnesota, Twin Cities, Minneapolis, 55455, USA

*Corresponding Author: Kumar K. Tamma. Email: ktamma@umn.edu

Received: 07 April 2022 Accepted: 01 July 2022

ABSTRACT

We propose a novel computational framework that is capable of employing different time integration algorithms and different space discretized methods such as the Finite Element Method, particle methods, and other spatial methods on a single body sub-divided into multiple subdomains. This is in conjunction with implementing the well known Generalized Single Step Single Solve (GS4) family of algorithms which encompass the entire scope of Linear Multistep algorithms that have been developed over the past 50 years or so and are second order accurate into the Differential Algebraic Equation framework. In the current state of technology, the coupling of altogether different time integration algorithms has been limited to the same family of algorithms such as the Newmark methods and the coupling of different algorithms usually has resulted in reduced accuracy in one or more variables including the Lagrange multiplier. However, the robustness and versatility of the GS4 with its ability to accurately account for the numerical shifts in various time schemes it encompasses, overcomes such barriers and allows a wide variety of arbitrary implicit-implicit, implicit-explicit, and explicit-explicit pairing of the various time schemes while maintaining the second order accuracy in time for not only all primary variables such as displacement, velocity and acceleration but also the Lagrange multipliers used for coupling the subdomains. By selecting an appropriate spatial method and time scheme on the area with localized phenomena contrary to utilizing a single process on the entire body, the proposed work has the potential to better capture the physics of a given simulation. The method is validated by solving 2D problems for the linear second order systems with various combination of spatial methods and time schemes with great flexibility. The accuracy and efficacy of the present work have not yet been seen in the current field, and it has shown significant promise in its capabilities and effectiveness for general linear dynamics through numerical examples.

KEYWORDS

Time integration; structural dynamics; multiple scale and multiple methods; ordinary differential equations; differential algebraic equations

1 Introduction

As numerical analysis is utilized in the various fields and applications, the desire for analyzing complex problems more efficiently has grown. The Finite Element Method (FEM) is one of the most widely used numerical analysis methods. While it features simplicity and versatility, it is susceptible to numerical errors such as shear locking and complexity in adopting it for large deformation problems and the like.

Concurrently, the strong form based particle methods has been gaining interest in the field of numerical simulations as the particle methods have the capability and flexibility for handling large distortion, crack propagation and free surface detecting, etc. Some of the notable strong form particle methods, Smoothed Particle Hydrodynamics (SPH) [1–4] and Moving Particle Semi-implicit Method (MPS) [5,6], have been used to solve partial differential equations. Later, Silling et al. proposed a nonlocal extension of classical continuum mechanics called peridynamics (PD) [7,8], which represents spatial derivatives of field quantities in terms of internal interactions between points/particles via integrals involving differences. The peridynamics model shows a comprehensive ability to solve linear and nonlinear second order system problems and simulate structural dynamics and molecular dynamics as well as multi-scale problems [9]. More recently, a mixed form representation of the strong form particle method, where multiple primary variables are solved at once, is presented in [10]. Other related works appear in references [11–14].

As problems become more complex, controlling the spatial discretization locally is a desirable feature. While an area where fast dynamics are presented is discretized with high spatial resolutions, a coarser mesh may be sufficient in the majority of the structure. In addition, the selection of spatial methods in different parts of a body can also be an effective strategy such as by utilizing the FEM for the majority of a body, and applying particle based methods only on an area where localized phenomena such as crack or fracture may occur.

One of the ways to locally control spatial discretization is dividing a body into regions of subdomains in which different methods can be applied in each subdomain. The challenge in using subdomains is handling the interface condition between the subdomains, and over the years, numerous coupling methods have been developed. Interface coupling is largely divided into two categories: overlapping and non-overlapping domain coupling. In the case of the overlapping domain decomposition, a portion of the subdomains are overlapped and the shape function similar to the framework of the FEM is used to constrain the common Degrees of Freedom (DOFs). However, the large difference in each of the subdomain's stiffness may lead to the global stiffness matrix to be ill-conditioned, as noted by Orsini et al. in [15]. Moreover, Metsis et al. [16] implied that the weak form based domain decomposition methods may be more prone to this issue due to the implementation of the Gaussian integral.

On the other hand, the non-overlapping technique is more straight forward where the subdomains are only connected by an edge or a surface in 2D or 3D, respectably, and are constrained by the Lagrange multiplier at the interface. At the interface, usually the nodes/particles are overlapped; however, Park et al. [17] and Brezzi et al. [18] introduced an approach where the subdomains are not directly interacting but through an intermediate interface. This approach allows more flexibility in meshing as one-to-one matching of the nodes is not required.

Implementing the different spatial discretization methods for localized phenomena has been a study which has been explored for many years. The coupling of the FEM with molecular dynamics via a bridging domain which enforces compatibility by utilizing the Lagrange multiplier is proposed by Xiao et al. [19]. Alternately, the Discrete Element Method (DEM) [20] is a widely used meshfree method in the rock engineering simulation field and various FEM-DEM coupling methods have been developed [21,22]. Utilizing the peridynamics method for localized phenomena such as crack propagation is also a topic with extensive studies including coupling the FEM and the PD [23–25]. In this work, one to one node/particle matching is used to simply demonstrate the concept of coupling different spatial methods and different time integration schemes unlike state of art which is limited to just the spatial methods. We couple the FEM and particle methods with different time schemes for each subdomain and such features are not feasible in the state of art.

There also has been high interest in coupling subdomains for transient analysis. Gravouil et al. presented the GC method in [26], employing different time increments in subdomains using the limited Newmark family of algorithms. However, it was later discovered that artificial numerical dissipation arises at the interface resulting in the loss of energy and loss of second order accuracy in the GC method.

Similarly, Pegon extended the GC method and proposed the PM method which employs an inter-field parallel solution procedure to couple different subdomains [27,28]. Prakash et al. [29] developed the PH method in order to alleviate the GC method's energy issue by solving the interface at a large time scale. The drawback of the PH method was described by Karimi et al. who stated in [30], that the PH method can only apply to two subdomains. Karimi et al. [30] developed the subdomain DAE framework for the Newmark family of algorithms. A noteworthy aspect of this work is that the constrained continuous system in time is in fact a system of differential algebraic equations rather than ordinary differential equations in time which most coupling methods have been based upon. Nakshatrala et al. later applied the DAE approach to the Finite Element Tearing and Interconneting (FETI) method based domain decomposition to solve time dependent first order systems with the fourth order Nakshatrala et al. [31]. More recently, Shimada [32] integrated the Generalized Single Step Single Solve (GS4) family of algorithms with the subdomain DAE framework using the FEM. And, Maxam et al. [33] proposed an unified approach of the DAE-GS4 framework on thermoelastic problems using the FEM. There exists interest in coupling different subdomains for transient analysis. However, the current algorithm technology is very limited and does not permit features to interface a wide variety of time algorithms and spatial methods. And, this is the focus of the advances and contributions of this paper.

In this work, the subdomain DAE framework is particularly advanced to the Generalized Single Step Single Solve (GSSSS or GS4) framework and family of algorithms developed by Tamma et al. [34–37]. The versatility of the GSSSS family of algorithms in the treatment of numerical dissipation, overshoot and ease of adapting to explicit methods, proves extremely useful to the subdomain framework. In addition, the coupling of the FEM based on the weak form, and the generalized particle method (GPS) based on the strong form is also presented in this work to demonstrate integrating altogether different spatial methods as well. The coupling of different spatial and time discretization methods may lead to more accurate computation as it allows the selection of the appropriate method for a specific situation. The novel GS4 framework and its family of algorithms is a general purpose structure. It encompasses the entire class of Linear Multistep (LMS) methods including those that have been developed over the past 50 years or so and in addition, also includes new and advanced optimal designs of algorithms, in 1st/2nd order systems that are second order time accurate and used for commercial software to enable practical real word applications to be conducted. In addition, under one umbrella, the frameworks inherit and have new and optimal algorithms and designs in addition to existing methods such as Crank-Nicolson, Gear's BDF, etc. for first order systems; and new and optimal choices in second order systems as well in contrast to existing ones such as the Newmark family, HHT, WBZ and the like. A particularly unique feature is that the framework permits interfacing a wide variety of implicit-implicit, implicit-explicit, and explicit-explicit algorithm designs in a single analysis unlike the existing methods which are limited and not capable of doing so; in particular coupling of different implicit-implicit algorithms with/without controllable numerical dissipation which is not feasible in the current state-of-the-art. This is a significant breakthrough and is capitalized herein. The generality, flexibility and robustness of the present computational framework which embraces subdomain and time integration algorithms and differing spatial methods fused into one is not possible with today's traditional methods and technology that shows limitations such as reduced order accuracy, consistency problems and the like in various field variables such as displacement, velocity, acceleration and Lagrange multiplier. The present developments provide a wide variety of choices to the analyst. In this work, we consistently preserve and maintain second order time accuracy in all variables such as displacement, velocity and acceleration including the Lagrange multipliers. In this paper, the FEM and GPS are first reviewed to explain the basics in integrating different spatial methods as well as different time integration methods in a single analysis. Then in the following section, the incorporation of the GS4 family of algorithms with the DAE framework is investigated and validated through various numerical examples.

2 Finite Element Method

A brief overview of the finite element method is reviewed in this section for better understanding of the various numerical examples and its methodology that are presented in a later section. In addition, the objective is to get clarity of the computations involved in the calculations of the FEM integrals and particle method integrals.

2.1 Finite Element Approximations

In the finite element method, a body of domain Ω is divided into numerous regions called elements formed by connecting points known as nodes. The finite dimensional field variable such as the displacement u_h is approximated within each element by the basis functions. The most common basis functions have the Kronecker delta δ_{ij} property so that the nodal values $\hat{u}$ can be accurately represented. The integration of the finite dimension is performed by summing up the integrals over each element domain.

$$\int_{\Omega} g(u)d\Omega = \sum_{j=1}^{N_h} \int_{\Omega_j} g(u_h)d\Omega_j = \sum_{j=1}^{N_{elem}} \int_{\Omega_j} g\left(N_i\hat{u}_i\right) d\Omega_j \tag{1}$$

where N_i are the elemental basis functions which varies for different types of element. The basis functions for a commonly used 2D bilinear quadrilateral are given as follows:

$$u = N_i\hat{u}_i = \begin{bmatrix} N_1(\xi,\eta) & N_2(\xi,\eta) & N_3(\xi,\eta) & N_4(\xi,\eta) \end{bmatrix} \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T \tag{2}$$

where

$$N_1(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta)$$

$$N_2(\xi,\eta) = \frac{1}{4}(1+\xi)(1-\eta)$$

$$N_3(\xi,\eta) = \frac{1}{4}(1+\xi)(1+\eta)$$

$$N_4(\xi,\eta) = \frac{1}{4}(1-\xi)(1+\eta) \tag{3}$$

Quadrilateral elements with two Gauss points in each x and y direction will be used throughout in the numerical example section.

2.2 Weak Form Formulation for FEM

The equilibrium equation expressed in terms of the displacement can be written as (shown for 1D as a simple illustration)

$$\frac{\partial \sigma}{\partial x} + b_x = \rho \frac{\partial^2 u}{\partial t^2} \tag{4}$$

Next we introduce an arbitrary function $w(x)$ that is defined in the domain Ω and multiply the equilibrium equation.

$$g\left(w, u, \sigma_x\right) = w\left(x\right)\left(\rho \frac{\partial^2 u}{\partial t^2} - \frac{\partial \sigma}{\partial x} - b_x\right) = 0 \tag{5}$$

Then we integrate over the domain to obtain an integral form that is set to zero.

$$\mathbf{G}\left(w, u, \sigma_x\right) = \int_\Omega w\left(x\right)\left(\rho \frac{\partial^2 u}{\partial t^2} - \frac{\partial \sigma}{\partial x} - b_x\right) dx = 0 \tag{6}$$

The stress term is then integrated by parts as

$$-\int_\Omega w\left(x\right) \frac{\partial \sigma_x}{\partial x} dx = \int_\Omega \frac{\partial w}{\partial x} \sigma_x dx - w\left(x\right) n_x \sigma_x |_\Gamma \tag{7}$$

where Γ is the boundary of Ω and n_x is the outward pointing normal to the boundary. By substitution of Eq. (7) into Eq. (6), we get the weak form of the equilibrium as

$$\mathbf{G}\left(w, u, \sigma_x\right) = \int_\Omega w\left(x\right)\left(\rho \frac{\partial^2 u}{\partial t^2} - b_x\right) dx + \int_\Omega \frac{\partial w}{\partial x} \sigma_x dx - w t_x |_{\Gamma_u} - w \bar{t}_x |_{\Gamma_t} = 0 \tag{8}$$

Introduce the constitutive equation to obtain governing equation based on displacement, and we get

$$\mathbf{G}\left(w, u\right) = \int_\Omega w\left(x\right)\left(\rho \frac{\partial^2 u}{\partial t^2} - b_x\right) dx + \int_\Omega \frac{\partial w}{\partial x} E \frac{\partial u}{\partial x} dx - w \bar{t}_x |_{\Gamma_t} = 0 \tag{9}$$

It is worth noting that we removed t_x by setting $w = 0$ where the displacements are specified.

The resulting weak formulation in space is given as

$$\rho \int_\Omega w \frac{\partial^2 u}{\partial t^2} d\Omega + \int_\Omega \nabla w E \nabla u \, d\Omega = \int_\Omega w b d\Omega + \int_{\partial\Omega \backslash \Gamma_u} w \bar{t} d\Gamma \tag{10}$$

and limiting the trial and test spaces to high fidelity finite element spaces, we get

$$\rho \int_\Omega w_h \frac{\partial^2 u_h}{\partial t^2} d\Omega + \int_\Omega \nabla w_h E \nabla u_h \, d\Omega = \int_\Omega w_h b d\Omega + \int_{\partial\Omega \backslash \Gamma_u} w_h \bar{t} d\Gamma \tag{11}$$

Setting $u_h = N_i \hat{u}_{hi}$ and $w_h = N_i \hat{w}_{hi}$, for the Ritz-Galerkin approximation, we have

$$\rho \hat{w}_{hi}^T \int_\Omega N^T N \, \ddot{\hat{u}}_{hi} \, d\Omega + \hat{w}_{hi}^T \int_\Omega \nabla N^T E \nabla N \, \hat{u}_{hi} \, d\Omega = \hat{w}_{hi}^T \int_\Omega N^T b \, d\Omega + \hat{w}_{hi}^T \int_\Gamma N^T \bar{t} d\Gamma \tag{12}$$

The above equation is valid for any $\hat{w}_h$. Therefore we may write

$$\rho \int_\Omega N^T N \, \ddot{\hat{u}}_{hi} \, d\Omega + E \int_\Omega \nabla N^T \nabla N \, \hat{u}_{hi} \, d\Omega = \int_\Omega N^T b \, d\Omega + \int_\Gamma N^T \bar{t} d\Gamma \tag{13}$$

leading to

$$\mathbf{M} \ddot{\hat{u}}_{hi} + \mathbf{K} \hat{u}_{hi} = \hat{\mathbf{F}} \tag{14}$$

$$\mathbf{M} = \rho \int_{\Omega} N^T N \, d\Omega \tag{15}$$

$$\mathbf{K} = E \int_{\Omega} \nabla N^T \nabla N \, d\Omega \tag{16}$$

$$\hat{\mathbf{F}} = \int_{\Omega} N^T b \, d\Omega + \int_{\Gamma} N^T \bar{t} d\Gamma \tag{17}$$

3 Generalized Particle Method

3.1 Gradient and Divergence of Classical Physical Field

A physical field can be thought of as the assignment of a physical quantity at each point in space and time. Consider a domain in Euclidean space, $\mathbb{E}$. Let $\mathbf{X} \in \Omega_0 \subset \mathbb{E}$ denotes the geometric position vector of a certain position in the domain and $\underline{\mathbf{X}} \in \Omega_0 \subset \mathbb{E}$ denotes another point which is close to the position $\mathbf{X}$ as shown in Fig. 1a. Therefore, any physical quantity at each point in space and time can be defined as $\boldsymbol{\varphi}(\mathbf{X}, t) : \Omega_0 \times \mathbb{I} \rightarrow \mathbb{E}$. Define $\Omega_{\mathbf{X}} \subset \Omega_0$ as a domain where the center point position vector $\mathbf{X}$ is in, then we have

$$\boldsymbol{\varphi}\left(\mathbf{X}, t\right) = \lim_{V_{\Omega_{\mathbf{X}}} \rightarrow 0} \frac{\int_{V_{\Omega_{\mathbf{X}}}} \boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) dV}{V_{\Omega_{\mathbf{X}}}} \tag{18}$$

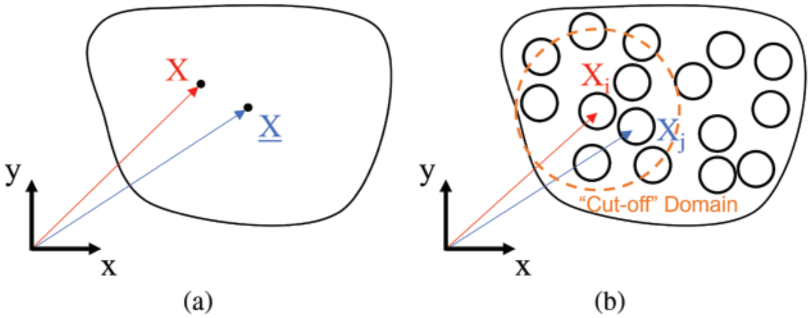


Figure 1: (a) Continuous domain with position vector $\mathbf{X}$ and $\underline{\mathbf{X}}$, (b) Discretized domain with position vectors $\mathbf{X}_i$ and $\mathbf{X}_j$

The weighted residual method has been widely used in solving different kinds of ordinary/partial differential equations, in particular, using the Galerkin method, which is the fundamental theory of the finite element method. In this work, we exploit the weighted residual method directly to the Taylor series expansion, such that a generalized approach of developing particle based methods can be explored.

In mathematics, the Taylor series expansion is a representation of a function as an infinite sum of terms that are calculated from the values of the function's derivatives at a single point. The generalized Taylor series of a general physical quantity $\boldsymbol{\varphi}_t\left(\underline{\mathbf{X}}\right)$ about $\mathbf{X}$ at a fixed time t yields

$$\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) = \boldsymbol{\varphi}\left(\mathbf{X}, t\right) + \Lambda_1 \nabla \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\left(\underline{\mathbf{X}} - \mathbf{X}\right) + \Lambda_2 \boldsymbol{H}\left(\mathbf{X}, t\right)\left(\underline{\mathbf{X}} - \mathbf{X}\right)\left(\underline{\mathbf{X}} - \mathbf{X}\right) + \ldots + \mathcal{O}\left(\|\underline{\mathbf{X}} - \mathbf{X}\|^n\right) \tag{19}$$

where $\nabla \boldsymbol{\varphi}(\mathbf{X}, \mathbf{t})$ is a vector when $\varphi(\mathbf{X}, t)$ is a scalar value and is a second order tensor when $\boldsymbol{\varphi}(\mathbf{X}, t)$ is a vector value; and $\Lambda_i \in \mathbb{R}(i = 1, 2, 3 \ldots .)$ is the algorithmic parameters.

Omitting the higher order terms from Eq. (19), we have the first order Taylor series approximation

$$\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right) \cong \Lambda_1 \nabla\boldsymbol{\varphi}\left(\mathbf{X}, t\right)\left(\underline{\mathbf{X}} - \mathbf{X}\right) \tag{20}$$

where $\Lambda_1 = 1$. Define a residual with regards to Eq. (20), $\boldsymbol{R}$, as

$$\boldsymbol{R} = \boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right) - \nabla\boldsymbol{\varphi}\left(\mathbf{X}, t\right)\left(\underline{\mathbf{X}} - \mathbf{X}\right) \tag{21}$$

According to the standard weighted residual method, we introduce a weight function, $\boldsymbol{C}$[1], and minimize the residual. Eq. (21) is multiplied by the weight function $\boldsymbol{C}$ and integrated over the domain $\Omega_{\mathbf{X}}$, which then yields

$$\int_{\Omega_{\mathbf{X}}} \boldsymbol{RC}dV = \mathbf{0} \tag{22}$$

It is worth noting that the domain is the cut-off influence domain of the position $\boldsymbol{X}$, and the residual is a local value. The weight function $\boldsymbol{C}$ provides a wide scope of approximations for constructing the derivative definitions used in various particle-based methods and new approach of particle methods as well.

Gradient Recall the residual $\boldsymbol{R}$ in Eq. (21) and the weighted residual, $\boldsymbol{RC}$, with respect to a vector property $\boldsymbol{\varphi}$ is

$$(\boldsymbol{RC})_{\mathbf{X}\underline{\mathbf{X}}} = \left(\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right) \otimes \boldsymbol{C} - \nabla\boldsymbol{\varphi}\left(\mathbf{X}, t\right)_{\mathbf{X}\underline{\mathbf{X}}}\left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{C} \tag{23}$$

where $\mathbf{C}$ is a vector with the size of geometric dimension of a problem; for example, $\boldsymbol{C}$ is a two by one vector in 2D problems. We postulate that Eq. (23) is the exact relationship between the point $\mathbf{X}$ and one of its neighboring point $\underline{\mathbf{X}}$ located in the domain $\Omega_{\mathbf{X}}$ in terms of the weighted residual, $(\boldsymbol{RC})_{\mathbf{X}\underline{\mathbf{X}}}$. Therefore, by integrating Eq. (23) over the domain $\Omega_{\mathbf{X}}$, the following formulations can be obtained as

$$\int_{V_{\Omega_{\mathbf{X}}}} \boldsymbol{RC}dV = \int_{V_{\Omega_{\mathbf{X}}}} \left[\left(\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right) \otimes \boldsymbol{C}\right] - \nabla\boldsymbol{\varphi}\left(\mathbf{X}, t\right)_{\mathbf{X}\underline{\mathbf{X}}}\left[\left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{C}\right] dV = 0 \tag{24}$$

By treating $\left(\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right) \otimes \boldsymbol{C}$ and $\nabla\boldsymbol{\varphi}\left(\mathbf{X}, \mathbf{t}\right)\left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{C}$ in the integral presented above as field properties in the domain $\Omega_{\mathbf{X}}$, the following formulations can be obtained as

$$\frac{1}{V_{\Omega_{\mathbf{X}}}} \int_{V_{\Omega_{\mathbf{X}}}} \left[\left(\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right) \otimes \boldsymbol{C}\right] dV = \frac{1}{V_{\Omega_{\mathbf{X}}}} \int_{V_{\Omega_{\mathbf{X}}}} \nabla\boldsymbol{\varphi}\left(\mathbf{X}, t\right)_{\mathbf{X}\underline{\mathbf{X}}}\left[\left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{C}\right] dV \tag{25}$$

Hence, the gradient of a vector can be obtained as

$$\nabla\boldsymbol{\varphi}\left(\mathbf{X}, \mathbf{t}\right) = \lim_{V_{\Omega_{\mathbf{X}}} \to 0} \left(\int_{V_{\Omega_{\mathbf{X}}}} \left(\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right) \otimes \boldsymbol{C}dV\right) \lim_{V_{\Omega_{\mathbf{X}}} \to 0} \left(\int_{V_{\Omega_{\mathbf{X}}}} \left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{C}dV\right)^{-1} \tag{26}$$

The proposed gradient operator has the ability to recover the $'\nabla'$ in the Moving Particle Semi-implicit (MPS) method, the deformation gradient in the state-based peridynamics as well as in the corrected Smoothed Particle Hydrodynamics (SPH) method, by selecting the appropriate corresponding $\boldsymbol{C}$.

[1] In this work, the weight function used in the weighted residual method is defined as $\boldsymbol{C}$, and the weight function for describing the influence generated from neighboring particle is defined as W.

Remark 3.1. By selecting the arbitrary $\boldsymbol{C} = \boldsymbol{n}_{ij}$, the Gradient in terms of the original MPS method can be derived from the proposed approach as well. Substituting $\boldsymbol{C} = \boldsymbol{n}_{ij}$ in the discretized system yields,

$$\nabla\varphi_i = \left[\sum_{j\in\Omega_{\mathbf{X}_i}} \left(\mathbf{X}_j - \mathbf{X}_i\right) \otimes \boldsymbol{n}_{ij} V_j\right]^{-1} \left[\sum_{j\in\Omega_{\mathbf{X}_i}} \left(\varphi_j - \varphi_i\right) \boldsymbol{n}_{ij} V_j\right] \tag{27}$$

Let $\mathbf{A} = \left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{n}_{ij}$ and it can be replaced by its normalized weighted average value within an infinitesimal domain via exploiting Eq. (18),

$$\mathbf{A}_{\mathbf{X}\underline{\mathbf{X}}avg} = \frac{1}{V_{\Omega_X}} \lim_{V_\Omega\to 0} \int_{\Omega_X} \left[\left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{C}\right] dV \tag{28}$$

Discretizing into particles, and applying $\boldsymbol{n}_{ij} = (\boldsymbol{X}_j - \boldsymbol{X}_i)/\left|\left|\boldsymbol{X}_j - \boldsymbol{X}_i\right|\right|$, it becomes

$$\begin{aligned}\mathbf{A}_{ij} &= \frac{1}{V_{\Omega_{\mathbf{X}_i}}} \sum_{j\in\Omega_{\mathbf{X}_i}} \left(\boldsymbol{n}_{ij} \otimes \boldsymbol{n}_{ij}\right) V_j \left(\left|\left|\mathbf{X}_j - \mathbf{X}_i\right|\right|\right) \\ &= \underbrace{\frac{\sum_{j\in\Omega_{\mathbf{X}_i}} \left(\boldsymbol{n}_{ij} \otimes \boldsymbol{n}_{ij}\right) W_{ij} V_j}{V_{\Omega_{\mathbf{X}_i}} \sum_{j\in\Omega_{\mathbf{X}_i}} W_{ij}}}_{\textit{Normalized Weighted Average } \mathbf{A}} \left(\left|\left|\mathbf{X}_j - \mathbf{X}_i\right|\right|\right)\end{aligned} \tag{29}$$

Thus, Eq. (27) can be written as

$$\nabla\varphi_{ij} = \left[\frac{\sum_{k\in\Omega_{\mathbf{X}_j}} \left(\boldsymbol{n}_{jk} \otimes \boldsymbol{n}_{jk}\right) W_{jk} V_k}{V_{\Omega_{\mathbf{X}_j}} \sum_{k\in\Omega_{\mathbf{X}_j}} W_{jk}}\right]^{-1} \frac{\left(\varphi_j - \varphi_i\right) \boldsymbol{n}_{ij}}{\left|\left|\mathbf{X}_j - \mathbf{X}_i\right|\right|} \tag{30}$$

Gradient Operator:

Introducing the weighting function W_{ij} and supposing $V_i = V_j$, $V_{\Omega_{\mathbf{X}_i}} = V_{\Omega_{\mathbf{X}_j}}$ yields

$$\begin{aligned}\nabla\varphi_i &= \frac{1}{V_{\Omega_{\mathbf{X}_i}} \sum_{j\in\Omega_{\mathbf{X}_i}} W_{ij}} \sum_{j\in\Omega_{\mathbf{X}_i}} \underbrace{\left[\frac{\sum_{k\in\Omega_{\mathbf{X}_j}} \left(\boldsymbol{n}_{jk} \otimes \boldsymbol{n}_{jk}\right) W_{jk} V_k}{V_{\Omega_{\mathbf{X}_j}} \sum_{k\in\Omega_{\mathbf{X}_j}} W_{jk}}\right]^{-1}}_{\mathbf{A}_{jk}} \frac{\left(\varphi_j - \varphi_i\right) \boldsymbol{n}_{ij}}{\left|\left|\mathbf{X}_j - \mathbf{X}_i\right|\right|} W_{ij} V_j \\ &= \frac{1}{\sum_{j\in\Omega_{\mathbf{X}_i}} W_{ij}} \sum_{j\in\Omega_{\mathbf{X}_i}} \left[\frac{\sum_{k\in\Omega_{\mathbf{X}_j}} \left(\boldsymbol{n}_{jk} \otimes \boldsymbol{n}_{jk}\right) W_{jk}}{\sum_{k\in\Omega_{\mathbf{X}_j}} W_{jk}}\right]^{-1} \frac{\left(\varphi_j - \varphi_i\right) \boldsymbol{n}_{ij}}{\left|\left|\mathbf{X}_j - \mathbf{X}_i\right|\right|} W_{ij}\end{aligned} \tag{31}$$

It is worth noting that for a general particle within a uniformly distributed system, the weighted average matrix $\boldsymbol{A}_{jk}$ is a diagonal matrix. Consequently, $\boldsymbol{A}_{jk} = d\mathbf{I}$, where $\mathbf{I}$ is identity matrix and d equals to the dimensional number. Therefore, Eq. (31) recovers the gradient calculation in the MPS method, and can be rewritten as

$$\nabla\varphi_i = \frac{d}{n_0} \sum_{j\in\Omega_{\mathbf{X}_i}} \frac{\left(\varphi_j - \varphi_i\right)}{\left|\left|\mathbf{X}_j - \mathbf{X}_i\right|\right|^2} \left(\mathbf{X}_j - \mathbf{X}_i\right) W_{ij} \tag{32}$$

where $n_0 = \sum_j W_{ij}$.

Divergence By representing the divergence model as gradient operator, ∇, dot product with a physical quantity, $\boldsymbol{\varphi}$, the divergence model can be written as following

$$\nabla \cdot \boldsymbol{\varphi}\left(\mathbf{X}, t\right) = \nabla^{T}\boldsymbol{\varphi} = \lim_{V_{\Omega_{\mathbf{X}}} \to 0}\left(\int_{V_{\Omega_{\mathbf{X}}}} \left(\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right) \cdot \boldsymbol{C} dV\right) \lim_{V_{\Omega_{\mathbf{X}}} \to 0}\left(\int_{V_{\Omega_{\mathbf{X}}}} \left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{C} dV\right)^{-1} \tag{33}$$

The discretized formulations is listed as following

$$\nabla \cdot \boldsymbol{\varphi}_{i} = \left[\sum_{j \in \Omega_{\mathbf{X}_i}} \left(\boldsymbol{\varphi}_{j} - \boldsymbol{\varphi}_{i}\right) \cdot \boldsymbol{C}_{j} W_{ij} V_{j}\right]\left[\sum_{j \in \Omega_{\mathbf{X}_i}} \left(\mathbf{X}_{j} - \mathbf{X}_{i}\right) \otimes \boldsymbol{C}_{j} W_{ij} V_{j}\right]^{-1} \tag{34}$$

Laplacian The definition of the Laplacian can be written as

$$\nabla^{2}\boldsymbol{\varphi}(\mathbf{X}, t) = \nabla \cdot \nabla\boldsymbol{\varphi}(\mathbf{X}, t) \tag{35}$$

In order to obtain the Laplacian relationship between the center point $\mathbf{X}$ and its neighboring point $\underline{\mathbf{X}}$, we reconstruct the gradient between two points via taking advantage of Eq. (26). One may notice that it is convenient to find the gradient from Eq. (23) by setting the $(\boldsymbol{RC})_{\mathbf{X}\underline{\mathbf{X}}}$ as 0, which yields

$$\nabla\varphi\left(\mathbf{X}, t\right)_{\mathbf{X}\underline{\mathbf{X}}} = \left[\left(\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right) \otimes \boldsymbol{C}\right]\left[\left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{C}\right]^{-1} \tag{36}$$

However, the rank of matrix $\left[\left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{C}\right]$ is 1, which is not available for the calculation of the inverse when the gradient is carried over from the vector gradient calculation. To circumvent the issue, we introduce $\mathbf{A} = \left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{C}$. Then it can be replaced by its average value within an infinitesimal domain via exploiting Eq. (18),

$$\mathbf{A}_{\mathbf{X}\underline{\mathbf{X}}avg} = \frac{1}{V_{\Omega_{\mathbf{X}}}} \lim_{V_{\Omega_{X}} \to 0} \int_{\Omega_{\mathbf{X}}} \left[\left(\underline{\mathbf{X}} - \mathbf{X}\right) \otimes \boldsymbol{C}\right] dV \tag{37}$$

Therefore, an alternative gradient of a vector is given by taking advantage of the average property, Eq. (37).

$$\nabla\boldsymbol{\varphi}\left(\mathbf{X}, t\right)_{\mathbf{X}\underline{\mathbf{X}}} = \left[\left(\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right) \otimes \boldsymbol{C}\right]\left(\mathbf{A}_{\mathbf{X}\underline{\mathbf{X}}avg}\right)^{-1} \tag{38}$$

Consequently, the Laplacian can be obtained via combining Gauss's Theorem and the gradient formulations.

$$\left(\nabla \cdot \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right)_{\mathbf{X}\underline{\mathbf{X}}} V_{\underline{\mathbf{X}}} = \boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) \cdot \boldsymbol{n}_{\mathbf{X}\underline{\mathbf{X}}} S_{\underline{\mathbf{X}}} \tag{39}$$

where $\underline{S} = \frac{\underline{V}}{||\underline{\mathbf{x}} - \mathbf{x}||}$. As will be explained in the later section, we can also replace $\boldsymbol{n}_{\mathbf{X}\underline{\mathbf{X}}}$ with $\mathbf{C}$. Substitution of $\underline{S}$ into Eq. (39) yields

$$\left(\nabla \cdot \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right)_{\mathbf{X}\underline{\mathbf{X}}} \left|\left|\underline{\mathbf{X}} - \mathbf{X}\right|\right| = \boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) \cdot \boldsymbol{C} \tag{40}$$

Therefore, the Laplacian of $\boldsymbol{\varphi}(\mathbf{X}, t)$ becomes

$$\left(\nabla \cdot \nabla\boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right)_{\mathbf{X}\underline{\mathbf{X}}} \left|\left|\underline{\mathbf{X}} - \mathbf{X}\right|\right| = \left[\left(\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right) \otimes \boldsymbol{C}\right]\left(\mathbf{A}_{\mathbf{X}\underline{\mathbf{X}}avg}\right)^{-1} \cdot \boldsymbol{C} \tag{41}$$

Hence the following Laplacian formulations can be obtained as

$$\nabla^{2}\boldsymbol{\varphi}\left(\mathbf{X}, t\right) = \nabla \cdot \nabla\boldsymbol{\varphi}\left(\mathbf{X}, t\right) = \lim_{V_{\Omega_{\mathbf{X}}} \to 0} \frac{1}{V_{\Omega_{\mathbf{X}}}} \int_{\Omega_{\mathbf{X}}} \frac{\left[\left(\boldsymbol{\varphi}\left(\underline{\mathbf{X}}, t\right) - \boldsymbol{\varphi}\left(\mathbf{X}, t\right)\right) \otimes \boldsymbol{C}\right] \mathbf{A}_{\mathbf{X}\underline{\mathbf{X}}avg}^{-1} \cdot \boldsymbol{C}}{\left|\left|\underline{\mathbf{X}} - \mathbf{X}\right|\right|} dV \tag{42}$$

The discretized formulations is listed as following:

$$\nabla^2 \boldsymbol{\varphi}_i = \frac{1}{V_{\Omega_{\mathbf{X}_i}}} \sum_{j \in \Omega_{\mathbf{X}_i}} \frac{\left[(\boldsymbol{\varphi}_j - \boldsymbol{\varphi}_i) \otimes \boldsymbol{C}_j\right] \mathbf{A}_{ij_{avg}}^{-1} \cdot \boldsymbol{C}_j}{\left|\left|\mathbf{X}_j - \mathbf{X}_i\right|\right|} W_{ij} V_j \tag{43}$$

3.2 Stiffness Matrix for Particle Method

We will explore how the stiffness matrix is formed in 2D using the GPS method. We start with the same governing equation and constitutive law,

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} &= \boldsymbol{F} \\ \boldsymbol{\sigma} &= \boldsymbol{D} \nabla \boldsymbol{u} \end{aligned} \tag{44}$$

where $\boldsymbol{D}$ is elastic matrix as following

$$D = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix} \text{for plane stress} \tag{45}$$

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \dfrac{1}{2}\nu \end{bmatrix} \text{for plane strain} \tag{46}$$

The first step to get the stiffness matrix is to describe $\nabla \boldsymbol{u}$ with the gradient operator from the GPS method as described in Eq. (32).

$$\nabla \mathbf{u}_i = \left[\sum_{j \in \Omega_{\mathbf{X}_i}} (\boldsymbol{u}_j - \boldsymbol{u}_i) \otimes \boldsymbol{C}_{ij} W_{ij} V_j \right] \underbrace{\left[\sum_{j \in \Omega_{\mathbf{X}_i}} (\mathbf{X}_j - \mathbf{X}_i) \otimes \boldsymbol{C}_{ij} W_{ij} V_j \right]^{-1}}_{[A]} \tag{47}$$

where $[A]$ is a two by two matrix that is computed first. The summation is applied for every particle within particle i's influence circle. But for simplicity, we will consider there are two particles for the system, particles 1 and 2. The Eq. (47) will be written as following for particle 1,

$$\nabla \mathbf{u}_1 = \begin{bmatrix} (u_2^x - u_1^x) \\ (u_2^y - u_1^y) \end{bmatrix} \begin{bmatrix} (C_{12}^x) & (C_{12}^y) \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} W_{12} V_2 \tag{48}$$

$$= \begin{bmatrix} C_{12}^x (u_2^x - u_1^x) & C_{12}^y (u_2^x - u_1^x) \\ C_{12}^x (u_2^y - u_1^y) & C_{12}^y (u_2^y - u_1^y) \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} W_{12} V_2 \tag{49}$$

Thus,

$$\begin{bmatrix} \dfrac{\partial u_{12}^x}{\partial x} & \dfrac{\partial u_{12}^x}{\partial y} \\ \dfrac{\partial u_{12}^y}{\partial x} & \dfrac{\partial u_{12}^y}{\partial y} \end{bmatrix} = \begin{bmatrix} \left(A_{11} C_{12}^x + A_{21} C_{12}^y\right) (u_2^x - u_1^x) & \left(A_{12} C_{12}^x + A_{22} C_{12}^y\right) (u_2^x - u_1^x) \\ \left(A_{11} C_{12}^x + A_{21} C_{12}^y\right) (u_2^y - u_1^y) & \left(A_{12} C_{12}^x + A_{22} C_{12}^y\right) (u_2^y - u_1^y) \end{bmatrix} W_{12} V_2 \tag{50}$$

Next, we need to convert the 2 by 2 matrix on the left hand side of Eq. (50) to a 4 by 1 vector form.

$$\begin{bmatrix} \dfrac{\partial u^x_{12}}{\partial x} \\ \dfrac{\partial u^x_{12}}{\partial y} \\ \dfrac{\partial u^y_{12}}{\partial x} \\ \dfrac{\partial u^y_{12}}{\partial y} \end{bmatrix} = W_{12} V_2 \begin{bmatrix} \left(A_{11} C^x_{12} + A_{21} C^y_{12}\right) (u^x_2 - u^x_1) \\ \left(A_{12} C^x_{12} + A_{22} C^y_{12}\right) (u^x_2 - u^x_1) \\ \left(A_{11} C^x_{12} + A_{21} C^y_{12}\right) (u^y_2 - u^y_1) \\ \left(A_{12} C^x_{12} + A_{22} C^y_{12}\right) (u^y_2 - u^y_1) \end{bmatrix}$$

$$= W_{12} V_2 \begin{bmatrix} -\left(A_{11} C^x_{12} + A_{21} C^y_{12}\right) & 0 & \left(A_{11} C^x_{12} + A_{21} C^y_{12}\right) & 0 \\ -\left(A_{12} C^x_{12} + A_{22} C^y_{12}\right) & 0 & \left(A_{12} C^x_{12} + A_{22} C^y_{12}\right) & 0 \\ 0 & -\left(A_{11} C^x_{12} + A_{21} C^y_{12}\right) & 0 & \left(A_{11} C^x_{12} + A_{21} C^y_{12}\right) \\ 0 & -\left(A_{12} C^x_{12} + A_{22} C^y_{12}\right) & 0 & \left(A_{12} C^x_{12} + A_{22} C^y_{12}\right) \end{bmatrix}$$

$$\times \begin{bmatrix} u^x_1 \\ u^y_1 \\ u^x_2 \\ u^y_2 \end{bmatrix} \tag{51}$$

We can also get the engineering strain as follows:

$$\begin{bmatrix} \dfrac{\partial u^x_{12}}{\partial x} \\ \dfrac{\partial u^x_{12}}{\partial y} \\ \dfrac{1}{2}\left(\dfrac{\partial u^x_{12}}{\partial y} + \dfrac{\partial u^y_{12}}{\partial x}\right) \end{bmatrix}$$

$$= \underbrace{W_{12} V_2 \begin{bmatrix} -\left(A_{11} C^x_{12} + A_{21} C^y_{12}\right) & 0 & \left(A_{11} C^x_{12} + A_{21} C^y_{12}\right) & 0 \\ 0 & -\left(A_{12} C^x_{12} + A_{22} C^y_{12}\right) & 0 & \left(A_{12} C^x_{12} + A_{22} C^y_{12}\right) \\ \dfrac{-\left(A_{12} C^x_{12} + A_{22} C^y_{12}\right)}{2} & \dfrac{-\left(A_{11} C^x_{12} + A_{21} C^y_{12}\right)}{2} & \dfrac{\left(A_{12} C^x_{12} + A_{22} C^y_{12}\right)}{2} & \dfrac{-\left(A_{11} C^x_{12} + A_{12} C^y_{12}\right)}{2} \end{bmatrix}}_{[\nabla_1]\text{:gradient operator for particle 1}} \underbrace{\begin{bmatrix} u^x_1 \\ u^y_1 \\ u^x_2 \\ u^y_2 \end{bmatrix}}_{[\boldsymbol{u}]} \tag{52}$$

Eq. (52) can be simply written as

$$[\epsilon_1] = [\nabla_1][\boldsymbol{u}] \tag{53}$$

Stress σ can be computed by multiplying elastic matrix, $[D]$, from Eqs. (45) and (46).

$$[\sigma_1] = [D][\nabla_1][\boldsymbol{u}] \tag{54}$$

It is worth noting that $[D][\nabla][u]$ includes all the components in the particle i's influence domain.

$$[\sigma] = [D][\nabla][\boldsymbol{u}] = \begin{bmatrix} [D][\nabla_1] \\ [D][\nabla_2] \\ \vdots \\ [D][\nabla_n] \end{bmatrix} [\boldsymbol{u}] \tag{55}$$

where $[\nabla_1]$ and $[\nabla_2]$ represent the gradient operator obtained from the first and the second particle as the i th particle, respectively. This suggests that, in general, the gradient operator for every particle in the system needs to be computed before applying divergence operator.

Next, we will show how to construct the divergence operator into a matrix form similar to how the gradient operator matrix was formed as shown above.

$$\nabla \cdot \sigma_1 = \begin{bmatrix} C^x_{12} \\ C^y_{12} \end{bmatrix} \cdot \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \sigma^2_{11} - \sigma^1_{11} & \sigma^2_{12} - \sigma^1_{12} \\ \sigma^2_{21} - \sigma^1_{21} & \sigma^2_{22} - \sigma^1_{22} \end{bmatrix} W_{12} V_2 \tag{56}$$

$$= \begin{bmatrix} A_{11} C^x_{12} + A_{21} C^y_{12} & A_{12} C^x_{12} + A_{22} C^y_{12} \end{bmatrix} \begin{bmatrix} \sigma^2_{11} - \sigma^1_{11} & \sigma^2_{12} - \sigma^1_{12} \\ \sigma^2_{21} - \sigma^1_{21} & \sigma^2_{22} - \sigma^1_{22} \end{bmatrix} W_{12} V_2 \tag{57}$$

$$= \begin{bmatrix} B_1(\sigma^2_{11} - \sigma^1_{11}) & B_2(\sigma^2_{12} - \sigma^1_{12}) \\ B_1(\sigma^2_{21} - \sigma^1_{21}) & B_2(\sigma^2_{22} - \sigma^1_{22}) \end{bmatrix} \tag{58}$$

where $B_1 = (A_{11} C^x_{12} + A_{21} C^y_{12})$ and $B_2 = (A_{12} C^x_{12} + A_{22} C^y_{12})$ for simplicity. Here we further isolate σ together.

$$\begin{bmatrix} \nabla \cdot \sigma^x_1 \\ \nabla \cdot \sigma^y_1 \end{bmatrix} = W_{12} V_2 \begin{bmatrix} -B_1 & -B_2 & 0 & 0 & B_1 & B_2 & 0 & 0 \\ 0 & 0 & -B_1 & -B_2 & 0 & 0 & B_1 & B_2 \end{bmatrix} \begin{bmatrix} \sigma^1_{11} \\ \sigma^1_{12} \\ \sigma^1_{21} \\ \sigma^1_{22} \\ \sigma^2_{11} \\ \sigma^2_{12} \\ \sigma^2_{21} \\ \sigma^2_{22} \end{bmatrix} \tag{59}$$

or for the engineering stress,

$$\begin{bmatrix} \nabla \cdot \sigma^x_1 \\ \nabla \cdot \sigma^y_1 \end{bmatrix} = \underbrace{W_{12} V_2 \begin{bmatrix} -B_1 & 0 & -B_2 & B_1 & 0 & B_2 \\ 0 & -B_2 & -B_1 & 0 & B_2 & B_1 \end{bmatrix}}_{[\nabla\cdot]\text{:divergence operator for particle 1}} \begin{bmatrix} \sigma^1_{xx} \\ \sigma^1_{yy} \\ \tau^1_{xy} \\ \sigma^2_{xx} \\ \sigma^2_{yy} \\ \tau^2_{xy} \end{bmatrix} \tag{60}$$

Thus, we can get the stiffness matrix for the particle 1, by combining Eqs. (55) and (60),

$$[\nabla\cdot]\,[D][\nabla][\boldsymbol{u}] = \boldsymbol{F}$$

$$\underbrace{W_{12}V_2\begin{bmatrix} -B_1 & 0 & -B_2 & B_1 & 0 & B_2 \\ 0 & -B_2 & -B_1 & 0 & B_2 & B_1 \end{bmatrix}\begin{bmatrix} [D]\,[\nabla_1] \\ [D]\,[\nabla_2] \end{bmatrix}}_{[K_1]}[[\boldsymbol{u}]] = \begin{bmatrix} F_1^x \\ F_1^y \end{bmatrix} \tag{61}$$

This process is repeated for each particle in the system. After generating $[K_i]$ matrices, they are assembled together to form the system stiffness matrix $[K]$.

4 Equations of Motion for a System of Multiple Subdomains

Differential Algebraic Equations (DAE) are differential equations with algebraic constraints. Consider the equation of motion of a body that is subject to constraints $g(\mathbf{u}) = 0$. The action integral is given as

$$\mathscr{S}(\mathbf{u}, \lambda^*) = \int_{t_0}^{t_f} \mathscr{L}\partial t + \lambda^{*T} g(\mathbf{u}) \tag{62}$$

and we can take the variation of the action integral,

$$\delta\mathscr{S}\,(\mathbf{u}, \lambda^*) = 0 \rightarrow \frac{\partial}{\partial t}\left(\frac{\partial\mathscr{L}\left(\mathbf{u}, \dot{\mathbf{u}}\right)}{\partial\dot{\mathbf{u}}}\right) - \frac{\partial\mathscr{L}\left(\mathbf{u}, \dot{\mathbf{u}}\right)}{\partial\mathbf{u}} + G^T\,(\mathbf{u})\,\lambda^* = 0 \tag{63}$$

$$g(\mathbf{u}) = 0 \tag{64}$$

where $G(\mathbf{u}) = \partial g(\mathbf{u})/\partial\mathbf{u}$. For the standard DAE equations of second order transient systems, the above equation can be written as

$$\mathbf{Q}^{iner}\left(\mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}}\right) + \mathbf{Q}^{int}\,(\mathbf{u}) = \mathbf{Q}^{appl}\left(\mathbf{u}, \dot{\mathbf{u}}, t\right) + \mathbf{G}^T\,(\mathbf{u})\,\lambda^* \tag{65}$$

$$g(\mathbf{u}) = 0 \tag{66}$$

where, $\mathbf{Q}^{iner}$, $\mathbf{Q}^{int}$ and $\mathbf{Q}^{appl}$ are the inertial, internal and applied forces, respectively. The Lagrange multiplier variable, $\lambda^*(t)$, is introduced in order to accomplish the constraint equation.

Now, consider a domain of interest Ω that is divided into n_{dom} number of subdomains, Ω_i (for $i = 1, 2, \ldots, n_{\text{dom}}$), with each subdomain joined by a non-overlapping interface in between. An example with three subdomains is shown in Fig. 2. In the case of linear semidiscrete second order transient systems, the equation of motion for subdomain Ω_i is, (for $i = 1, 2, \ldots, n_{\text{dom}}$),

$$\mathbf{M}_i\ddot{\mathbf{u}}_i = \mathbf{Q}_i^{\text{int}}\,(\mathbf{u}_i) + \mathbf{Q}_i^{\text{appl}}\left(\mathbf{u}_i, \dot{\mathbf{u}}_i, t\right) + \mathbf{G}_i^T\,\lambda^*\,(t) \tag{67a}$$

$$\mathbf{0} = \sum_{i=1}^{n_{\text{dom}}} \mathbf{G}_i\dot{\mathbf{u}}_i \tag{67b}$$

where $\mathbf{u}_i(t)$, $\dot{\mathbf{u}}_i\,(t) := d\mathbf{u}_i/dt$, and $\ddot{\mathbf{u}}_i\,(t) := d^2\mathbf{u}_i/dt^2$ are the displacement, velocity, and acceleration vectors of the i^{th}-subdomain, respectively; $\mathbf{M}_i$ is the mass matrix of the i^{th}-subdomain; and $\mathbf{Q}_i^{\text{int}}(\mathbf{u}_i)$ and $\mathbf{Q}_i^{\text{appl}}\left(\mathbf{u}_i, \dot{\mathbf{u}}_i, t\right)$ are the internal and applied force of the i^{th}-subdomain, respectively.

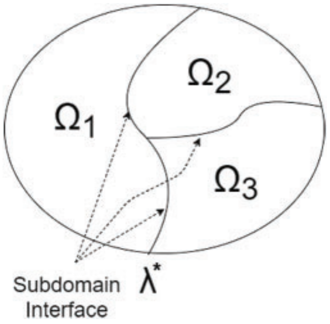


Figure 2: A domain Ω divided into three subdomains with non-overlapping interface

The matrix $\mathbf{G}_i$ is a boolean matrix with entries either -1, 0, or $+1$, that is defined by the subdomain interface conditions. The nodes or particles on one side of the interface will have -1 while the corresponding nodes or particles on the other side of the interface will have value of $+1$ and all other nodes/particles will have 0. Eq. (67b) indicates that the velocity continuity along the interface is enforced.

4.1 DAE Framework with GS4 Family of Algorithms for a System with Multiple Subdomains and Multiple Time Step Sizes

Let Δt and Δt_i be the system time step size and the subdomain time step size for Ω_i, respectively, and define the ratio as $\tau_i = \Delta t / \Delta t_i \geq 1$ for $i = 1, 2, \ldots, n_{\text{dom}}$. The system and subdomain time step sizes are defined as

$$\Delta t = t_{n+1} - t_n \tag{68a}$$

$$\Delta t_i = t_{n+\frac{m+1}{\tau_i}} - t_{n+\frac{m}{\tau_i}} \tag{68b}$$

where n denotes the system time step, and m represents subdomain time step. The visual representation of time subcylcing is shown in Fig. 3.

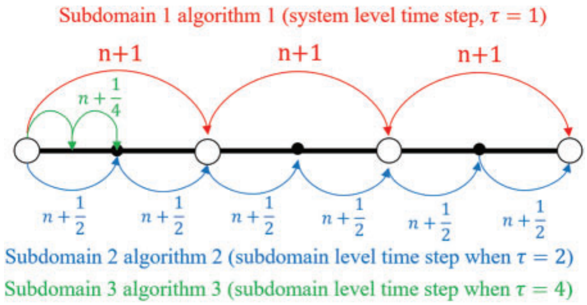

Subdomain 1 algorithm 1 (system level time step, $\tau = 1$)
n+1
n+1
n+1
$n+\frac{1}{4}$
$n+\frac{1}{2}$
$n+\frac{1}{2}$
$n+\frac{1}{2}$
$n+\frac{1}{2}$
$n+\frac{1}{2}$
$n+\frac{1}{2}$
Subdomain 2 algorithm 2 (subdomain level time step when $\tau = 2$)
Subdomain 3 algorithm 3 (subdomain level time step when $\tau = 4$)

Figure 3: Time step subcycling for different subdomains

Now applying the GS4 family of algorithms, the equation of motion of each subdomain can be written as following:

$$\mathbf{M}_i \widetilde{\mathbf{a}}_i^{(n,m)} = \mathbf{Q}_i^{\text{int}} \left(\widetilde{\mathbf{u}}_i^{(n,m)} \right) + \mathbf{Q}_i^{\text{appl}} \left(\widetilde{\mathbf{u}}_i^{(n,m)}, \widetilde{\mathbf{v}}_i^{(n,m)}, t_{n+\frac{m+W_1}{\tau_i}} \right) + \mathbf{G}_i^T \widetilde{\lambda}^{*(n,m)} \tag{69a}$$

$$\mathbf{0} = \sum_{i=1}^{n_{\text{dom}}} \mathbf{G}_i \mathbf{v}_i^{n+1} \tag{69b}$$

where

$$t_{n+\frac{m+W_1}{\tau_i}} := t_n + \left(\frac{m+W_1}{\tau_i}\right)\Delta t = t_0 + n\Delta t + (m+W_1)\,\Delta t_i \tag{69c}$$

$$\widetilde{\lambda}^{*(n,m)} := \lambda^{*n} + \left(\frac{m+W_1}{\tau_i}\right)\left(\lambda^{*n+1} - \lambda^{*n}\right) \tag{69d}$$

$$\widetilde{\mathbf{a}}_i^{(n,m)} := \mathbf{a}_i^{n+\frac{m}{\tau_i}} + W_1\Lambda_6\Delta\mathbf{a}_i^{(n,m)} \tag{69e}$$

$$\widetilde{\mathbf{v}}_i^{(n,m)} := \mathbf{v}_i^{n+\frac{m}{\tau_i}} + \Delta t W_1\Lambda_4\mathbf{a}_i^{n+\frac{m}{\tau_i}} + \Delta t W_2\Lambda_5\Delta\mathbf{a}_i^{(n,m)} \tag{69f}$$

$$\widetilde{\mathbf{u}}_i^{(n,m)} := \mathbf{u}_i^{n+\frac{m}{\tau_i}} + \Delta t W_1\Lambda_1\mathbf{v}_i^{n+\frac{m}{\tau_i}} + \Delta t^2 W_2\Lambda_2\mathbf{a}_i^{n+\frac{m}{\tau_i}} + \Delta t^2 W_3\Lambda_3\Delta\mathbf{a}_i^{(n,m)} \tag{69g}$$

It is worth noting that the Lagrange multipliers are approximated in the system time level rather than in the subdomain time level. This factor suggests that the solution must be in the system time level even though the displacement, velocity, and acceleration are approximated in the subdomain time step.

The updates of each of the unknown variables with the GS4 framework of algorithms are

$$\mathbf{u}_i^{n+\frac{m+1}{\tau_i}} = \mathbf{u}_i^{n+\frac{m}{\tau_i}} + \Delta t\lambda_1\mathbf{v}_i^{n+\frac{m}{\tau_i}} + \Delta t^2\lambda_2\mathbf{a}_i^{n+\frac{m}{\tau_i}} + \Delta t^2\lambda_3\Delta\mathbf{a}_i^{(n,m)} \tag{70a}$$

$$\mathbf{v}_i^{n+\frac{m+1}{\tau_i}} = \mathbf{v}_i^{n+\frac{m}{\tau_i}} + \Delta t\lambda_4\mathbf{a}_i^{n+\frac{m}{\tau_i}} + \Delta t\lambda_5\Delta\mathbf{a}_i^{(n,m)} \tag{70b}$$

$$\mathbf{a}_i^{n+\frac{m+1}{\tau_i}} = \mathbf{a}_i^{n+\frac{m}{\tau_i}} + \Delta\mathbf{a}_i^{(n,m)} \tag{70c}$$

$W\Lambda$ and λ are the GS4 algorithmic parameters and can be found in [37]. The advantage of the GS4 algorithm is that it allows one to easily switch between various time integration algorithms as well as explicit and implicit families based on the GS4 algorithmic parameters. A single analysis code readily enables current and future researchers and analysts to investigate a multitude of algorithms and combinations in the academic and commercial software world; thereby making it a much desired computational simulation toolkit.

In linear dynamical systems, Eq. (67) may be written as

$$\mathbf{M}_i\widetilde{\mathbf{a}}_i + \boldsymbol{C}_i\widetilde{\mathbf{v}}_i + \mathbf{K}_i\widetilde{\mathbf{u}}_i = \mathbf{Q}_i^{\text{appl}}(t) + \mathbf{G}_i^T\,\widetilde{\lambda}^*(t) \tag{71a}$$

$$\mathbf{0} = \sum_{i=1}^{n_{\text{dom}}}\mathbf{G}_i\dot{\mathbf{u}}_i^{n+1} \tag{71b}$$

where $\boldsymbol{C}_i$ and $\mathbf{K}_i$ are the damping and stiffness matrices for Ω_i, respectively; and $\mathbf{Q}_i^{\text{appl}}(t)$ is the external force vector for Ω_i. Since the stiffness matrix is different for each subdomain, various methods can be coupled together with each subdomain having the stiffness matrix derived from the corresponding methods. For example, subdomain 1 and 2 in Fig. 2 can use the stiffness matrix derived from the FEM while subdomain 3 adopts the stiffness matrix formulated using the GPS method as outlined in the previous section.

A set of all kinematic unknowns for Ω_i in its subdomain time step is defined as

$$\mathbf{X}_i^{n+\frac{m}{\tau_i}} := \begin{bmatrix} \mathbf{a}_i^{n+\frac{m}{\tau_i}} \\ \mathbf{v}_i^{n+\frac{m}{\tau_i}} \\ \mathbf{u}_i^{n+\frac{m}{\tau_i}} \end{bmatrix} \tag{72}$$

Eq. (71a) can then be discretized and written as follows:

$$\mathbf{L}_i\mathbf{X}_i^{n+\frac{m+1}{\tau_i}} - \left(\frac{m+W_1}{\tau_i}\right)\left(\hat{\mathbf{G}}_i^{(1)}\right)^T\left(\lambda^{*n+1}\right) = \mathbf{Q}_i^{n+\frac{m+W_1}{\tau_i}} + \left(1-\frac{m+W_1}{\tau_i}\right)\left(\hat{\mathbf{G}}_i^{(1)}\right)^T\lambda^{*n} + \boldsymbol{R}_i\mathbf{X}_i^{n+\frac{m}{\tau_i}} \tag{73}$$

where

$$\mathbf{L}_i := \begin{bmatrix} \mathbf{L}_i^* & \mathbf{O} & \mathbf{O} \\ -\Delta t_i\lambda_5\mathbf{I} & \mathbf{I} & \mathbf{O} \\ -\Delta t_i^2\lambda_3\mathbf{I} & \mathbf{O} & \mathbf{I} \end{bmatrix} \text{ with } \mathbf{L}_i^* := W_1\Lambda_6\mathbf{M}_i + \Delta t_i W_2\Lambda_5\boldsymbol{C}_i + \Delta t_i^2 W_3\Lambda_3\mathbf{K}_i \tag{74}$$

$$\boldsymbol{R}_i := \begin{bmatrix} \boldsymbol{R}_i^* & -(1+\Delta t_i W_1\Lambda_1)\mathbf{I} & -\mathbf{I} \\ \Delta t_i(\lambda 4-\lambda 5)\mathbf{I} & \mathbf{I} & \mathbf{O} \\ \Delta t_i^2(\lambda 2-\lambda 3)\mathbf{I} & \Delta t_i\lambda_1\mathbf{I} & \mathbf{I} \end{bmatrix} \tag{75a}$$

with

$$\boldsymbol{R}_i^* := (W_1\Lambda_6-1)\mathbf{M}_i + \Delta t_i(W_2\Lambda_5 - W_1\Lambda_4)\boldsymbol{C}_i + \Delta t_i^2(W_3\Lambda_3 - W_2\Lambda_2)\mathbf{K}_i \tag{75b}$$

$$\mathbf{Q}_i^{n+\frac{m+W_1}{\tau_i}} := \begin{bmatrix} \mathbf{Q}_i\left(t_{n+\frac{m+W_1}{\tau_i}}\right) \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{76}$$

and

$$\hat{\mathbf{G}}_i^{(1)} := \left[\mathbf{G}_i \quad \mathbf{O}_{n_c\times n_i} \quad \mathbf{O}_{n_c\times n_i}\right] \tag{77}$$

where $\mathbf{O}$ represent zero-matrix, n_c and n_i refer to the number of constraint degrees of freedom, and total number of degrees of freedom in one domain, respectively. Now, we define a set of all kinematic unknowns, i.e., the acceleration, velocity, and displacement vectors, for Ω_i over its subdomain time step and a set of all kinematic unknowns of all subdomains in the system over a system time step as

$$\mathbf{X}_i^{(n+1)} := \begin{bmatrix} \mathbf{X}_i^{n+\frac{1}{\tau_i}} \\ \mathbf{X}_i^{n+\frac{2}{\tau_i}} \\ \vdots \\ \mathbf{X}_i^{n+\frac{\tau_i-1}{\tau_i}} \\ \mathbf{X}_i^{n+1} \end{bmatrix} \text{ and } \mathbb{X}^{(n+1)} := \begin{bmatrix} \mathbf{X}_1^{(n+1)} \\ \mathbf{X}_2^{(n+1)} \\ \vdots \\ \mathbf{X}_{n_{\text{dom}}}^{(n+1)} \end{bmatrix} \tag{78}$$

The constraint equation at the velocity level, i.e., Eq. (71b), can be written in the form

$$\mathbf{0} = \sum_{i=1}^{n_{\text{dom}}} \mathbb{G}_i\mathbb{X}_i^{(n+1)} = \mathbb{G}\mathbb{X}^{(n+1)} \tag{79}$$

where

$$\mathbb{G} := \begin{bmatrix} \mathbb{G}_1 & \mathbb{G}_2 & \cdots & \mathbb{G}_{n_{\text{dom}}} \end{bmatrix} \tag{80}$$

and

$$\mathbb{G}_i := \begin{bmatrix} -\frac{1}{\tau_i}\mathbf{O}_{n_c \times 3n_i} & -\frac{2}{\tau_i}\mathbf{O}_{n_c \times 3n_i} & \cdots & -\frac{\tau_i - 1}{\tau_i}\mathbf{O}_{n_c \times 3n_i} & \hat{\mathbf{G}}_i \end{bmatrix} \tag{81}$$

with

$$\hat{\mathbf{G}}_i := \begin{bmatrix} \mathbf{O}_{n_c \times n_i} & \mathbf{G}_i & \mathbf{O}_{n_c \times n_i} \end{bmatrix} \tag{82}$$

Eq. (82) shows the velocity constraint. The size of $\mathbb{G}_i$ is n_c by $3n_i\tau_i$ since the system solves for the displacement, velocity and acceleration at every subcycling step simultaneously. However, Eq. (81) suggests that the constraint is enforced only at the n+1 time step.

The equation of motion for the entire system for a system time step level can be represented in a simple matrix form as following:

$$\begin{bmatrix} \mathbb{A} & \mathbb{B} \\ \mathbb{G} & \mathbb{O} \end{bmatrix} \begin{bmatrix} \mathbb{X}^{(n+1)} \\ \lambda^{*n+1} \end{bmatrix} = \begin{bmatrix} \mathbb{F}^{(n)} \\ \mathbf{0} \end{bmatrix} \tag{83}$$

where

$$\mathbb{A} := \operatorname{diag}\left[\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{n_{\text{dom}}}\right] \tag{84}$$

in which

$$\mathbf{A}_i := \begin{bmatrix} \mathbf{L}_i & & & & \\ -\boldsymbol{R}_i & \mathbf{L}_i & & & \\ & -\boldsymbol{R}_i & \mathbf{L}_i & & \\ & & \ddots & \ddots & \\ & & & -\boldsymbol{R}_i & \mathbf{L}_i \end{bmatrix} \tag{85}$$

$$\mathbb{B} := \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_{n_{\text{dom}}} \end{bmatrix} \tag{86}$$

in which

$$\mathbf{B}_i := \begin{bmatrix} -\frac{1}{\tau_i}\hat{\mathbf{G}}_i^{(1)} & -\frac{2}{\tau_i}\hat{\mathbf{G}}_i^{(1)} & \cdots & -\frac{\tau_i - 1}{\tau_i}\hat{\mathbf{G}}_i^{(1)} & -\hat{\mathbf{G}}_i^{(1)} \end{bmatrix}^T \tag{87}$$

and

$$\mathbb{F}^{(n)} := \begin{bmatrix} \mathbf{F}_1^{(n)} \\ \mathbf{F}_2^{(n)} \\ \vdots \\ \mathbf{F}_{n_{\text{dom}}}^{(n)} \end{bmatrix} \tag{88}$$

in which

$$\mathbf{F}_i^{(n)} := \begin{bmatrix} \mathbf{Q}_i^{n+\frac{1+W_1}{\tau_i}} + \left(1 - \hat{\mathbf{G}}_i^{(1)}\right)^T \lambda^{*n} + \boldsymbol{R}_i \mathbf{X}_i^n \\ \mathbf{Q}_i^{n+\frac{2+W_1}{\tau_i}} + \left(1 - \hat{\mathbf{G}}_i^{(1)}\right)^T \lambda^{*n} \\ \vdots \\ \mathbf{Q}_i^{n+\frac{\tau_i+W_1}{\tau_i}} + \left(1 - \hat{\mathbf{G}}_i^{(1)}\right)^T \lambda^{*n} \end{bmatrix} \tag{89}$$

4.2 Computational Procedure

The computation flowchart for the GS4-II DAE framework for the subdomain method employing different spatial methods, algorithms, and time steps is shown below in Fig. 4 which here which provides a wide variety of algorithms and choices as options to the analyst.

```
for $dom := 1$ to $N_{dom}$ do
    Calculate $\mathbf{M}, \mathbf{C}, \mathbf{K}$ using corresponding method for the domain
    Calculate $\mathbf{L}_i$, and $\mathbf{R}_i$ using Eqs. (74) and (75a)
    Calculate $\hat{\mathrm{G}}_{\mathrm{i}}$ using Eq. (82)
     for $\tau := 1$ to n do
        Assemble $\mathbf{A}_i$ using Eq. (85)
        Assemble $\mathbf{B}_i$ using Eq. (87)
     end for
end for
Assemble $\mathbb{A}$ using Eq. (84)
Assemble $\mathbb{B}$ using Eq. (86)
Transpose $\mathbb{B}$ to obtain $\mathbb{G}$
for ($t := 1$ to Timesteps) do
    for $dom := 1$ to $N_{dom}$ do
        Calculate $\mathbf{F}_i$ using Eq. (89)
    end for
    Assemble $\mathbb{F}$ using Eq. (88)
    Assemble $\hat{}\mathbf{R}_{glob}$ and $\hat{\mathbf{J}}_{glob}$ from Eq. (83)
    Solve ($\hat{\mathbf{J}}_{glob}\mathbf{X}_{n+1} = \hat{}\mathbf{R}_{glob}$)
end for
```

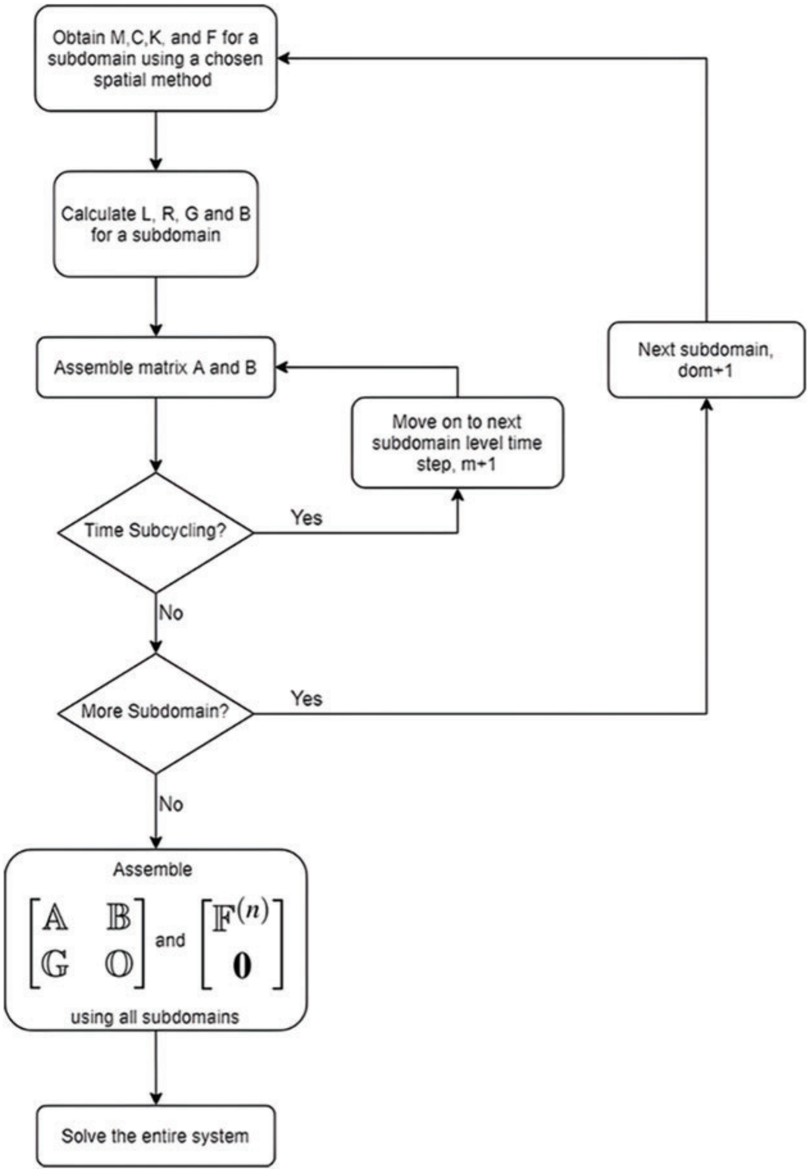

Obtain M,C,K, and F for a subdomain using a chosen spatial method
Calculate L, R, G and B for a subdomain
Assemble matrix A and B
Move on to next subdomain level time step, m+1
Next subdomain, dom+1
Time Subcycling?
Yes
No
More Subdomain?
Yes
No
Assemble
$\begin{bmatrix} \mathbb{A} & \mathbb{B} \\ \mathbb{G} & \mathbb{O} \end{bmatrix}$ and $\begin{bmatrix} \mathbb{F}^{(n)} \\ \mathbf{0} \end{bmatrix}$
using all subdomains
Solve the entire system

Figure 4: Flowchart for constructing DAE second order system

4.3 Subdomain Coupling Condition and Time Level Consistency for Second Order Systems

The second order time accuracy of all the primary variables and the Lagrange multipliers can be achieved for interfacing different subdomains (this is in general not plausible with traditional methods) if W_1, a parameter in GS4, is the same for all subdomains. In order to guarantee the second order time accuracy in the acceleration, the algorithmic acceleration $\mathbf{a}_i^{n+\frac{m}{\tau_i}}$ must be an approximation at time level t_n:

$$\mathbf{a}_i^{n+\frac{m}{\tau_i}} \approx \ddot{\mathbf{u}}\left(t_n + (m - \phi_i)\,\Delta t_i\right) \tag{90}$$

where $\phi_i := W_1\Lambda_6 - W_1$ in which W_1 and $W_1\Lambda_6$ are the values used for subdomain Ω_i.

The GS4-II algorithms are written in the form of U0(ρ_{min}, ρ_{max}, ρ_s) for the U0 family and V0 (ρ_{min}, ρ_{max}, ρ_s) for the V0 family. The relation of the variables follows $0 \le \rho_s \le \rho_{min} \le \rho_{max} \le 1$. And depending on the value given to each variable, it recovers various existing time schemes including a wide variety of new and optimal methods and also provides new avenues to integrate multiple methods in a single analysis; for example, U0(1, 1, 0) recovers Newmark time scheme, and U0(0, 0, 0) is equal to the WBZ scheme. The V0 family represents an altogether new family of algorithms and is not well known to others and is relatively new and not well understood by the traditional research community at large; mathematically, the V0 family of algorithms are the mirror counterparts of the U0 family but with different overshoot behavior unlike several of the traditional methods. More recent efforts show the advances in the GS4 V0∗ family of algorithms that enable consistent second order time accurate features with and without physical damping [38]. Certain algorithms within the U0 family are also new with certain exceptions which are existing methods. The GS4 encompasses the entire family of LMS methods that are second order time accurate. Only three parameters are needed to identify a particular algorithm; and no two algorithms can have the same DNA (Discrete Numerically Assigned markers).

With the W_1 parameter condition, the GS4-II family of time integration scheme combinations guarantee second order time accuracy of all variables and the Lagrange multipliers; this is unlike existing technology which does not permit such implicit interfacing of arbitrary methods whilst consistently preserving second order accuracy. Some of the possible GS4-II time scheme combinations are listed in Table 1. For example, one cannot simply or haphazardly couple the Midpoint rule with the Newmark method; or couple the Newmark, HHT, WBZ methods, etc., arbitrarily in different subdomains in a single analysis without loss and reduced accuracy in the convergence rate of one or more variables.

Table 1: Coupling different time scheme algorithms

Case	Subdomain 1	Subdomain 2
1	U0(1, 1, 0) (Newmark)	U0(0, 0, 0) (WBZ)
2	U0(1, 1, 0) (Newmark)	U0(0, 1, 0) (U0V0 optimal)
3	U0(0.5, 1, 0.5) (U0V0 optimal)	U0(0.5, 0.5, 0.5) (Three parameter optimal)
4	V0(1, 1, 0) (MPR-MPA)	V0(1, 1, 1) (MPR-EPA)
5	V0(0.7, 0.7, 0.7)	V0(0.7, 0.7, 0.21)

For the U0 family of GS4-II algorithms, the W_1 value is determined by ρ_s parameter; whereas, for the V0 family, ρ_{min} and ρ_{max} parameters control the W_1 value as shown below. It is also possible to couple U0 and V0 family if the W_1 condition is satisfied. For example, U0(1, 1, 0) scheme and V0(1/3, 1/3, 1/3) scheme both produce W_1 value of 1; thus, they can be coupled together. These criteria must be followed to preserve second-order time accuracy in all the variables and the Lagrange multipliers.

For GS4-II U0 Family: $$W_1 = \frac{1}{1+\rho_s} \tag{91}$$

For GS4-II V0 and V0* Family: $$W_1 = \frac{3+\rho_{min}+\rho_{max}-\rho_{min}\rho_{max}}{2(1+\rho_{min})(1+\rho_{max})} \tag{92}$$

5 Numerical Examples

5.1 2D Wave Propagation Problem

Fig. 5 shows the 2D geometry of a wave propagation problem. It is a rectangular bar of 10 m by 1 m with 1 m thickness, which is fixed on the left side wall as boundary condition. The semidiscrete equation of motion for the system is given as

$$\mathbf{M}_i\ddot{\mathbf{u}}_i + \mathbf{K}_i\mathbf{u}_i = \mathbf{G}_i^T\lambda_i^* \tag{93}$$

$$\sum_{i=1}^{i=N_{dom}} \mathbf{G}_i\mathbf{v}_i = 0 \tag{94}$$

$$\mathbf{u}(0, y, t) = 0 \tag{95}$$

$$\frac{\partial \mathbf{u}}{\partial x}(L, y, t) = p \tag{96}$$

$$\mathbf{u}(x, y, 0) = 0 \tag{97}$$

$$\dot{\mathbf{u}}\,(x, y, 0) = 0 \tag{98}$$

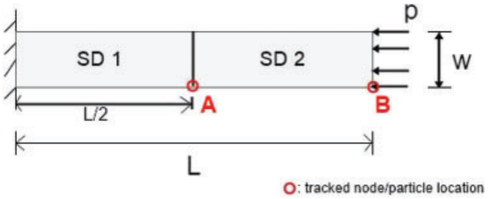


Figure 5: Diagram of structure for wave propagation problem

The bar is compressed from the right side with p, the distributed load, of 10,000 N/m. This material has 72 GPa for Young's modulus, 0.3 for Poisson's ratio, and 2712 kg/m^3 for density. The bar is divided into two subdomains, and each subdomain is implemented with 17 nodes/particles in length, and 5 nodes/particles in height making it total of 170 nodes/particles with 5 nodes/particles overlap on the interface. Figs. 6 and 7 show the FEM mesh and particle distribution for the two subdomains. The system Δt for time integration is set to be 0.8 μs. The time history of the x-direction displacement, velocity and acceleration of a node/particle at the interface on the subdomain 1 (SD1) side (point A in Fig. 5) and the free end on the subdomain 2 (SD2) side (point B in Fig. 5) are evaluated. We compare the result in four sets: spatial method variance where we combine the different spatial methods; explicit and implicit variance where we pair explcit and implicit version of the same time scheme; time shceme variance where the different combination of the GS4-2 time schemes are tested; and time subcycling variance.

Fig. 8 compares the three different cases of coupling different methods: FEM+FEM, GPS+GPS, and FEM+GPS; and shows the time history of the displacement, velocity, and acceleration at A in subdomain 1 and at point B in subdomain 2. It is clear that coupling the same methods or coupling different methods, be it weak form or strong form methods together, produce a comparable result with the result from commercial software such as ABAQUS.

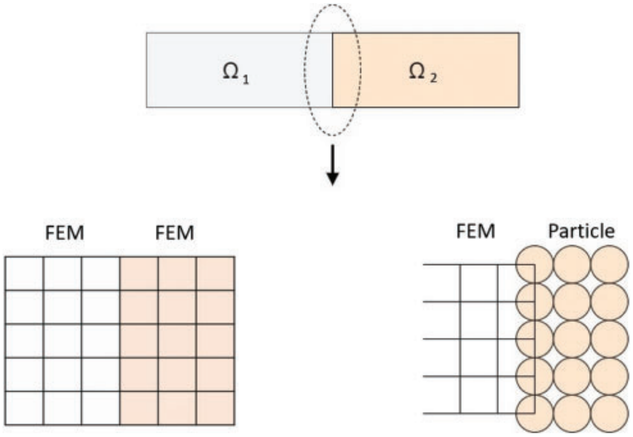


Figure 6: FEM+FEM and FEM+GPS interface

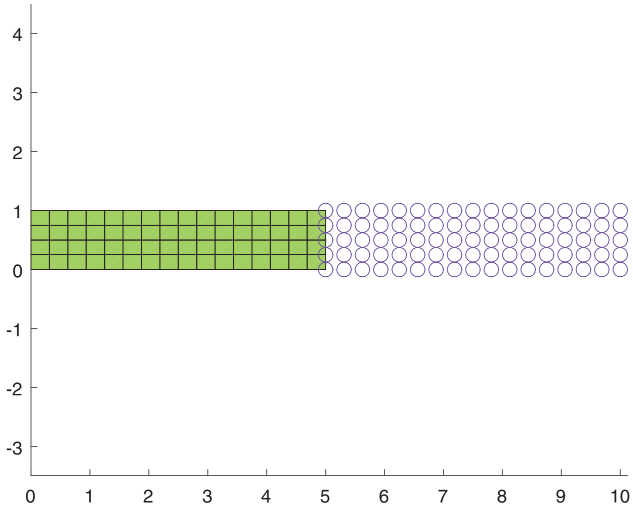
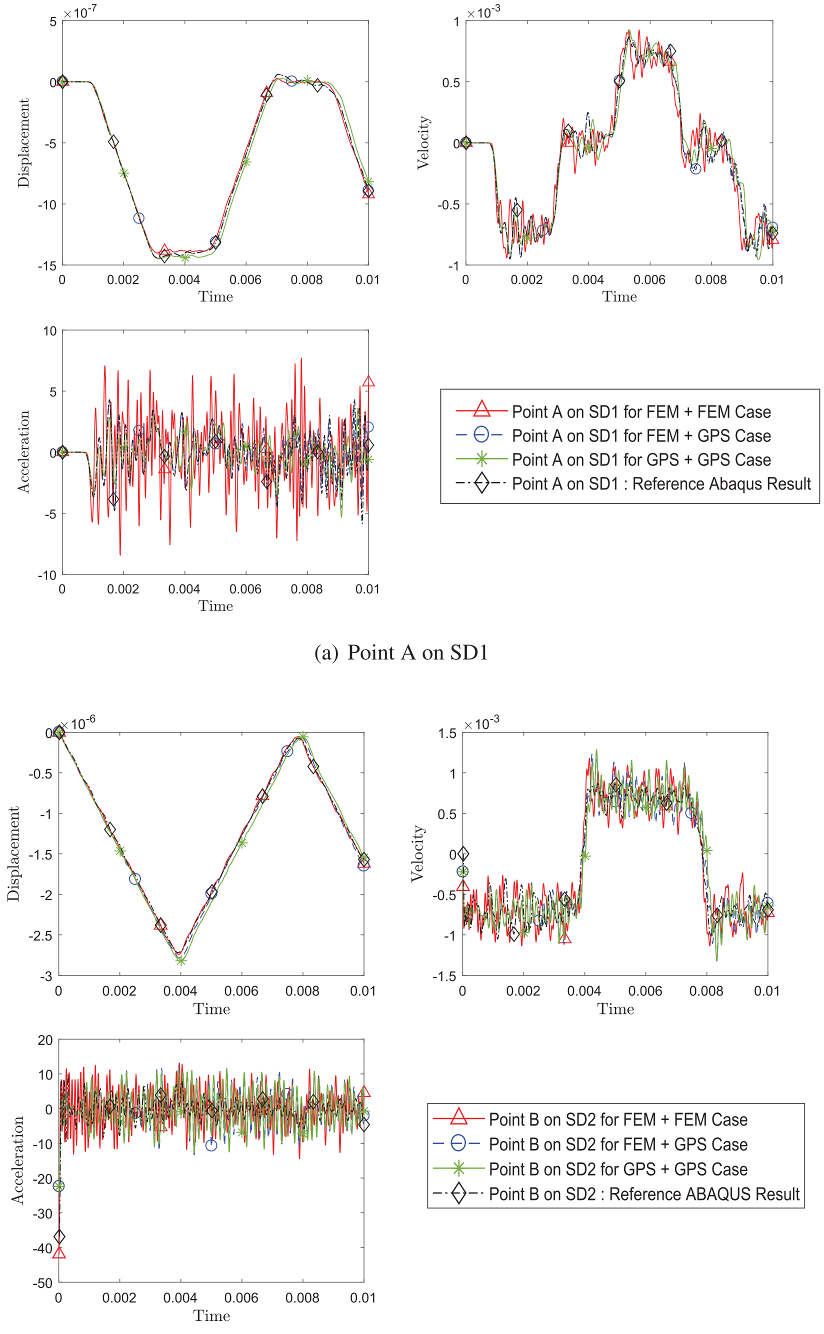


Figure 7: FEM+GPS two domain mesh of the structure

Fig. 9 shows the result for coupling explicit version and implicit version of the schemes. The evaluated coupling cases are explicit and explicit case, explicit and implicit case, and implicit and explicit case where the first condition refers to the time scheme algorithm for subdomain 1 and the latter for subdomain 2. This figure not only suggests that explicit and implicit time algorithms can be coupled together, it also shows that coupling is not limited to just Newmark family of algorithms as this example couples U0(1, 1, 0) (equivalent to Newmark, central difference for explicit, and average acceleration for implicit) and U0(0, 0, 0) (equivalent to WBZ).

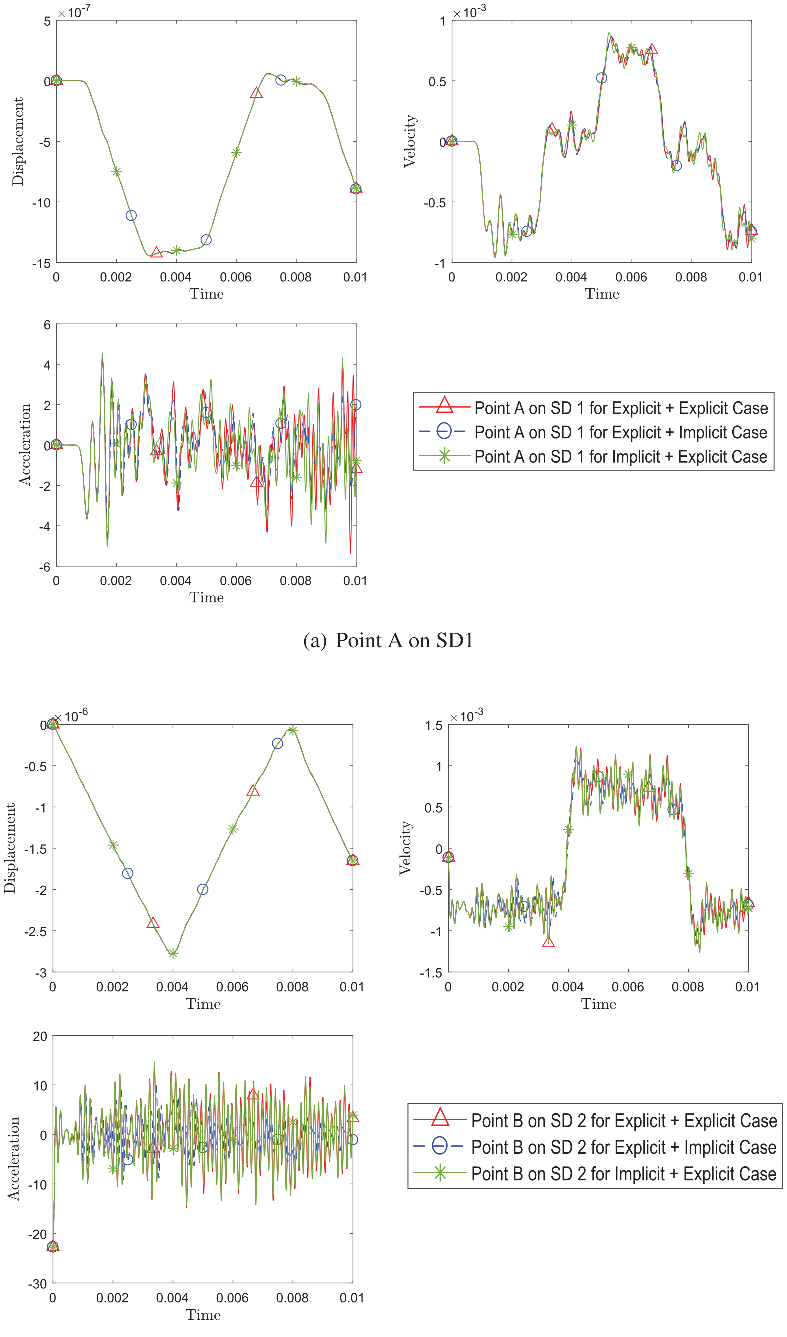
To further present evidence of coupling various time scheme algorithms, five different cases listed in Table 1, are evaluated and illustrated in Fig. 10. The time scheme algorithms are selected such that a combination has identical W_1 to ensure the second order accuracy in time. It is evident that all the results align closely.



(a) Point A on SD1

(b) Point B on SD2

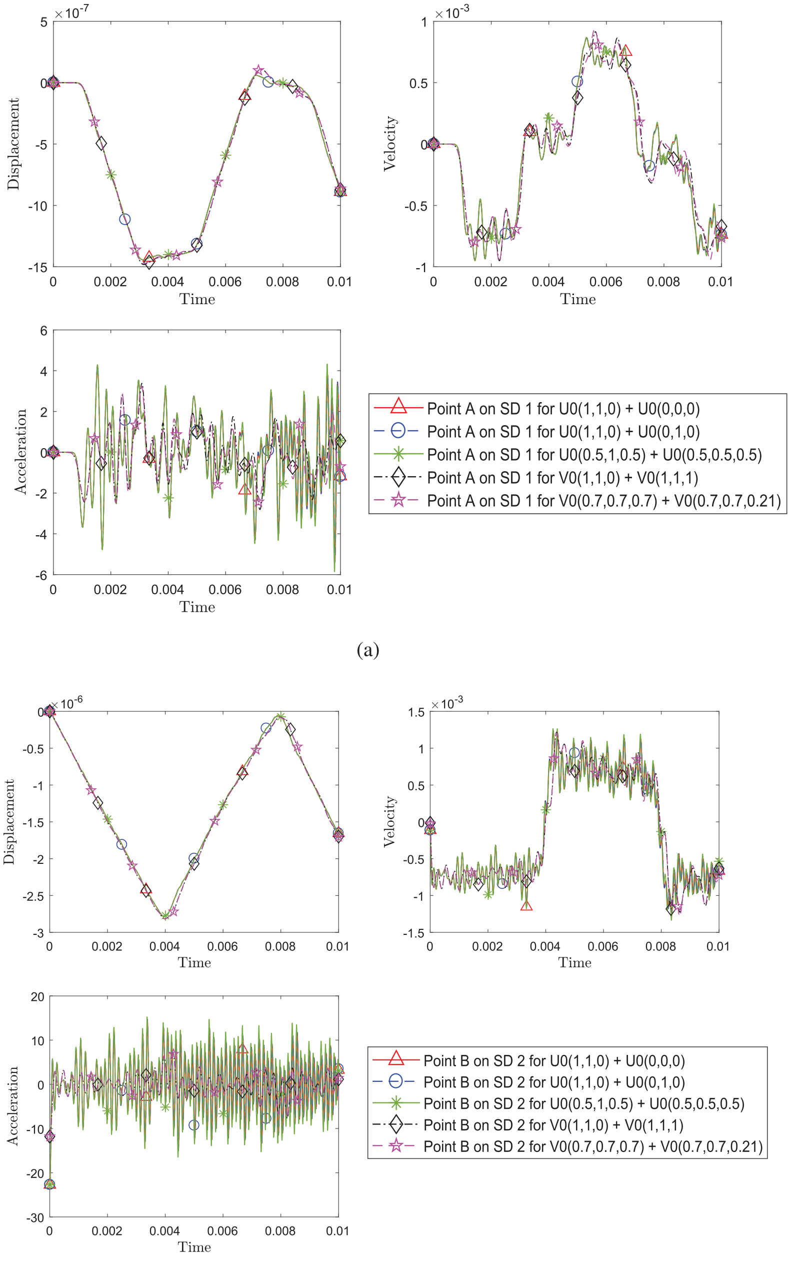
Figure 8: Comparison of transient response of displacement, velocity and acceleration (a) at point A on SD1 and (b) at point B on SD2, between three cases of different pairing of methods (Case 1: FEM+FEM, Case 2: FEM+GPS, Case 3: GPS+GPS) while using explicit version (with $\eta_3 = 0$) of GS4-II U0(1, 1, 0) algorithm for subomain 1 and explicit version (with $\eta_3 = 0$) of U0(0, 0, 0) algorithm for subdomain 2 with constant Δt in each case



(a) Point A on SD1

(b) Point B on SD2

Figure 9: Comparison of transient response of displacement, velocity and acceleration (a) at point A on SD1 and (b) at point B between different pairing of explicit (with $\eta_3 = 0$) and implicit version (with $\eta_3 = 1$) of GS4-II (Case 1: explicit+explicit, Case 2: explicit+implicit, Case 3: implicit+explicit) while using FEM with GS4-II U0(1, 1, 0) algorithm for subomain 1 and GPS with U0(0, 0, 0) algorithm for subdomain 2 at constant Δt in each case



(a)

(b)

Figure 10: Comparison of transient response of displacement, velocity and acceleration (a) at point A on SD1 and (b) at point B between different pairing of explicit version (with $\eta_3 = 0$) of GS4-II algorithms listed in Table 1, while using FEM for subdomain 1 and GPS for subdomain 2 at constant Δt in each case

Lastly, the subcycling cases are also evaluated in Fig. 11. A case where subdomain 2 has double the time steps than subdomain 1 (time increment interval in subdomain 2 is half of that of subdomain 1) is compared with the no subcycling case and the results are very close.

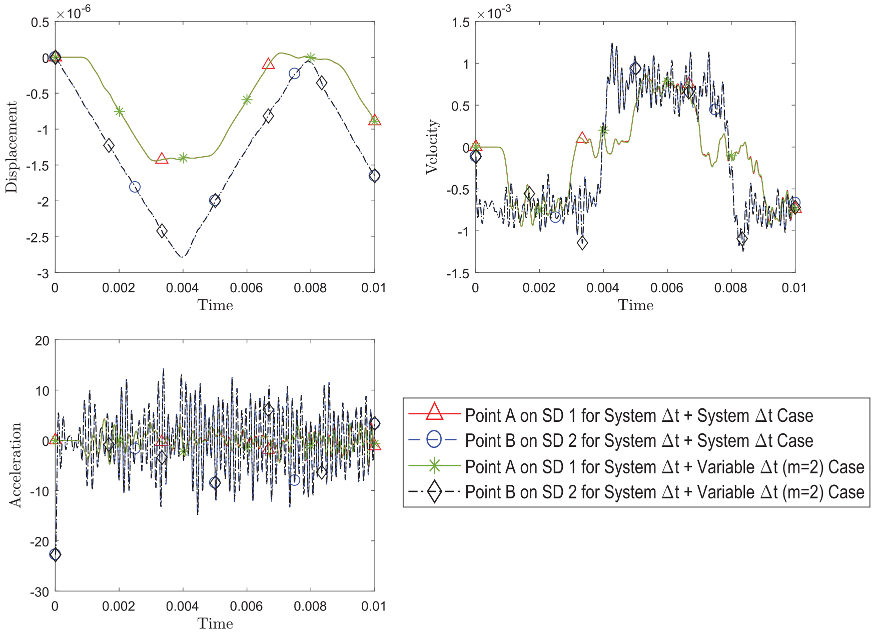
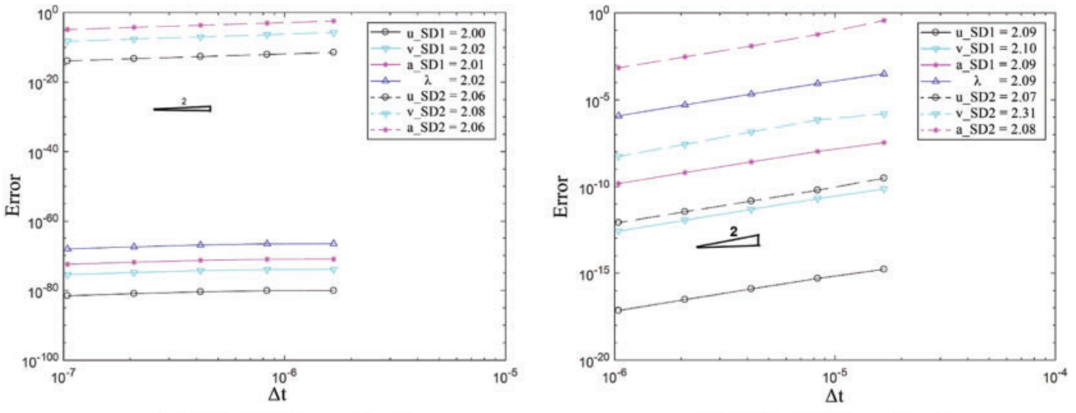


Figure 11: Comparison of transient response of displacement, velocity and acceleration at both point A and point B between simulation with system Δt for both subdomains and with variable $\Delta t = 1/2$ system Δt for subdomain 2 while using FEM with explicit version (with $\eta_3 = 0$) of GS4-II U0(1, 1, 0) algorithm for subdomain 1 and GPS with explicit version (with $\eta_3 = 0$) U0(0, 0, 0) algorithm for subdomain 2

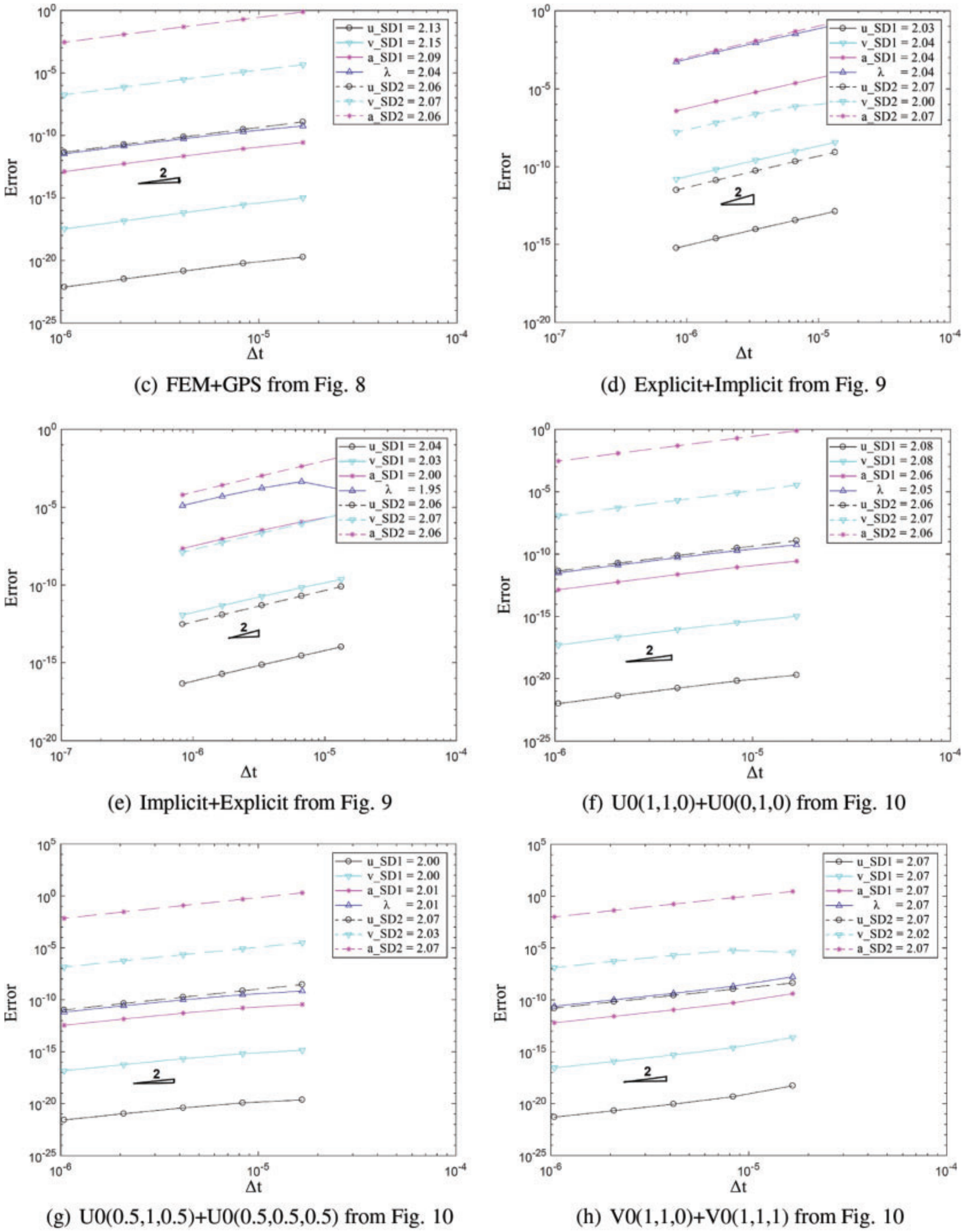
Fig. 12 demonstrates second order time accuracy in displacement, velocity, acceleration and the constraint variable λ* for different coupling conditions shown through Figs. 8–10.



(a) FEM+FEM from Fig. 8

(b) GPS+GPS from Fig. 8

Figure 12: (Continued)



(c) FEM+GPS from Fig. 8

(d) Explicit+Implicit from Fig. 9

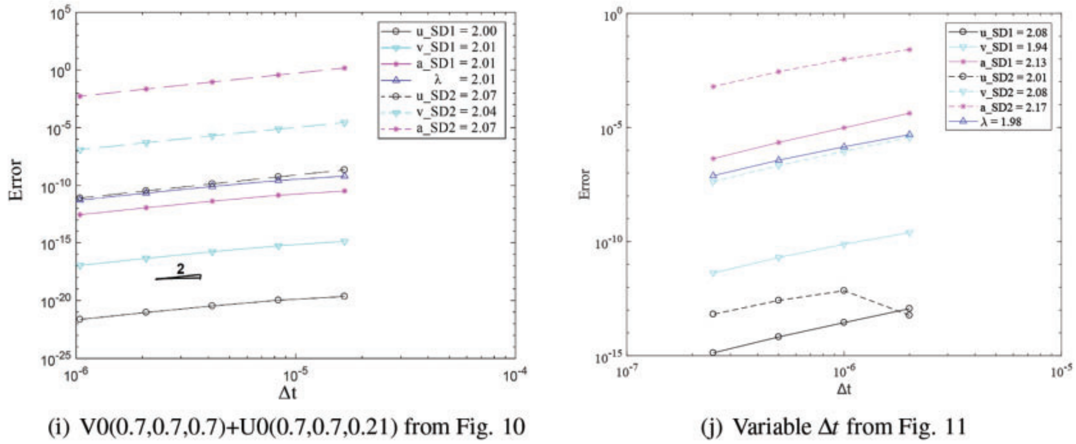
(e) Implicit+Explicit from Fig. 9

(f) U0(1,1,0)+U0(0,1,0) from Fig. 10

(g) U0(0.5,1,0.5)+U0(0.5,0.5,0.5) from Fig. 10

(h) V0(1,1,0)+V0(1,1,1) from Fig. 10

Figure 12: (Continued)



(i) V0(0.7,0.7,0.7)+U0(0.7,0.7,0.21) from Fig. 10

(j) Variable Δt from Fig. 11

Figure 12: Convergence order for the solution with respect to the system time step for domain 1 and 2 in different cases: (a) FEM with explicit version of GS4-II U0(1, 1, 0) and FEM with U0(0, 0, 0), (b) GPS with explicit version of GS4-II U0(1, 1, 0) and GPS with U0(0, 0, 0), (c) FEM with explicit version of GS4-II U0(1, 1, 0) and GPS with U0(0, 0, 0) algorithm, (d) FEM with explicit version of GS4-II U0(1, 1, 0) and GPS with implicit version U0(0, 0, 0), (e) FEM with implicit version of GS4-II U0(1, 1, 0) and GPS with explicit version U0(0, 0, 0), (f) FEM with explicit version of GS4-II U0(1, 1, 0) and FEM with implicit version U0(0, 1, 0), (g) FEM with explicit version of GS4-II U0(0.5, 1, 0.5) and GPS with U0(0.5, 0.5, 0.5), (h) FEM with explicit version of GS4-II V0(1, 1, 0) and GPS with V0(1, 1, 1), (i) FEM with explicit version of GS4-II V0(0.7, 0.7, 0.7) and GPS with V0(0.7, 0.7, 0.21), (j) variable $\Delta t = 1/2$ system Δt for subdomain 2 while using FEM with explicit version (with $\eta_3 = 0$) of GS4-II U0(1, 1, 0) algorithm for subdomain 1 and GPS with explicit version (with $\eta_3 = 0$) U0(0, 0, 0) algorithm for subdomain 2

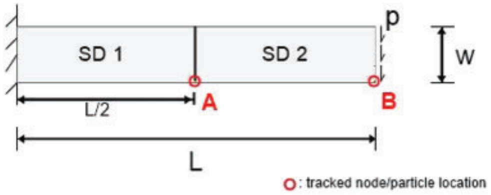


Figure 13: Diagram of structure for bending problem

5.2 2D Vibration Problem

A beam with the same geometry as the one in the wave propagation problem is used to illustrate the application of the multi-domain method in a plane stress dynamic problem. A cantilever beam of 10 m by 1 m with 1 m thickness is fixed on the left side and is experiencing downward traction, p, of 100,000 N/m as shown in Fig. 13. The density, ρ, of the beam is set to be 271.2 kg/m^3, Young's modulus, E, to be 72 GPa, and Poisson's ratio to be 0.3 for the properties of the material. The beam is divided into two subdomains, and each subdomain is implemented with 17 nodes/particles in length, and 5 nodes/particles in height making it total of 170 nodes/particles with 5 nodes/particles overlap on interface. The semidiscrete equation of motion for the system is given as

$$\mathbf{M}_i\ddot{\mathbf{u}}_i + \mathbf{K}_i\mathbf{u}_i = \mathbf{G}_i^T\lambda_i^* \qquad (99)$$

$$\sum_{i=1}^{i=N_{dom}} \mathbf{G}_i \mathbf{v}_i = 0 \tag{100}$$

$$\mathbf{u}(0, y, t) = 0 \tag{101}$$

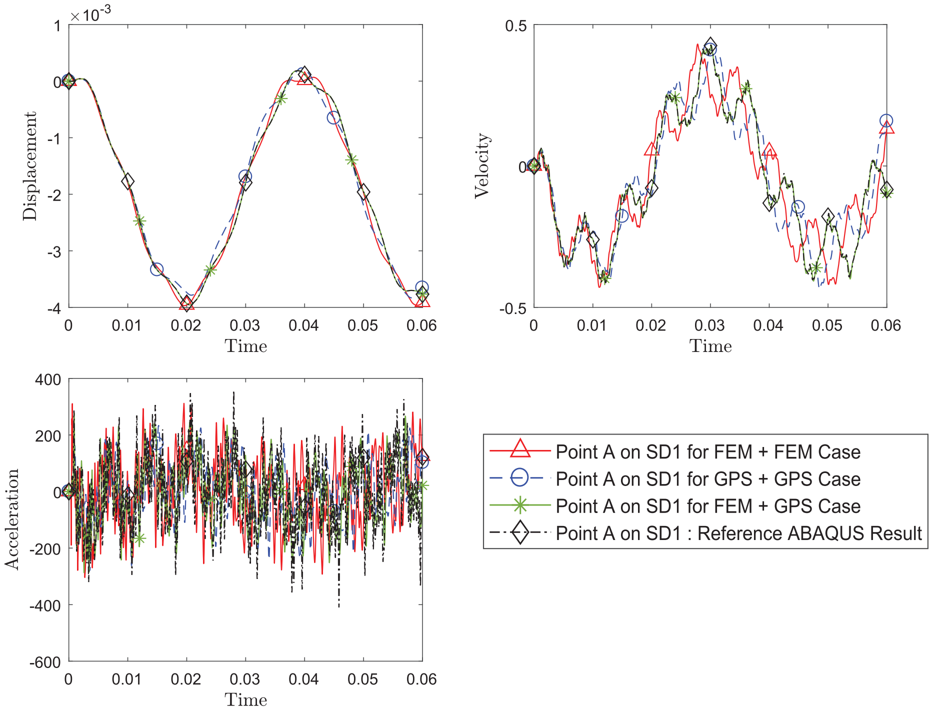
$$\frac{\partial \mathbf{u}}{\partial y}(L, y, t) = p \tag{102}$$

$$\mathbf{u}(x, y, 0) = 0 \tag{103}$$

$$\dot{\mathbf{u}}(x, y, 0) = 0 \tag{104}$$

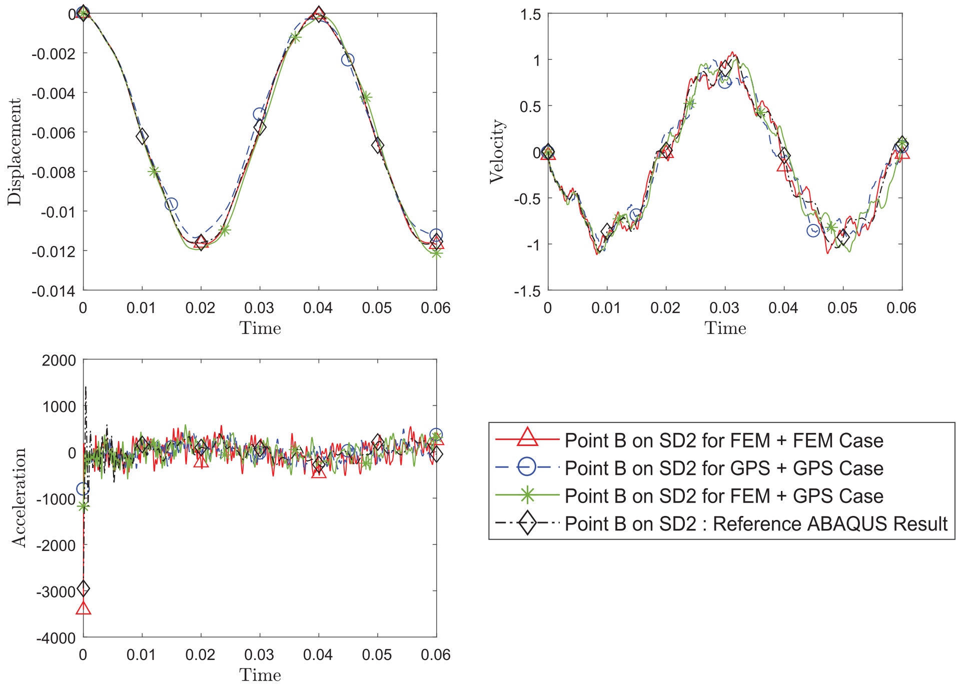
The system Δt for time integration is set to be 0.8 μs. The time history of y-direction displacement, velocity and acceleration of a node at the interface (point A) and the free end (point B) are evaluated.

Similar to the previous problem, a different combination of methods are compared in Fig. 14. It is apparent that the different methods can be coupled regardless of whether one is weak form or strong form.



(a)

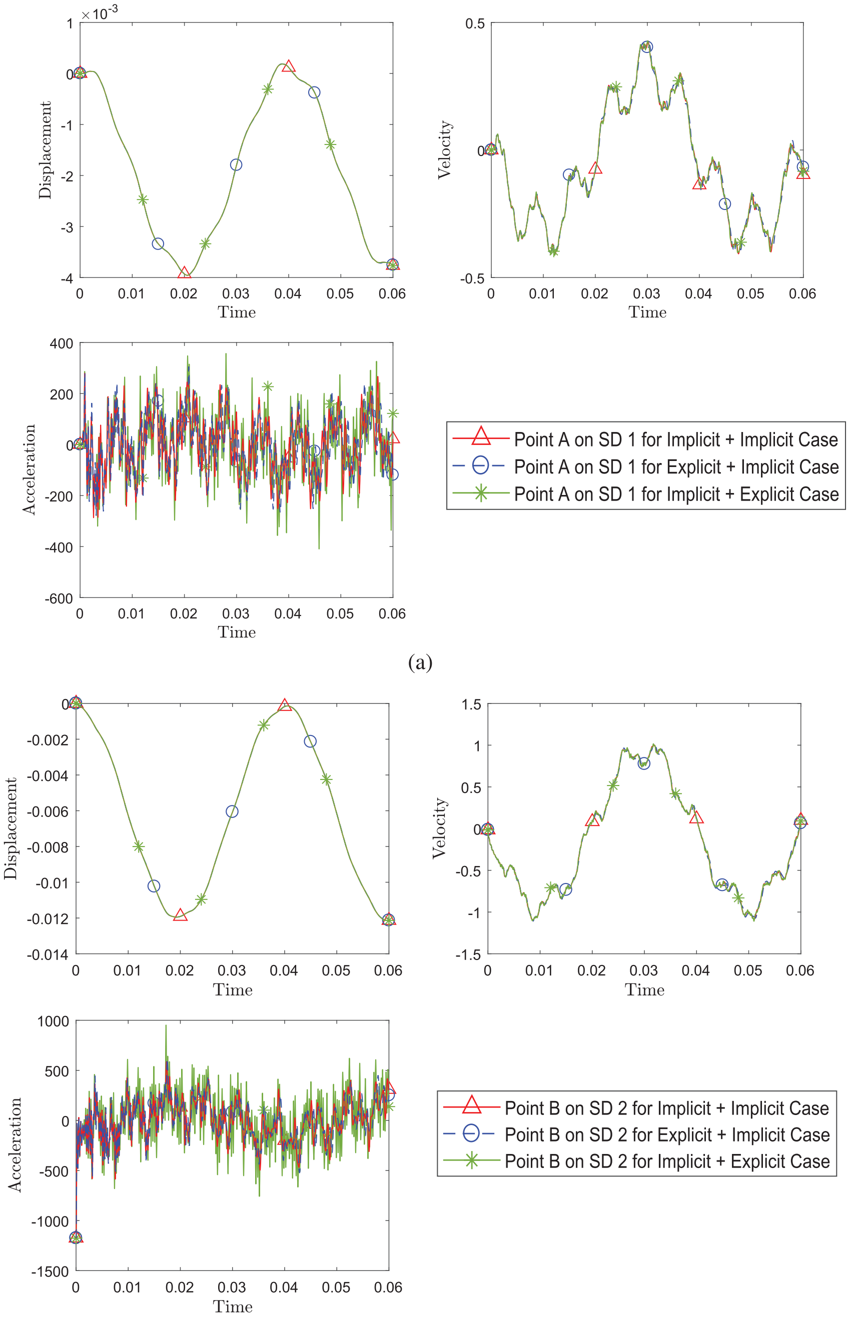
Figure 14: (Continued)



(b)

Figure 14: Comparison of transient response of displacement, velocity and acceleration (a) at point A on SD1 and (b) at point B between different pairing of methods (Case 1: FEM+FEM, Case 2: FEM+GPS, Case 3: GPS+GPS) while using implicit version (with $\eta_3 = 1$) of GS4-II U0(1, 1, 0) algorithm for subdomain 1 and implicit version of U0(0, 0, 0) algorithm for subdomain 2 with constant Δt in each case

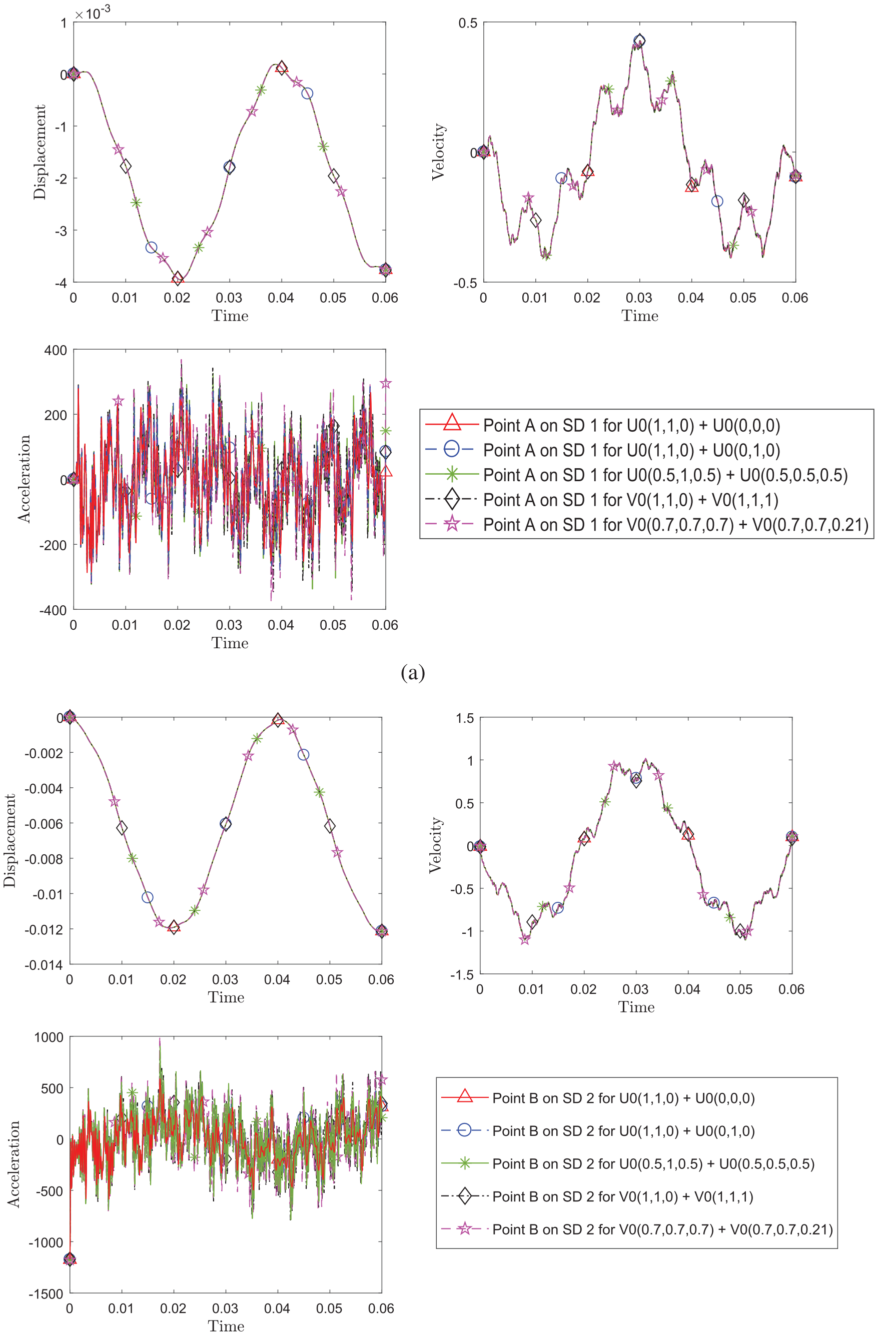
Fig. 15 shows results for coupling the explicit version and implicit version of the time scheme algorithms. The evaluated coupling cases this time are the implicit and implicit case, explicit and implicit case, and implicit and explicit case. The response results for the different cases align closely. The same five cases of time scheme algorithm combinations used in the previous example are evaluated and illustrated in Fig. 16. And the results for all cases match. The subcycling case with one domain having half the Δt as the other, is shown in Fig. 17. It is apparent that subcycling still produces an accurate result. Lastly, all of the different cases and conditions are second order accurate in time as evident from Fig. 18, illustrating slope of two in the log-log error convergence plot.



(a)

(b)

Figure 15: Comparison of transient response of displacement, velocity and acceleration (a) at point A on SD1 and (b) at point B between different pairing of explicit (with $\eta_3 = 0$) and implicit version (with $\eta_3 = 1$) of GS4-II (Case 1: implicit+implicit, Case 2: explicit+implicit, Case 3: implicit+implicit) while using FEM with GS4-II U0(1, 1, 0) algorithm for subdomain 1 and GPS with U0(0, 0, 0) algorithm for subdomain 2 at constant Δt in each case



(a)

(b)

Figure 16: Comparison of transient response of displacement, velocity and acceleration (a) at point A on SD1 and (b) at point B between different pairing of implicit version (with $\eta_3 = 1$) of GS4-II algorithms listed in Table 1 while using FEM for subdomain 1 and GPS for subdomain 2 at constant Δt in each case

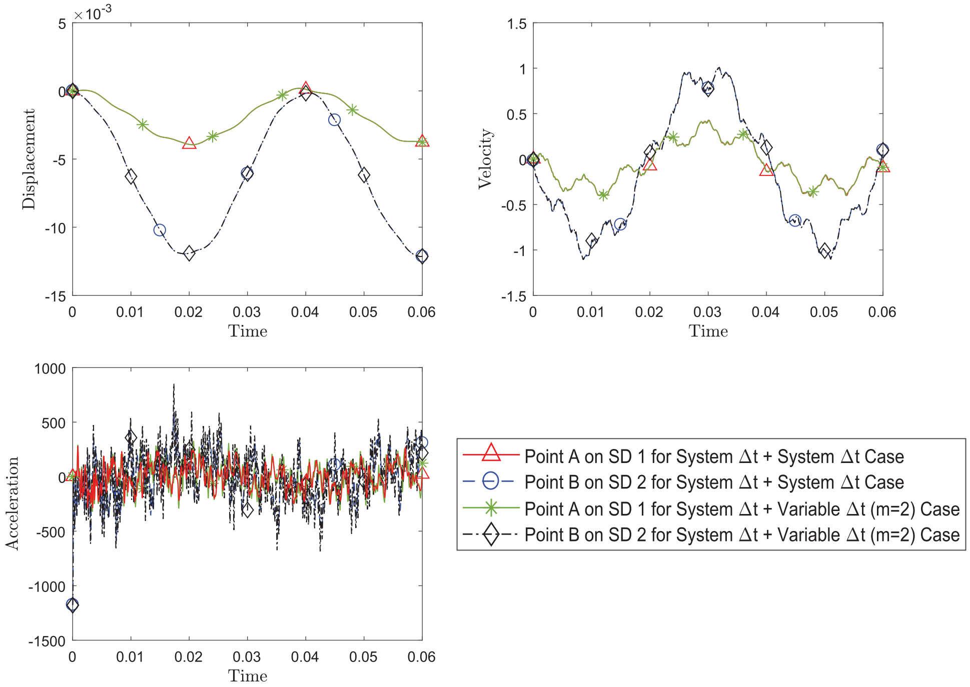
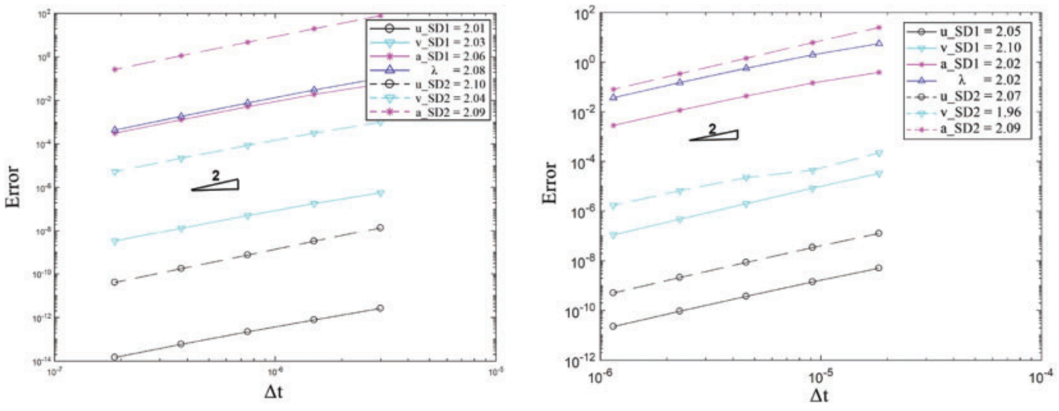


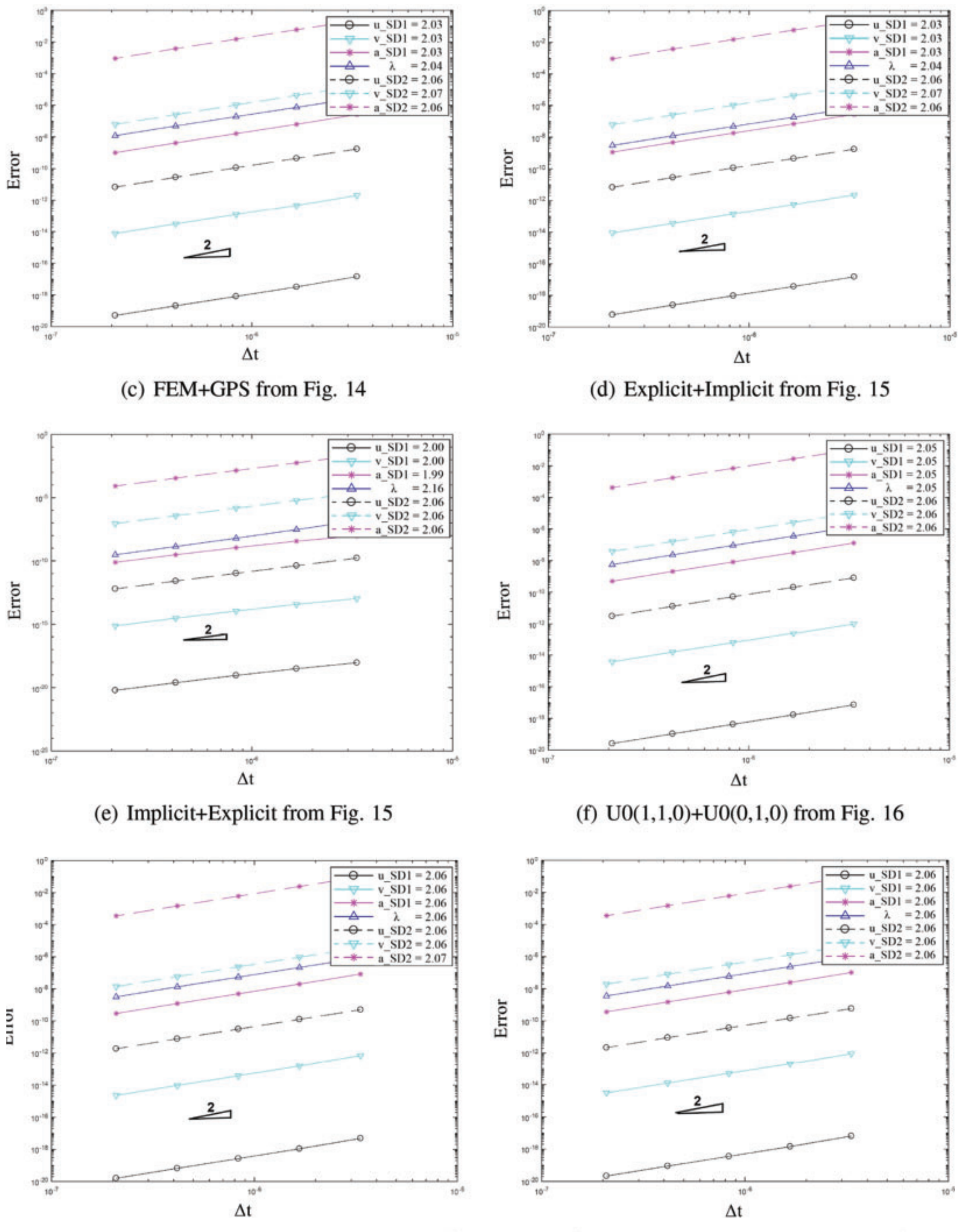
Figure 17: Comparison of transient response of displacement, velocity and acceleration at both point A and point B, between simulation with system Δt for both subdomains and with variable $\Delta t = 1/2$ system Δt for subdomain 2 while using FEM with implicit version (with $\eta_3 = 1$) of GS4-II U0(1, 1, 0) algorithm for subdomain 1 and GPS with implicit version (with $\eta_3 = 1$) U0(0, 0, 0) algorithm for subdomain 2



(a) FEM+FEM from Fig. 14

(b) GPS+GPS from Fig. 14

Figure 18: (Continued)



(c) FEM+GPS from Fig. 14

(d) Explicit+Implicit from Fig. 15

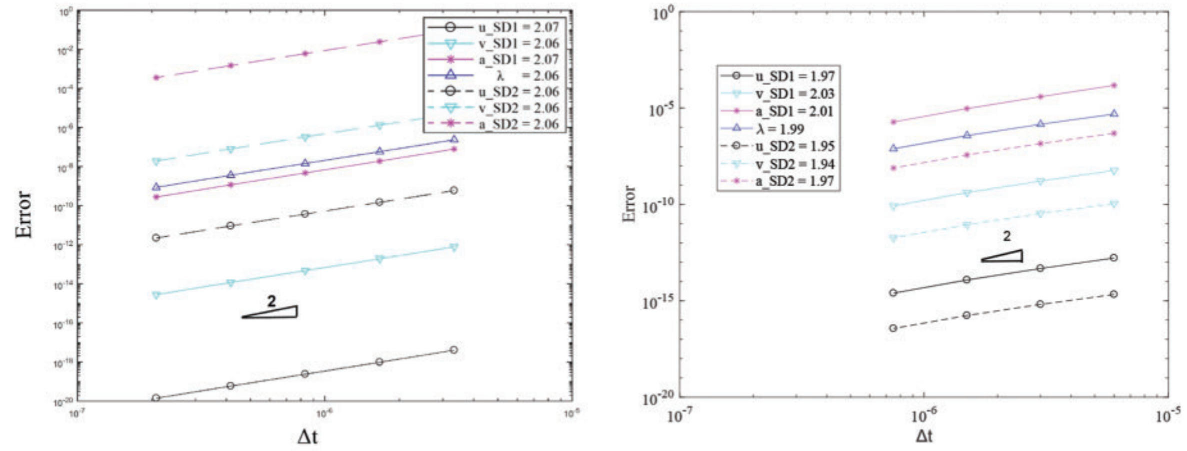
(e) Implicit+Explicit from Fig. 15

(f) U0(1,1,0)+U0(0,1,0) from Fig. 16

(g) U0(0.5,1,0.5)+U0(0.5,0.5,0.5) from Fig. 16

(h) V0(1,1,0)+V0(1,1,1) from Fig. 16

Figure 18: (Continued)



(i) V0(0.7,0.7,0.7)+U0(0.7,0.7,0.21) from Fig. 16

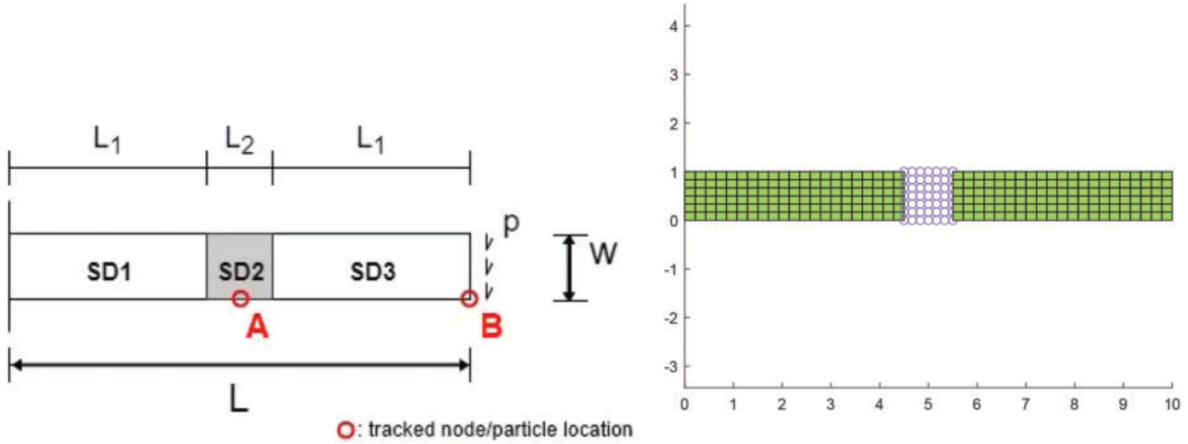
(j) variable Δt from Fig. 17

Figure 18: Convergence order for the solution with respect to the system time step for domain 1 and 2 in different cases: (a) FEM with implicit version of GS4-II U0(1, 1, 0) and FEM with U0(0, 0, 0), (b) GPS with implicit version of GS4-II U0(1, 1, 0) and GPS with U0(0, 0, 0), (c) FEM with implicit version of GS4- II U0(1, 1, 0) and GPS with U0(0, 0, 0), (d) FEM with explicit version of GS4-II U0(1, 1, 0) and GPS with implicit version U0(0, 0, 0), (e) FEM with implicit version of GS4-II U0(1, 1, 0) and GPS with explicit version U0(0, 0, 0), (f) FEM with implicit version of GS4-II U0(1, 1, 0) and FEM with U0(0, 1, 0), (g) FEM with implicit version of GS4-II U0(0.5, 1, 0.5) and GPS with U0(0.5, 0.5, 0.5) algorithm for subdomain 2, (h) FEM with implicit version of GS4-II V0(1, 1, 0) and GPS with V0(1, 1, 1), (i) FEM with implicit version of GS4-II V0(0.7, 0.7, 0.7) and GPS with V0(0.7, 0.7, 0.21), (j) variable $\Delta t_2 = 1/2$ system Δt for subdomain 2 while using FEM with implicit version (with $\eta_3 = 1$) of GS4-II U0(1, 1, 0) algorithm for subdomain 1 and GPS with implicit U0(0, 0, 0) algorithm for subdomain 2

5.3 2D Vibration Problem-Three Subdomains

Next, a body divided into three domains is examined next for the extension of the subdomain coupling concept. The same vibration problem above is evaluated again but this time, the beam is divided into three subdomains where $L_2 = \boldsymbol{L}/10$. The FEM is used on subdomains 1 and 3 with 22 nodes in length, and 7 nodes in height. GPS is implemented on subdomain 2, with 7 particles in length and 7 particles in height. The time scheme algorithm used in subdomain 1 and 3 is implicit U0(1, 1, 0); and in subdomain 2, U0(0, 0, 0) is used. The time history of y-direction displacement, velocity and acceleration of a node at the free end is evaluated.

Figs. 20 and 21 clearly illustrate that the result at points A and B (shown in Fig. 19a) from the three subdomain case matches closely with the result from the two subdomain case while achieving second order time convergence rate. The high frequency in the acceleration plot for the three domain free end node is caused by the use of non-dissipative U0(1, 1, 0) scheme.



(a) Diagram of three domain structure for bending problem

(b) Mesh of the problem

Figure 19: (a) Structure divided into three domains (b) An example of mesh of the structure: FEM GPS FEM

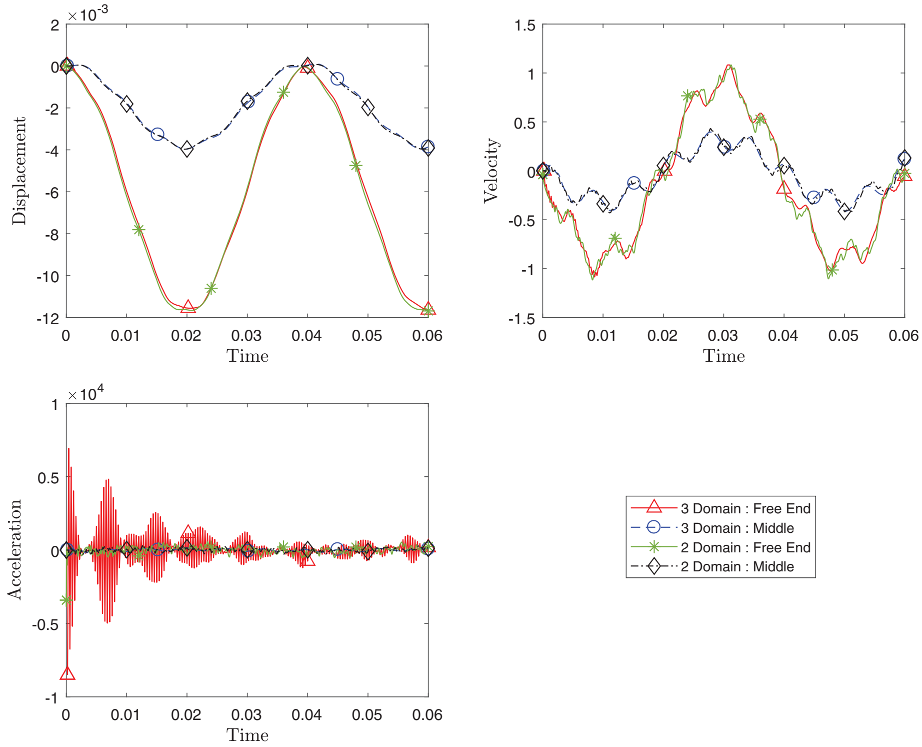
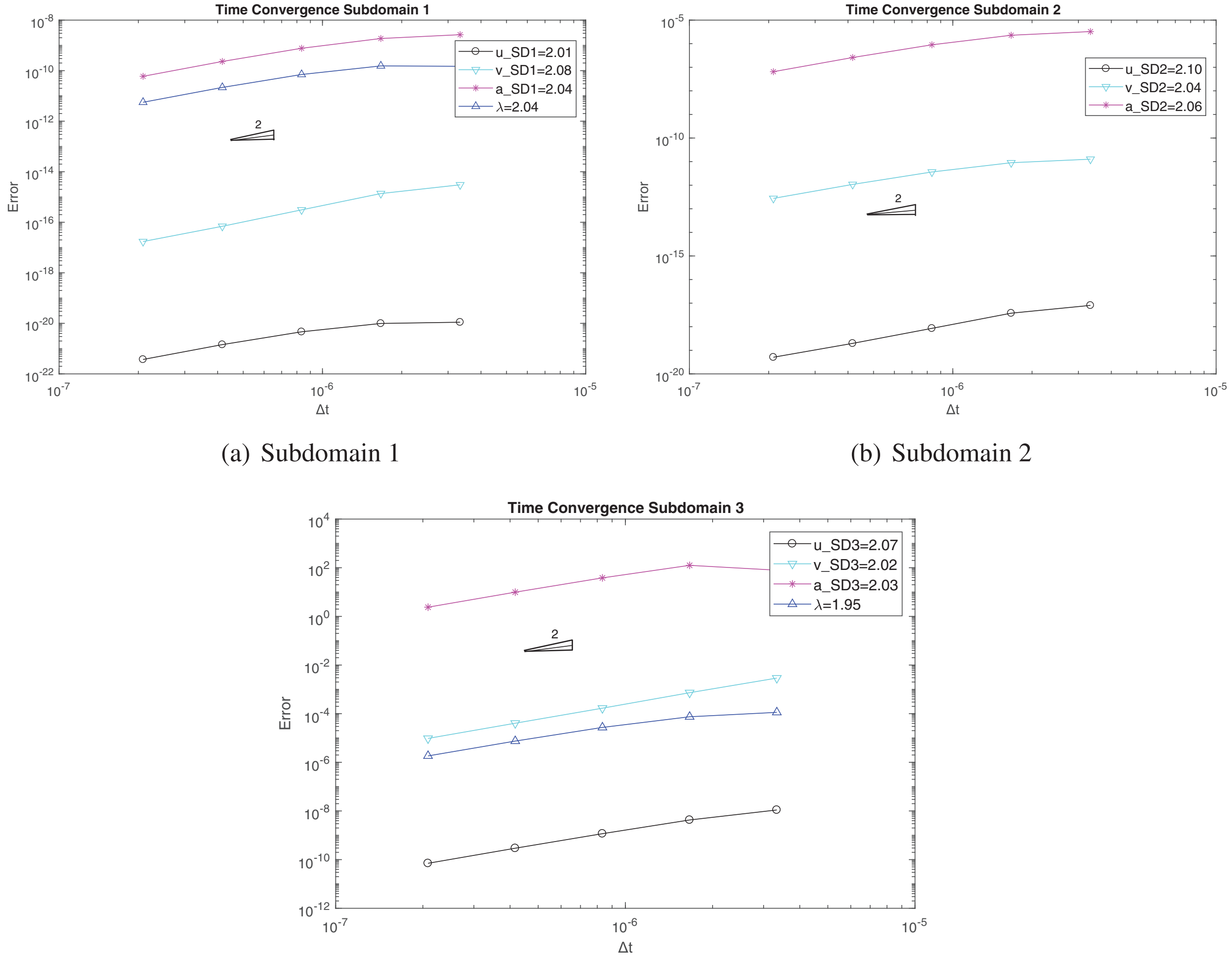


Figure 20: Comparison of transient response of displacement, velocity and acceleration at point A and point B between three subdomain configuration and two subdomain configuration while using FEM with implicit version (with $\eta_3 = 1$) of GS4-II U0(1, 1, 0) algorithm for subdomain 1 and subdomain 3 and GPS with implicit version of U0(0, 0, 0) algorithm for subdomain 2 with constant Δt in all subdomains



(a) Subdomain 1

(b) Subdomain 2

(c) Subdomain 3

Figure 21: Convergence order for the solution with respect to the system time step for domain 1 and 2 and 3 in three domain configuration case: (a) subdomain 1 using FEM with implicit version of GS4-II U0(1, 1, 0) algorithm, (b) subdomain 2 using GPS with U0(0, 0, 0) algorithm, (c) subdomain 3 using FEM with GS4-II U0(1, 1, 0) algorithm

5.4 2D Notched Specimen Problem

Figs. 22 and 23 show the geometry of a 2D plane stress problem and an example of the mesh, respectively. It consists of three subdomains where identical subdomain 1 and 3 are connected by a thin subdomain 2. The subdomain 1 is 3 m by 2 m rectangle which are discretized into 176 FEM nodes. On the other hand, subdomain 2 is 1 m by 2 m rectangle bar, with notches on top and bottom, discretized into 99 particles. And subdomain 3 is identical to the subdomain 1 geometrically, but discretized with

176 particles. The whole geometry has thickness of 1 m. The semidiscrete equation of motion for the system is given as

$$
\begin{aligned}
&\rho_i \ddot{\mathbf{u}}_i + \nabla \cdot \mathbf{D} \nabla \mathbf{u}_i = \mathbf{G}_i^T \lambda_i^* \\
&\sum_{i=1}^{i=N_{dom}} \mathbf{G}_i \mathbf{v}_i = 0 \\
&\mathbf{u}(0, y, t) = 0 \\
&\frac{\partial \mathbf{u}}{\partial x}(L, y, t) = -210 \\
&\mathbf{u}(x, y, 0) = 0 \\
&\dot{\mathbf{u}}(x, y, 0) = 0
\end{aligned}
\tag{105}
$$

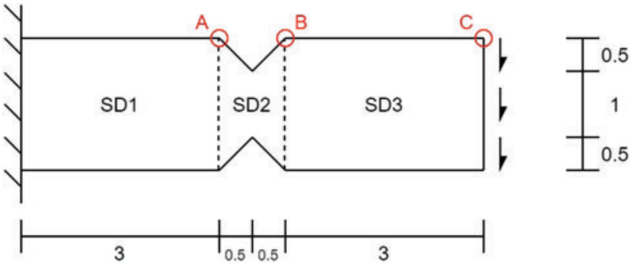


Figure 22: Diagram of structure divided into three subdomain. The red circles represent the locations where **u**, **u̇**, and **ü** are tracked in each subdomain

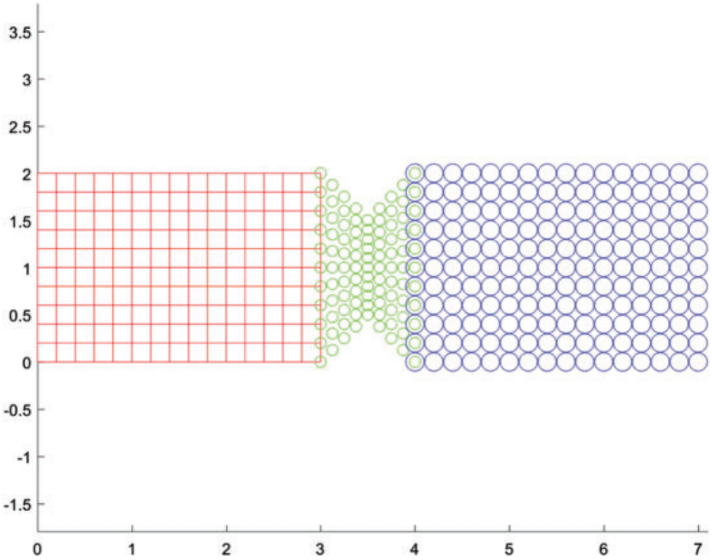


Figure 23: Mesh of all three subdomains

The bar is fixed at the left end and pulled down from the right end with a distributed load of 210 N/m. This material has 70 GPa for Young's modulus, 0.3 for Poisson's ratio, and 1,000 kg/m^3 for density. The simulation ran for 0.2 s with the system Δt for time integration set to 0.4 ms. To demonstrate the coupling of different spatial methods and different time schemes, four cases are studied as shown in Table 2.

Table 2: Coupling different time scheme algorithms

Case	Subdomain 1	Subdomain 2	Subdomain 3
1	FEM U0(1, 1, 0)	Peridynamics U0(0, 0, 0)	MPS U0(1, 1, 0)
2	FEM U0(0.8, 0.5, 0.1)	Peridynamics U0(0.6, 0.3, 0.1)	MPS U0(1, 0.7, 0.1)
3	FEM V0(0.8, 0.5, 0)	Peridynamics V0(0.8, 0.5, 0.2)	MPS V0(0.8, 0.5, 0.4)
4	FEM U0(1, 1, 0.7)	Peridynamics U0(1, 1, 0.1)	MPS U0(1, 1, 0.6)

The time history of the y-direction displacement, velocity and acceleration of nodes/particles in each subdomain (points A, B, and C in Fig. 22) are evaluated.

Figs. 24–26 show the transient response of the displacement, velocity and acceleration for points A, B and C for all cases listed in the Table 2 compared with the reference results. All responses from the different spatial methods and different time schemes combinations match well.

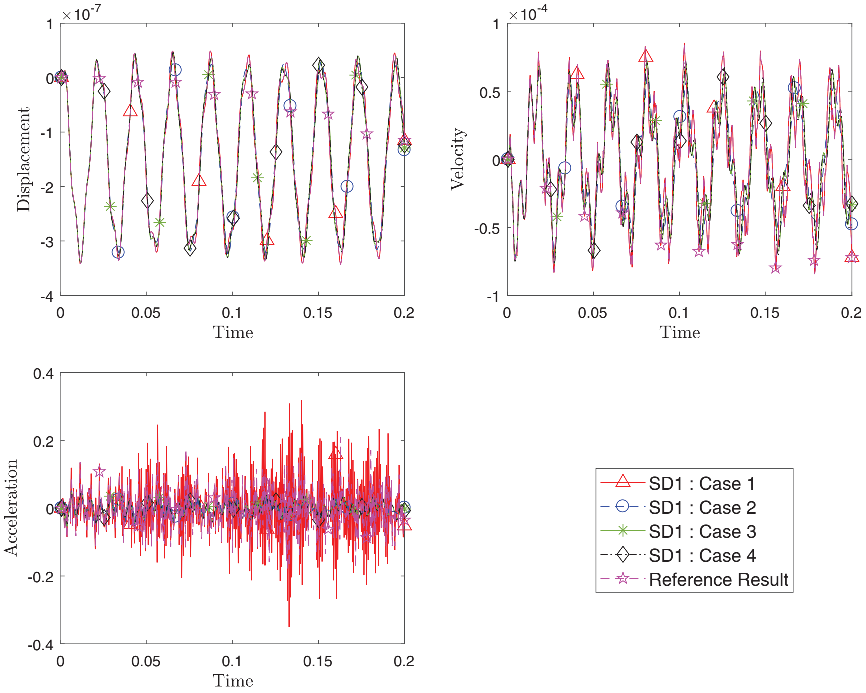


Figure 24: Comparison of transient response of displacement, velocity and acceleration at point A between different cases with varying method and time integration schemes for each domain as listed in Table 2

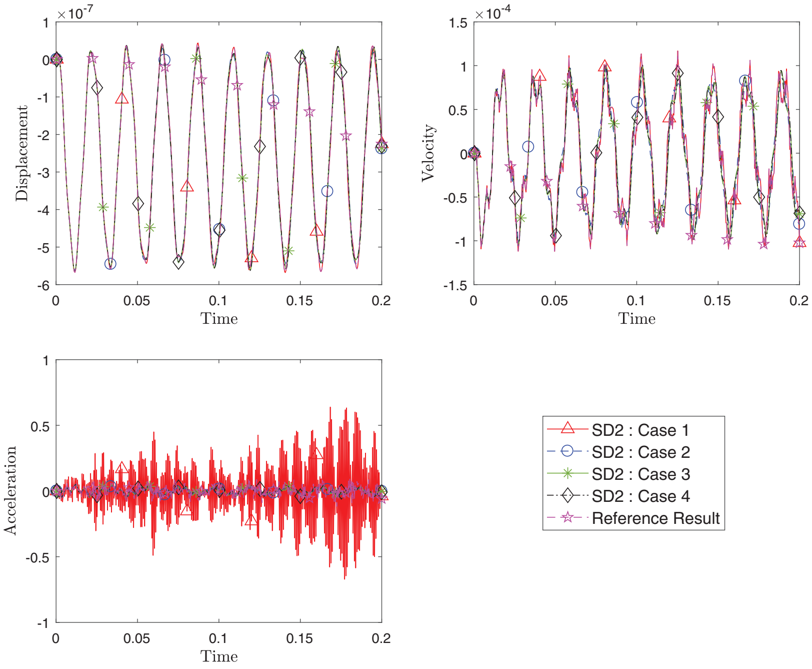


Figure 25: Comparison of transient response of displacement, velocity and acceleration at point B among different cases with varying method and time integration schemes for each domain as listed in Table 2

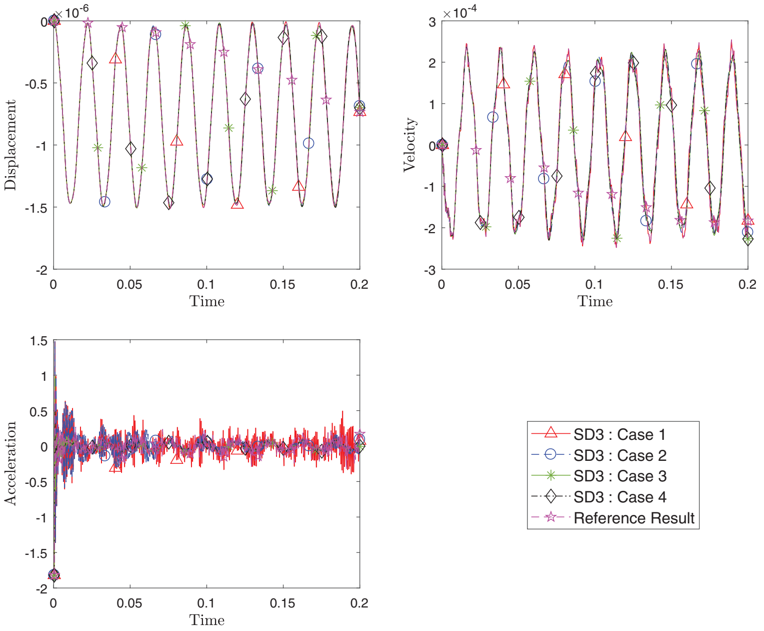


Figure 26: Comparison of transient response of displacement, velocity and acceleration at point C among different cases with varying method and time integration schemes for each domain as listed in Table 2

6 Conclusions

In this paper, we have proposed a novel implementation of the Generalized Single Step Single Solve time integration framework for ODE's into the DAE framework. This proposed method, unlike traditional approaches in current technology with LMS methods, allows the coupling of different numerical analysis methods such as FEM and particle methods to be used on a single body while ensuring the accuracy of the result with the use of a wide variety of time integration algorithms within the GS4 framework. For example, one cannot simply or haphazardly couple the Midpoint rule with the Newmark method; or couple a multitude of schemes such as the Newmark, Midpoint rule, HHT, WBZ methods, etc., arbitrarily in different subdomains in a single analysis without loss and reduced accuracy in the convergence rate of one or more variables. However the GS4 framework circumvents various deficiencies, preserves accuracy, and permits a much broader design space for coupling algorithms in a single analysis and provides much desired robust features for applications to large scale industrial real world problems as well. It has the potential to increase the accuracy of the physics by selecting an appropriate method for only the area with a localized phenomena rather than utilizing only a single method that may not be suitable for certain applications on the whole body. The method is verified by applying it to 2D simple bar and beam problems. The results from various combination of methods and time scheme algorithms match closely with the reference result. With the basis of the proposed computational methodology established, the extension of the method for the future studies includes computational efficiency studies, extension to nonlinearity, and implementation of reduced order modeling. This work provides generality and versatility of the computational framework incorporating a wide variety of subdomain based spatial and time integration algorithms in a single analysis with great accuracy.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Gingold, R. A., Monaghan, J. J. (1977). Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society, 181(3),* 375–389. DOI 10.1093/mnras/181.3.375.
2. Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal, 82,* 1013–1024. DOI 10.1086/112164.
3. Benz, W. (1990). Smooth particle hydrodynamics: A review. In: *The numerical modelling of nonlinear stellar pulsations*, pp. 269–288. Dordrecht: Springer Netherlands.
4. Liu, G. R., Liu, M. B. (2003). *Smoothed particle hydrodynamics: A meshfree particle method.* Singapore: World Scientific.
5. Koshizuka, S., Oka, Y. (1996). Moving-particle semi-implicit method for fragmentation of imcompressible fluid. *Nuclear Science and Engineering, 123,* 421–434. DOI 10.13182/NSE96-A24205.
6. Koshizuka, S., Nobe, A., Oka, Y. (1998). Numerical analysis of breaking waves using the moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids, 26,* 751–769. DOI 10.1002/(ISSN)1097-0363.
7. Silling, S. A. (2000). Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids, 48(1),* 175–209. DOI 10.1016/S0022-5096(99)00029-0.
8. Silling, S. A., Lehoucq, R. B. (2010). *Peridynamic theory of solid mechanics*, vol. 44. USA: Elsevier.

9. Seleson, P., Parks, M., Gunzburger, M., Lehoucq, R. B. (2009). Peridynamics as an upscaling of molecular dynamics. *Multiscale Modeling & Simulation, 8(1),* 204–227. DOI 10.1137/09074807X.

10. Tae, D., Tamma, K. K. (2018). Mixed strong form representation particle method for solids and structures. *Journal of Applied and Computational Mechanics*, *4, 4,* 429–441.

11. Ozaki, S., Hashiguchi, K., Okayasu, T., Chen, D. (2007). Finite element analysis of particle assembly-water coupled frictional contact problem. *Computer Modeling in Engineering & Sciences, 18(2),* 101–120. DOI 10.3970/cmes.2007.018.101.

12. Bardenhagen, S. G., Kober, E. M. (2004). The generalized interpolation material point method. *Computer Modeling in Engineering & Sciences, 5(6),* 477–496. DOI 10.3970/cmes.2004.005.477.

13. Sladek, J., Stanak, P., Han, Z., Sladek, V., Atluri, S. N. (2013). Applications of the MLPG method in engineering & sciences: A review. *Computer Modeling in Engineering & Sciences, 92(5),* 423–475. DOI 10.3970/cmes.2013.092.423.

14. Li, J., Wang, X., Lee, J. D. (2012). Multiple time scale algorithm for multiscale material modeling. *Computer Modeling in Engineering & Sciences, 85(5),* 463–480. DOI 10.3970/cmes.2012.085.463.

15. Orsini, P., Power, H., Lees, M. (2011). The hermite radial basis function control volume method for multi-zones problems; A non-overlapping domain decomposition algorithm. *Computer Methods in Applied Mechanics and Engineering, 200(5),* 477–493. DOI 10.1016/j.cma.2010.05.001.

16. Metsis, P., Papadrakakis, M. (2012). Overlapping and non-overlapping domain decomposition methods for large-scale meshless EFG simulations. *Computer Methods in Applied Mechanics and Engineering, 229,* 128–141. DOI 10.1016/j.cma.2012.03.012.

17. Park, K., Felippa, C. (2000). A variational principle for the formulation of partitioned structural systems. *International Journal for Numerical Methods in Engineering, 47(1–3),* 395–418. DOI 10.1002/(ISSN)1097-0207.

18. Brezzi, F., Marini, L. (2005). The three-field formulation for elasticity problems. *GAMM-Mitteilungen, 28(2),* 124–153. DOI 10.1002/gamm.201490016.

19. Xiao, S. P., Belytschko, T. (2004). A bridging domain method for coupling continua with molecular dynamics. *Computer Methods in Applied Mechanics and Engineering, 193(17–20),* 1645–1669. DOI 10.1016/j.cma.2003.12.053.

20. Cundall, P. A., Peter, A., Strack, O. (1979). A discrete numerical model for granular assemblies. *Geotechnique, 29(1),* 47–65. DOI 10.1680/geot.1979.29.1.47.

21. Morris, J., Rubin, M. B., Block, G. I., Bonner, M. P. (2006). Simulations of fracture and fragmentation of geologic materials using combined FEM/DEM analysis. *International Journal of Impact Engineering, 33(1–12),* 463–473. DOI 10.1016/j.ijimpeng.2006.09.006.

22. Kozicki, J., Donze, F. V. (2008). A new open-source software developed for numerical simulations using discrete modeling methods. *Computer Methods in Applied Mechanics and Engineering, 197(49–50),* 4429–4443. DOI 10.1016/j.cma.2008.05.023.

23. Macek, R. W., Silling, S. A. (2007). Peridynamics via finite element analysis. *Finite Elements in Analysis and Design, 43(15),* 1169–1178. DOI 10.1016/j.finel.2007.08.012.

24. Kilic, B., Madenci, E. (2010). Coupling of peridynamic theory and the finite element method. *Journal of Mechanics of Materials and Structures, 5(5),* 707–733. DOI 10.2140/jomms.

25. Zaccariotto, M., Mudric, T., Tomasi, D., Shojaei, A., Galvanetto, U. (2018). Coupling of FEM meshes with peridynamic grids. *Computer Methods in Applied Mechanics and Engineering, 330,* 471–497. DOI 10.1016/j.cma.2017.11.011.

26. Gravouil, A., Combescure, A. (2001). Multi-time-step explicit-implicit method for non-linear structural dynamics. *International Journal for Numerical Methods in Engineering, 50(1),* 199–225. DOI 10.1002/(ISSN)1097-0207.

27. Pegon, P. (2008). Continuous PsD testing with substructuring. In: *Modern testing techniques for structural systems*, pp. 197–257. Vienna: Springer Vienna.

28. Bursi, A. B. O. S., He, L., Magonette, G., Pegon, P. (2008). Convergence analysis of a parallel interfield method for heterogeneous simulations with dynamic substructuring. *International Journal for Numerical Methods in Engineering, 75(7),* 800–825. DOI 10.1002/nme.2285.

29. Prakash, A., Hjelmstad, K. (2004). A FETI-based Multi-time-step Coupling Method for Newmark Schemes in Structural Dynamics.. *International Journal for Numerical Methods in Engineering, 61(13),* 2183–2204.

30. Karimi, S., Nakshatrala, K. B. (2014). On multi-time-step monolithic coupling algorithms for elastodynamics. *Journal of Computational Physics, 273,* 671–705. DOI 10.1016/j.jcp.2014.05.034.

31. Nakshatrala, K. B., Hjelmstad, K. D., Tortorelli, D. A. (2008). A FETI-based domain decomposition technique for time-dependent first-order systems based on a DAE approach. *International Journal for Numerical Methods in Engineering, 75(12),* 1385–1415. DOI 10.1002/nme.2303.

32. Shimada, M. (2014). *Novel design and development of isochronous time integration architectures for ordinary differential equations and differential-algebraic equations: Computational science and engineering applications (Ph.D. Thesis)*. The University of Minnesota-Twin Cities, Minneapolis, Minnesota, USA.

33. Maxam, D., Deokar, R., Tamma, K. K. (2019). A unified computational methodology for dynamic thermoelasticity with multiple subdomains under the GSSSS framework involving differential algebraic equation systems. *Journal of Thermal Stresses, 42(1),* 163–184. DOI 10.1080/01495739.2018.1536869.

34. Tamma, K. K., Zhou, X., Sha, D. (2000). The time dimension: A theory towards the evolution, classification, characterization and design of computational algorithms for transient/dynamic applications. *Archives of Computational Methods in Engineering, 7(2),* 67–290.

35. Zhou, X., Tamma, K. (2006). Algorithms by design with illustrations to solid and structural mechanics/dynamics. *International Journal for Numerical Methods in Engineering, 66(11),* 1738–1790. DOI 10.1002/(ISSN)1097-0207.

36. Zhou, X., Tamma, K. K. (2004). Design, analysis, and synthesis of generalized single step single solve and optimal algorithms for structural dynamics. *International Journal for Numerical Methods in Engineering, 59,* 597–668. DOI 10.1002/(ISSN)1097-0207.

37. Har, J., Tamma, K. K. (2012). *Advances in computational dynamics of particles, materials and structures*. Chichester, UK: John Wiley & Sons.

38. Maxam, D. J., Tamma, K. K. (2022). A re-evaluation of overshooting in time integration schemes: The neglected effect of physical damping in the starting procedure. *International Journal for Numerical Methods in Engineering, 123(12),* 2683–2704. DOI 10.1002/nme.6955.