Computers and Structures 133 (2014) 51–63

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Computationally efficient multi-time-step method for partitioned time integration of highly nonlinear structural dynamics

Arun Prakash [a,*], Ertugrul Taciroglu [b], Keith D. Hjelmstad [c]

[a] School of Civil Engineering, Purdue University, United States
[b] Civil & Environmental Engineering, Univ. of California, Los Angeles, United States
[c] School of Sustainable Engineering & the Built Environment, Arizona State University, United States

ARTICLE INFO

Article history:
Received 23 January 2013
Accepted 27 November 2013
Available online 25 December 2013

Keywords:
Time integration
Structural dynamics
Domain decomposition
Multi-time-step
Multi-scale
Asynchronous integrators

ABSTRACT

An efficient and accurate method for solving large-scale problems in non-linear structural dynamics is presented. The method uses dual-Schur domain decomposition to divide a large finite element mesh into a number of smaller subdomains, which are solved independently using a suitable mesh-size and time-step to capture the local spatial and temporal scales of the problem. Continuity of the solution between subdomains is enforced by Lagrange multipliers. It is shown that the proposed method is stable, accurate and computationally more efficient than using a uniform time-step for the entire mesh. Numerical examples are presented to illustrate and corroborate these properties.

1. Introduction

Non-linear structural dynamics of a large and complex structural system usually involves capturing not only the global response of the structure but also small scale local phenomena such as crack propagation, plastic yielding etc. This requires a very fine spatial and temporal discretization in certain regions of interest within the domain whereas a relatively coarser discretization suffices elsewhere. Traditionally, such problems have been solved by using a finite element discretization in space and a finite difference time stepping scheme for numerical time integration. However, most methods in the current literature are formulated using a uniform time step for the entire mesh. This is computationally very expensive for problems where capturing physical phenomena at multiple temporal scales is important.

Researchers in different fields have explored several approaches to overcome the limitations of uniform time stepping methods. The text by Hairer, Lubich and Wanner [1] (see section VIII.4 for details) summarizes some of the earliest attempts to split the fast and slow temporal scales in ordinary differential equations (ODEs) using *multi-rate* methods [2]. These methods have gained popularity for applications where the computational problem size is usually very large such as astronomy and molecular dynamics.

In structural and solid mechanics, an approach based on variational multi-scale methods in space, proposed by Hughes and co-workers [3,4], was explored for multi-scale temporal integration by Bottasso [5]. A space–time multi-scale homogenization approach has also been presented by Fish and co-workers [6]. Other space–time formulations based on time-discontinuous Galerkin finite elements have been explored by Hughes and Hulbert [7,8] for elasto-dynamics. Haber and co-workers [9] further investigated adaptive time refinement to capture temporal multi-scale phenomena. For long-time integration, Bathe and co-workers [10,11] proposed an implicit composite time integration scheme that works well for conserving energy and momentum especially in problems with large deformations. Recently, Bathe and Noh [12] also formulated an explicit composite time-integration scheme and showed that the Bathe schemes (both implicit and explicit) have many desirable numerical properties such as enhanced stability, better accuracy, improved high-frequency dissipation and negligible dispersion in comparison to many existing time-integration schemes [13,14]. Important contributions to this area were also made by extending the variational time integrators to include asynchronous time integration by Lew et al. [15,16] and Matous and co-workers [17,18].

In this paper, an approach based on domain decomposition (DD) is adopted for separating different spatial & temporal scales. As the name suggests, DD methods usually divide the computational domain into several smaller subdomains, solve the subdomains separately and couple the solutions back together. The

* Corresponding author. Tel.: +1 765 494 6696; fax: +1 765 494 0395.
E-mail addresses: aprakas@purdue.edu (A. Prakash), etacir@ucla.edu (E. Taciroglu), Keith.Hjelmstad@asu.edu (K.D. Hjelmstad).

0045-7949/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compstruc.2013.11.013

text by Toselli and Widlund [19] and the survey article by Fragakis and Papadrakakis [20] provide an extensive summary of various DD methods in the literature. Some of the early works in this area are the mixed and multi-time integration methods by Belytschko and co-workers [21] and subcycling methods by Smolinski [22] and Daniel [23]. Further information on these types of methods is readily available in standard texts on finite element methods [24–26].

A dual-Schur domain decomposition method that uses Lagrange multipliers to enforce continuity of the solutions between subdomains was presented by Farhat and co-workers [27,28] as the 'finite element tearing and interconnecting' (FETI) method. The FETI method was extended for structural dynamics [29] by enforcing continuity of a chosen kinematic quantity (displacement, velocity or acceleration). A spectral stability analysis [30] showed that the the dynamic FETI algorithm is only *weakly* stable, and predicted a linear growth of instability. Farhat and co-workers also proposed a *time parallel* iterative method for dynamics [31,32]. Gravouil and Combescure [33,34] further extended the FETI method for dynamics to incorporate the use of multiple time-steps between the subdomains. They found that imposing continuity of time-discretized *velocities* at the interface led to a stable algorithm, and did not exhibit the theoretically predicted linear growth of instability. However, they also pointed out that the method had spurious numerical dissipation. Prakash and Hjelmstad [35] formulated a modified multi-time-step method using FETI for dynamics based on the time discretized equations of motion for linear problems. This method was shown to be stable and to *exactly preserve* the energy norms of individual subdomains. In terms of computational cost, this method was shown to be significantly more efficient, when compared to other coupling methods in the literature or to uniform time-step integration.

This paper presents a multi-time-step method for highly non-linear transient problems by first formulating the correct set of time-discrete equations of motion, and using a *consistent linearization* to extend the linear multi-time-step method of Prakash and Hjelmstad [35] to highly non-linear problems. It is shown that the proposed method is stable, accurate and computationally more efficient than using a uniform time-step for the entire mesh. Numerical examples are presented to illustrate the method and corroborate its performance in terms of the accuracy and efficiency of computation.

2. Solution of non-linear structural dynamics

First, we briefly describe conventional time integration approaches for problems in non-linear structural dynamics. The *semi-discrete* equations governing the non-linear dynamic behavior of a structural model can be written by defining the residual $\boldsymbol{r}$ as:

$$\boldsymbol{r}(\ddot{\boldsymbol{u}}(t),\dot{\boldsymbol{u}}(t),\boldsymbol{u}(t)) \equiv \boldsymbol{M}\ddot{\boldsymbol{u}}(t)+\boldsymbol{p}(\boldsymbol{u}(t),\dot{\boldsymbol{u}}(t))-\boldsymbol{f}(t) \tag{1}$$

where $\boldsymbol{M}$ represents the global mass matrix, $\boldsymbol{p}$ represents the internal force vector, $\boldsymbol{f}$ represents the external force vector, $\boldsymbol{u}$ denotes the vector of displacements and each super-imposed dot denotes one time-derivative. Note that the residual function $\boldsymbol{r}$ may include material and/or geometric non-linearities. The equation of motion can then be written as:

$$\boldsymbol{r}(\ddot{\boldsymbol{u}}(t),\dot{\boldsymbol{u}}(t),\boldsymbol{u}(t))=\boldsymbol{0} \tag{2}$$

This is a system of second-order, ordinary differential equations (ODEs) governing the non-linear dynamics of a structure. In addition to the governing Eq. (2), one also needs to specify the initial and boundary conditions. The initial state of the structure at time t_0 can be specified as:

$$\boldsymbol{u}(t_0)=\boldsymbol{u}_0; \qquad \dot{\boldsymbol{u}}(t_0)=\boldsymbol{v}_0 \tag{3}$$

where $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$ are the specified initial displacement and velocity vectors respectively. The Dirichlet and Neumann boundary conditions are specified as:

$$\boldsymbol{u}(t)=\bar{\boldsymbol{u}}(t) \quad \text{on } \Gamma_D \tag{4}$$

$$\boldsymbol{f}(t)=\bar{\boldsymbol{f}}(t) \quad \text{on } \Gamma_N \tag{5}$$

where $\bar{\boldsymbol{u}}$ denotes the specified displacement for the Dirichlet boundary and $\bar{\boldsymbol{f}}$ denotes the specified equivalent loads on the Neumann boundary.

The system of second-order ODEs (2) is usually solved numerically over a time interval of interest $I \equiv [t_0, t_N]$ where t_N is the final time of the simulation. The interval I is divided into N time-steps of size $\Delta t = t_n - t_{n-1}$ for $1 \leqslant n \leqslant N$ and the time derivatives are approximated using a finite difference (FD) scheme over these time-steps. The equation of motion is then solved at these discrete time-instants resulting in a system of *non-linear algebraic* equations.

The displacement, velocity, and acceleration vectors are approximated as $\boldsymbol{d}_n \approx \boldsymbol{u}(t_n)$, $\boldsymbol{v}_n \approx \dot{\boldsymbol{u}}(t_n)$ and $\boldsymbol{a}_n \approx \ddot{\boldsymbol{u}}(t_n)$ for a given instant of time t_n. A widely used time integration technique for this approximation is the *Newmark* family of schemes [36] given by:

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \Delta t[(1-\gamma)\boldsymbol{a}_n + \gamma \boldsymbol{a}_{n+1}] \tag{6}$$

$$\boldsymbol{d}_{n+1} = \boldsymbol{d}_n + \Delta t\,\boldsymbol{v}_n + \frac{1}{2}\Delta t^2[(1-2\beta)\boldsymbol{a}_n + 2\beta \boldsymbol{a}_{n+1}] \tag{7}$$

where γ and β are algorithmic parameters. Using these approximations, the equation of motion at an instant of time t_{n+1} can be written as:

$$\boldsymbol{r}(\boldsymbol{a}_{n+1},\boldsymbol{v}_{n+1},\boldsymbol{d}_{n+1}) = \boldsymbol{M}\boldsymbol{a}_{n+1}+\boldsymbol{p}(\boldsymbol{d}_{n+1},\boldsymbol{v}_{n+1})-\boldsymbol{f}_{\ n+1}=\boldsymbol{0} \tag{8}$$

Eqs. (6)–(8) form a system of *non-linear algebraic* equations, which can be solved using the Newton's method at every time-step in the interval I. Note also that at t_0 one must compute the initial acceleration $\boldsymbol{a}_0$ from the equilibrium equation at t_0:

$$\boldsymbol{M}\boldsymbol{a}_0 = \boldsymbol{f}(0)-\boldsymbol{p}(\boldsymbol{u}_0,\boldsymbol{v}_0) \tag{9}$$

To explain the iterative solution procedure, let the state of the system at any instant of time t_n be defined as $\boldsymbol{z}_n \equiv \{\boldsymbol{a}_n, \boldsymbol{v}_n, \boldsymbol{d}_n\}^T$. Thus, assuming that the state $\boldsymbol{z}_n$ is known, the task is to find $\boldsymbol{z}_{n+1}$ from Eqs. (6)–(8). In order to solve Eqs. (6)–(8) using the Newton's method, an initial guess (iteration $i=0$) for the state $\boldsymbol{z}^0_{n+1}$ at t_{n+1} is assumed. A commonly used guess is based on the previous converged state as:

$$\boldsymbol{a}^0_{n+1} = \boldsymbol{a}_n \tag{10}$$

$$\boldsymbol{v}^0_{n+1} = \boldsymbol{v}_n + \Delta t[(1-\gamma)\boldsymbol{a}_n + \gamma \boldsymbol{a}^0_{n+1}] \tag{11}$$

$$\boldsymbol{d}^0_{n+1} = \boldsymbol{d}_n + \Delta t\,\boldsymbol{v}_n + \frac{1}{2}\Delta t^2[(1-2\beta)\boldsymbol{a}_n + 2\beta \boldsymbol{a}^0_{n+1}] \tag{12}$$

where the superscript denotes the Newton iteration number, the subscript n refers to the converged values from the previous time step t_n. The solution at iteration i is updated by solving for $\Delta \boldsymbol{z}^i_{n+1} \equiv \{\Delta \boldsymbol{a}^i_{n+1}, \Delta \boldsymbol{v}^i_{n+1}, \Delta \boldsymbol{d}^i_{n+1}\}^T$ from the *linearized* form of the Eqs. (6)–(8) as follows:

$$\boldsymbol{M}\Delta \boldsymbol{a}^i_{n+1} + \boldsymbol{D}^i_{n+1}\Delta \boldsymbol{v}^i_{n+1} + \boldsymbol{K}^i_{n+1}\Delta \boldsymbol{d}^i_{n+1} = -\boldsymbol{r}(\boldsymbol{d}^i_{n+1}, \boldsymbol{v}^i_{n+1}) \tag{13}$$

$$\Delta \boldsymbol{v}^i_{n+1} = \Delta t \gamma \Delta \boldsymbol{a}^i_{n+1} \tag{14}$$

$$\Delta \boldsymbol{d}^i_{n+1} = \Delta t^2 \beta \Delta \boldsymbol{a}^i_{n+1} \tag{15}$$

where $\boldsymbol{K}^i_{n+1}$ and $\boldsymbol{D}^i_{n+1}$ represent the tangent stiffness and damping matrices. These matrices depend upon the assumed solution and need to be re-computed at every Newton iteration from the following relations: