Comput. Methods Appl. Mech. Engrg. 255 (2013) 158–166

Contents lists available at SciVerse ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

Asynchronous variational integration using continuous assumed gradient elements

Sebastian Wolff [a,*,1], Christian Bucher [b]

[a] Dynardo Austria GmbH, Wagenseilgasse 14, 1120 Wien, Austria
[b] Forschungsbereich für Baumechanik und Baudynamik, Technische Universität Wien, Karlsplatz 13/E2063, 1040 Wien, Austria

ARTICLE INFO

Article history:
Received 14 May 2012
Received in revised form 14 October 2012
Accepted 11 November 2012
Available online 7 December 2012

Keywords:
Asynchronous variational integrators
Continuous assumed gradient
Smoothed finite element method
Critical time step

ABSTRACT

Asynchronous variational integration (AVI) is a tool which improves the numerical efficiency of explicit time stepping schemes when applied to finite element meshes with local spatial refinement. This is achieved by associating an individual time step length to each spatial domain. Furthermore, long-term stability is ensured by its variational structure. This article presents AVI in the context of finite elements based on a weakened weak form (W2) Liu (2009) [1], exemplified by continuous assumed gradient elements Wolff and Bucher (2011) [2]. The article presents the main ideas of the modified AVI, gives implementation notes and a recipe for estimating the critical time step.

1. Introduction

Explicit time integration schemes may become inefficient in the presence of fast oscillators in the system. For example, finite elements of very small size and/or with stiff material properties reduce the critical time step of the whole system.

An approach to overcome this problem are mixed methods using explicit integrators for one domain and implicit methods for another mesh region [3,4]. Another line of development are multiple time stepping algorithms that integrate different parts of structures with different step sizes. This strategy is also known as subcycling [5]. The time step ratios are usually obtained by bisection, but also non-integer ratios can be used [6,7]. In the latter case, a clock is introduced which counts the smallest time step in the system. All nodes are thus updated which are behind the clock. Variants based on the symplectic Velocity Verlet method are presented in [8,7] together with stability criteria. The results are affirmed by Daniel [9] comparing different subcycling algorithms and proving the Smolinski and Sleith algorithm [8] as the only stable one. Belytschko's method [5] is shown to be only 'statistically stable' [9], improving stability with growing number of degrees of freedom.

A generalization of the multiple-time stepping scheme r-RESPA [10] are asynchronous variational integrators (AVIs) [11–14]. Therein, time step sizes are individually assigned to each finite element at arbitrary ratios. The method is variational. Hence it is symplectic and momentum preserving and, thus, well suited for long-term simulation. It implements a priority queue which decides on the sequence of drift phases and velocity kicks. The method can be extended to parallel implementations [15]. Convergence can be proved for linear elasticity [16]. Reliable stability criteria are difficult to find. A stability analysis was exemplified for a single degree of freedom system with two asynchronous potential functions in [17]. AVIs were successfully employed to improve accuracy and efficiency in domain decomposition [18,19] where each domain is assigned to individual time step sizes. They managed to develop implicit AVIs, but require synchronization times and lead to full-implicit couplings in between. The benefits of AVIs in efficiency were applied to contact/impact problems using a quadratic penalty formulation to compute contact forces [20] or using discontinuous momentum updates [21].

AVIs were originally formulated for isoparametric finite elements [11]. In this case, the critical time step size is approximated for each finite element by computing the elemental wave speed. This article formulates AVIs for a family of elements based on a weakened weak (W2) form, see [1] and the references therein. Such elements introduce either an assumed strain field (for infinitesimal strains) or an assumed deformation gradient field (for finite strains). The family includes nodal integration [22–24], smoothed finite element methods (SFEM) [25,26] and continuous assumed gradient (CAG) elements [2]. The mentioned methods were developed to improve the accuracy of low-order continuum elements. They conceptually differ from classical assumed strain methods, see for example [27], which enhance the interpolation functions of the strains within a finite element and solve the addi-

* Corresponding author. Fax: +43 (0)1 58801 20199.
E-mail addresses: sw@allmech.tuwien.ac.at, sebastian.wolff@dynardo.at (S. Wolff), christian.bucher@tuwien.ac.at (C. Bucher).
[1] Formerly at: Forschungsbereich für Baumechanik und Baudynamik, Technische Universität Wien, Karlsplatz 13/E2063, 1040 Wien, Austria.

0045-7825
http://dx.doi.org/10.1016/j.cma.2012.11.004

tional unknowns by static condensation. Among the W2 methods, the assumed strain (or gradient) field is an independent field which is solved only with information on the finite element geometry and the mesh topology involving some kind of smoothing operator applied to the natural strains (or gradients). The 'strain smoothing' is a very simple procedure which generally improves the accuracy of bending dominated problems and, in some formulations, the sensitivity to near incompressibility and mesh distortion. It is particularly efficient in explicit dynamics which can be performed with an increased critical time step due to softening the response by the smoothing operator.

1.1. Objectives and outline

Some stability properties of AVI were recently illustrated in [17] while its convergence was proven by [16] for linear elasticity. Some convergence properties of W2 forms were proven in [1]. Convergence and efficiency of CAG elements were extensively studied in [2,28] using a great variety of static and dynamic numerical experiments from linear and nonlinear elasticity involving infinitesimal and finite strains.

The objective of this article is to illustrate the implementation of explicit AVI when using CAG elements. Application of AVIs is not straight-forward. Due to the smoothing operator, no element stiffness matrix exists. Therefore, one needs to find new strategies for the spatial partition of asynchronous integration and for the estimation of the local time step size. The latter is solved by defining an equivalent "stiffness matrix" associated with a single smoothing cell and by redistributing the mass to the involved degrees of freedom.

The outline is as follows: Section 2 introduces the basics of CAG elements. For more details, the reader is referred to [2]. Section 3 and 4 presents a generalized formulation of asynchronous variational scheme following the ideas in [17] and then applies the formulation to structural elasticity using CAG elements. The strategy for estimating the time step length is illustrated in Section 5. Examples compare the efficiency of AVI with central differences in Section 6.

2. Continuous assumed gradient elements

2.1. Assumed gradient field

The deformation gradient of isoparametric elements is

$$F^h_{\alpha\beta}(\xi) = \nabla_\gamma N_i(\xi) J^{-1}_{\gamma\beta}(\xi) u_{i\alpha} + \delta_{\alpha\beta} \tag{1}$$

with shape function N_i and the inverse of the Jacobian $\mathbf{J}^{-1}$. Due to C^0-continuity of the displacements $\mathbf{F}^h$ is continuous in the interior of each element and discontinuous at finite element boundaries. This discontinuity is the reason that the error (in energy norm) in the interior of low-order elements is often smaller than at the finite element boundaries.

The idea behind CAG is now to replace $\mathbf{F}^h$ by a continuous field with interpolation function M_i

$$F^{AN}_{\alpha\beta}(\xi) = M_A(\xi) F_{A\alpha\beta} \tag{2}$$

where $\mathbf{F}_A$ are the values at the supporting points A and $\mathbf{F}^{AN}(\xi)$ is the assumed natural deformation gradient. In case of nodal integration, M_A are identical with the finite element shape functions. In case of SFEM, the interpolation functions are piecewise constant.

The values $\mathbf{F}_A$ are additional degrees of freedom. They are chosen in such a way that the resulting field $F^{AN}_{\alpha\beta}(\xi)$ is a good approximation to the field computed from the finite element shape derivatives $F^h_{\alpha\beta}(\xi)$. The identity of both fields is enforced in a weak sense using the principle of Veubeke-Hu-Washizu incorporating a continuous field of Lagrange multipliers $\boldsymbol{\lambda}(\xi)$, i.e.

$$\lambda_{\alpha\beta}(\xi) = L_A(\xi) \lambda_{A\alpha\beta} \tag{3}$$

with interpolation function $L_A(\xi)$. The multipliers are incorporated by adding the term Π^{VHW} to the strain energy,

$$\Pi^{VHW} = \int_V \lambda_{\alpha\beta}(\xi) \left(F^{AN}_{\alpha\beta}(\xi) - F^h_{\alpha\beta}(\xi) \right) dV \tag{4}$$

introducing $\lambda_{A\alpha\beta}$ as additional degrees of freedom. By using dual multiplier spaces [29] for the interpolation of λ one obtains a lumped matrix structure and, thus, the discrete multipliers can be explicitly eliminated. The multiplier space is constructed by the linear combination

$$L_A(\xi) = a_{AB} M_B(\xi) \tag{5}$$

where a_{AB} denotes some coefficient matrix which is chosen to satisfy the biorthogonality condition

$$\int_V L_A(\xi) M_B(\xi) dV = \delta_{AB} \int_V M_A(\xi) dV \tag{6}$$

Finally, after variation of the enhanced energy function with virtual degrees of freedom $\delta\mathbf{u}_A$, $\delta\boldsymbol{\lambda}_A$ and $\delta\mathbf{F}_A$, one obtains for the discrete deformation gradient

$$F_{A\alpha\beta} = \frac{\int_V L_A(\xi) F^h_{\alpha\beta}(\xi) dV}{\int_V M_A(\xi) dV} = \frac{\int_V L_A(\xi) \nabla_\gamma N_B(\xi) J^{-1}_{\gamma\beta}(\xi) dV}{\int_V M_A(\xi) dV} u_{B\alpha} + \delta_{\alpha\beta} \tag{7}$$

while the Lagrange multipliers λ_A are identified as the 1st Piola–Kirchhoff stress tensor. The integrals are numerically evaluated in a preprocessing step. The dual coefficients a_{AB} are geometry dependent; the averaging operator (or 'smoothing operator') in Eq. (7) depends on the mesh topology.

2.2. Strain energy

Using the explicit representation (7), the strain energy becomes

$$U = \int_V U^d(\mathbf{F}^{AN}, t) dV = \sum_A W_A U^d_A(\mathbf{F}_A, t) \tag{8}$$

with strain energy density function U^d. W_A is an integration weight denoting the fictive volume around integration point A, i.e.

$$W_A = \int_V M_A(\xi) dV \tag{9}$$

Therein, the numerical integration points are chosen to be identical to the support points A of the gradient interpolation.

Relation (7) defines an explicit smoothing operator which averages the discontinuous deformation gradients at finite element boundaries to obtain a single quantity. It does not require static condensation or other knowledge on constitutive relations, but is entirely dependent on the geometry of the adjacent elements and on the finite element mesh topology.

The choice of the interpolation functions and support point coordinates for the gradient field is crucial to ensure stability and accuracy of the formulation. For example, nodal integration and NS-FEM are unstable involving the appearance of spurious low-energy modes. They need non-physical penalty energy functions that stabilize them. The articles [2,28] numerically verify the stability, convergence and accuracy of several W2 variants including new elements which can be constructed based on the idea of assumed continuous deformation gradients. For first order hexahedral elements, [2,28] found good results for the element types C3D_8N_27C and C3D_8N_8I. The first is defined by 27 support points and a second order tensor-product interpolation of the deformation gradient by Lagrange polynomials. The latter element type is defined by 16 support points with 8 points being coincident with the nodes and 8 additional points in the element interior. Among the tested first order tetrahedra, the nodally integrated

tetrahedron with an additional bubble mode in the gradients was found to be most accurate. It turned out to be even the most efficient with respect to computing time in explicit analysis [28] because the enlarged critical time step compensates the slightly increased numerical cost per restoring force assembly. Fig. 1 illustrates the positions of support points for various CAG and SFEM formulations.

2.3. Linearization

The restoring force vector $\mathbf{F}$ and the stiffness matrix $\mathbf{K}$ are well defined and given through

$$F_{\mathrm{A}} = \sum_{B}\sum_{m\in B} W_B^m \sigma_{B\alpha}^m \frac{\partial \epsilon_\alpha^m(\mathbf{F}_B)}{\partial F_{B\beta\gamma}} \frac{\partial F_{B\beta\gamma}}{\partial u_{\mathrm{A}}} \tag{10}$$

$$K_{\mathrm{AB}} = \sum_{C}\sum_{m\in C} W_C^m \sigma_{C\alpha}^m \frac{\partial^2 \epsilon_\alpha^m(\mathbf{F}_C)}{\partial F_{C\beta\gamma}\partial F_{C\eta\omega}} \frac{\partial F_{C\beta\gamma}}{\partial u_{\mathrm{A}}} \frac{\partial F_{C\eta\omega}}{\partial u_{\mathrm{B}}} + \sum_{C}\sum_{m\in C} W_C^m C_{C\alpha\delta}^m \frac{\partial \epsilon_\alpha^m(\mathbf{F}_C)}{\partial F_{C\beta\gamma}} \frac{\partial \epsilon_\delta^m(\mathbf{F}_C)}{\partial F_{C\eta\omega}} \frac{\partial F_{C\beta\gamma}}{\partial u_{\mathrm{A}}} \frac{\partial F_{C\eta\omega}}{\partial u_{\mathrm{B}}} \tag{11}$$

wherein

$$\sigma_{A\alpha}^m = \frac{\partial}{\partial \epsilon_\alpha^m} U_A^m(\boldsymbol{\epsilon}^m(\mathbf{F}_A), \boldsymbol{\alpha}_A^m, t) \tag{12}$$

$$C_{A\alpha\beta}^m = \frac{\partial^2}{\partial \epsilon_\alpha^m \partial \epsilon_\beta^m} U_A^m(\boldsymbol{\epsilon}^m(\mathbf{F}_A), \boldsymbol{\alpha}_A^m, t) \tag{13}$$

denote the 2nd Piola–Kirchhoff stress and tangential material tensors. ϵ is the strain tensor and $m \in A$ denotes the index within the set of materials being topological neighbors of the Ath integration point.

To prepare the outline in Section 5 consider the structure of $\mathbf{K}$. In classical finite element formulations, even when using enhanced strains or incompatible modes, the stiffness matrix can be assembled by the union of element stiffness matrices. Each element stiffness matrix $\mathbf{K}^n$ contributes stiffness values to all degrees of freedom which belong to the nodes of the respective element n. The same is true for the element mass matrix $\mathbf{M}^n$. Since both are defined in the same conforming space of nodal displacements, the generalized element eigenvalues can be computed straight-forward. This approach is widely used to approximate the largest eigenvalue (and, thus, critical time step) of the entire system by the maximum element eigenvalue among all elements.

An element stiffness matrix in this sense does not exist for the weakened weak form. An element stiffness matrix may be obtained by integrating the strain energy density over the spatial domain of the nth element and calculating its Hessian. But this matrix then influences not only the degrees of freedom of the nth element, but also the degrees of freedom of all adjacent elements. A quantity being used in this article, however, is the Hessian $\mathbf{K}_A = W_A \nabla_{\mathbf{u}}^2 U_A^d$ associated with the Ath 'smoothing cell' [25] (or, repsectively, Ath support point of the assumed gradient interpolation). It generally contains terms regarding to all degrees of freedom influenced by Eq. (7). For inner-elemental support points, these are all element degrees of freedom. For nodal support points, these are the degrees of freedom of all surrounding elements.

3. Asynchronous variational integration

Assume that the total potential energy is obtained by some additive composition

$$V(q) = \sum_i V_i(q) \tag{14}$$

with generalized coordinates q.

In case of a synchronous time stepping scheme, the composition $V(q)$ is evaluated at discrete times, i.e. the potentials $V_i(q)$ and their derivatives are computed at synchronous times. The idea of asynchronous integration lies in evaluating the individual potentials at separate times. Asynchronous integration can, therefore, be interpreted as a generalization of multi time stepping. While multiple time stepping schemes create points in time which are synchronous, AVIs can be configured such that synchronous points exist; but their existence is not required.

Assign to each potential V_i a sequence of times $\{0 = t_i^0 < \cdots < t_i^{M_i} = T_i\}$. Another sequence is created by inserting all times t_i^j into a unique and sorted set which then contains all system times $\{\theta^0 < \theta^1 < \cdots < \theta^M\}$, see Fig. 2. The solution trajectory is obtained by a piecewise linear interpolation of the generalized coordinates along their supports at the system times θ_k, i.e. each θ_k is associated to a discrete coordinate q_k.

Define the function

$$\mathcal{A}(k,i) = j, \quad \max_j t_i^j \leqslant \theta_k \tag{15}$$

which determines the index j of time t_i^j where the potential V_i has been evaluated most recently prior system time θ_k. The function

$$\mathcal{K}(i,j) = k, \quad t_i^j = \theta_k \tag{16}$$

returns the index k on the total time scale θ for a given index pair (i,j) defining the potential V_i and the potential time index j. Then the discrete action for a single element writes

$$S_k = j_k^T q_k^+ - j_{k+1}^T q_{k+1}^- + \frac{1}{2(\theta_{k+1}-\theta_k)} \left\| q_{k+1}^- - q_k^+ \right\|_M^2 - (\theta_{k+1}-\theta_k)\left(\sum_i V_i\left(q_{\mathcal{K}(i,\mathcal{A}(k,i))}^+ \right) \right) \tag{17}$$

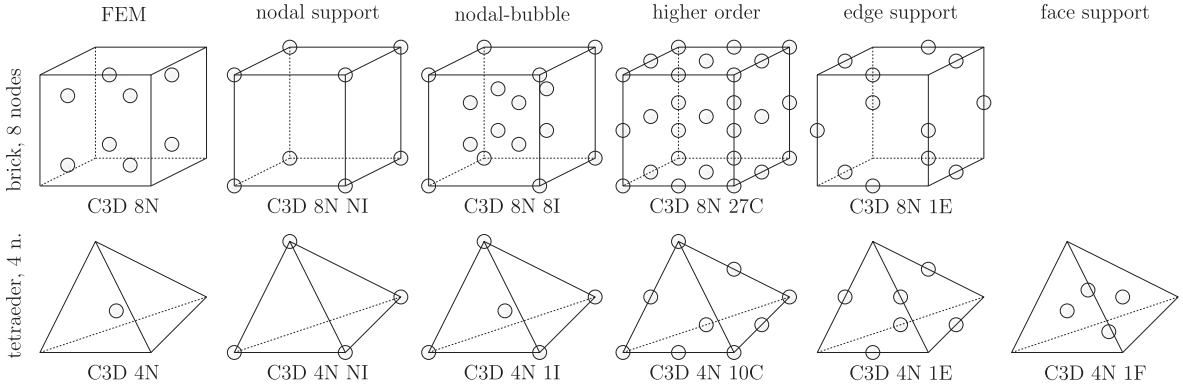

Fig. 1. Illustration of various 1st order CAG elements [2]. The circles define the positions of the support points of the assumed gradient field.

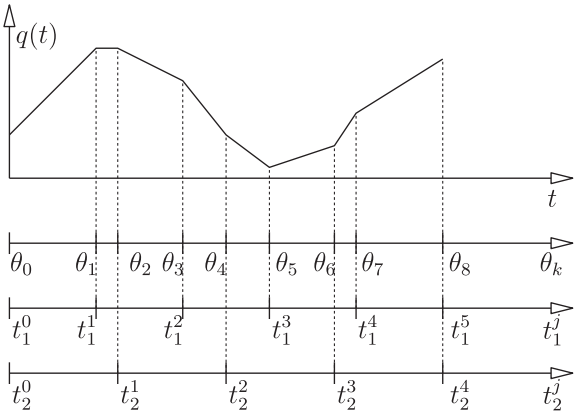

Fig. 2. Illustration of fixed-step size AVI of a SDOF system with 2 potentials V_i being evaluated either at t_1^j or t_2^j. $q(t)$ is interpolated between discrete times θ_k.

wherein the discrete momenta j_k are Lagrange multipliers which enforce continuity of $q(t)$ at the time step boundaries. Application of the stationarity principle leads to the time stepping scheme

$$j_{k+1} = j_k - \sum_{i\in\mathcal{I}(k)} \left(t_i^{A(k,i)+1} - t_i^{A(k,i)}\right) \nabla V_i(q_k^+) \tag{18}$$

$$q_{k+1}^+ = q_k^+ + (\theta_{k+1} - \theta_k) M^{-1} j_{k+1} \tag{19}$$

In words, at time θ_k one determines all potentials which are part of the set $\mathcal{I}(k)$, i.e. which are active at this time. The modification of the momentum is identical to symplectic synchronous Euler except that the time step sizes, which scale the contributions of all active potentials V_i, are not identical to the size of the time element $(\theta_{k+1} - \theta_k)$, but are the time steps of the potentials $\left(t_i^{A(k,i)+1} - t_i^{A(k,i)}\right)$. The trajectory of the generalized coordinates within the time element is characterized by a constant motion using the modified momentum j_{k+1}.

4. Application to structural elasticity

Let us turn to Hamilton's principle of structural mechanics. The action is given through the space–time integral

$$S = \int_0^T \int_V \left(\frac{1}{2}\rho(\xi)[\dot{x}(\xi,t)]^2 - U^d(\mathbf{F}^h(\xi,q),t)\right) dV\, dt \tag{20}$$

with material coordinate ξ, deformed coordinate x, mass density ρ, strain energy density U^d and volume in reference configuration V. In synchronous time stepping schemes, the spatial integral is discretized and evaluated first. Subsequently, the temporal integral is discretized. In asynchronous schemes, both integrals are discretized and solved simultaneously. For CAG elements, the strain density function in the above equation is replaced by $U^d(\mathbf{F}^{AN}(\xi,q),t)$ while the mixed terms enforcing the weak equivalence of $\mathbf{F}^{AN}$ and $\mathbf{F}^h$ do not contribute to the action and can be neglected.

For the kinetic action, one choses

$$T_d = \int_{t_a}^{t_b} \frac{1}{2} \dot{q}^h(t)^T M \dot{q}^h(t) dt \tag{21}$$

Therein, $q^h(t)$ is a piecewise linear interpolation of the generalized coordinates q with support at times θ_k. The mass matrix is given by

$$M_{AB} = \int_V \rho(\xi) N_A(\xi) N_B(\xi) dV \tag{22}$$

with finite element shape function $N_A(\xi)$. AVIs can only be efficient if a diagonal (lumped) mass matrix is assumed, for example through

$$M_{AB}^{lumped} = \delta_A^B \int_V \rho(\xi) N_A(\xi) dV \tag{23}$$

where δ_A^B denotes the Kronecker delta.

The potential energy can be expressed by a linear combination of discrete values of the strain energy density U_A^d and some integration weights W_A at spatial integration point A

$$V_d = \int_{t_a}^{t_b} V(q(t)) dt = \int_{t_a}^{t_b} \left(\sum_A W_A U_A^d(\mathbf{F}^{AN}(q(t)),t)\right) dt \tag{24}$$

Utilizing the symplectic Euler scheme one obtains

$$V_d = \sum_i \sum_{j=0}^{n_i-1} h_i^j V_i\left(q\left(t_i^j\right), t\right) \tag{25}$$

$$V_i(q,t) = W_i U_i^d(\mathbf{F}^{AN}(q),t) \tag{26}$$

$$h_i^j = t_i^{j+1} - t_i^j \tag{27}$$

Therein, h_i denotes the time step of the ith potential V_i. Notice, that symplectic Euler is second order in the displacements, but only first order in the velocities. A complete second order scheme is given through Velocity Verlet which computes the momenta as $j_{k+1}^{VV} = \left(j_{k+1}^{Euler} + j_k^{Euler}\right)/2$ wherein j_k^{Euler} denotes the momenta in Eq. (18).

The discretization is shown in Fig. 3. It also illustrates that the equation systems to be solved are very small compared with the synchronous case: Every spatial integration point is only dependent on the deformed coordinates of the adjacent nodes. Therefore, a force influences only the momentum of these nodes. All other nodes perform a constant motion during this time. It is, therefore, not necessary to update the deformed coordinates of all nodes at all system times θ_k, but only those nodes which are part of the influencing domain of the currently active potential V_i.

In the original formulation [11], each finite element is associated with an individual time step, i.e. the spatial integration points of each finite element are collected to an individual V_i. This strategy simplifies a few things: (1) It is easier to implement AVIs in existing finite element codes since the finite elements are independent from the temporal discretization. (2) An estimation of the critical time step is relatively easy when elemental wave speeds are computed.

When using interpolations of the deformation gradient being continuous at finite element interfaces then a finite element based subdivision may not be the best solution. Using classical isoparametric finite elements, the forces of the ith element influence the nodes of the same element only. Using continuous interpolations, an integration point located on an element interface may influence the nodes of all surrounding elements. In particular nodal integra-

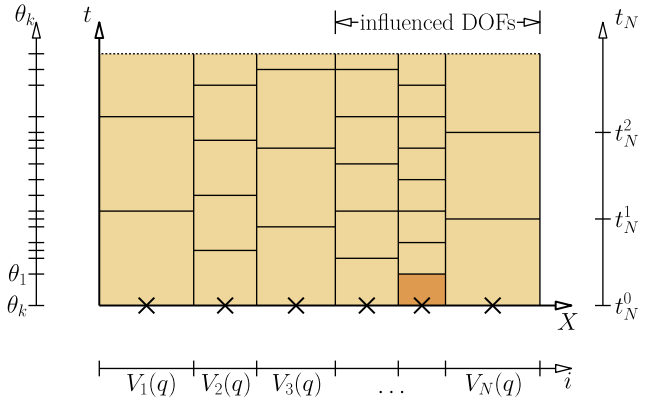

Fig. 3. Illustration of space–time integration in one spatial dimension. The colored area is the space–time domain to be integrated (material space X × time t). Each cross denotes a single spatial integration point representing a single potential V_i. The highlighted cell is a space–time cell associated with a single integration point in space and time.

tion points may belong to many finite elements. If the integration points of a single element are collected to build some $V_i(q,t)$ then one needs to update the nodes of all surrounding elements. Furthermore, since the same integration point is part of two or more elements, it may be evaluated more often than necessary, see Fig. 4. Therefore, it is proposed that every support point A of the continuous gradient interpolation describes an individual potential $V_i(q)$. By doing so, the number of material law evaluations is minimized during the time integration. On the other hand, the number of coordinate updates is larger compared with [11].

Variation of the space–time integral, Eq. (20), leads to the scheme given by Eqs. (18) and (19). Special care must be spent when updating the minimum required set of coordinates in Eq. (19). Every node A remembers the time θ^A at which its coordinates and momentum has been computed (and stored) the last time. Then the next update of the same node refers to the state at time θ_{k+1}, see Fig. 5 for more details. Of course, all nodes may be updated at all system times. But this greatly reduces numerical efficiency.

The resulting scheme is summarized in Algorithm 1. A main component is the priority queue. It decides which potential is the next one to be evaluated. Each potential appears once in the queue which is sorted according to the next evaluation times. Since multiple potentials may have identical times, a secondary sort condition is required to make the ordering unique. Further notice, the set of nodes which influence the currently active potential V_i in Algorithm 1 is identical to the set of nodes which obtains a momentum update.

Algorithm 1. AVI with constant step sizes

Set initial conditions q_0, j_0.
Create global counter $k := 0$
For each potential V_i, compute a time step h_i^0 and a first kick time $t_i := h_i^0$.
Create a priority queue of indices i which is sorted for t_i in ascending order.
Set global time $\theta_k := 0$
For each node A, set a global time $\theta^A := \theta_k$.
while $\theta_k < T$ **do**
  Take 1st element from priority queue and remember indices i and time t_i.
  $\theta_{k+1} := t_i$
  for all nodes A which influence V_i **do**
    Perform drift:
    $q_{k+1}^A := q_k^A + h_k j_k^A / M_A, \quad h_k := \theta_{k+1} - \theta^A$
    Set current time of node A : $\theta^A := \theta_{k+1}$
    Perform kick:
    $j_{k+1}^A = j_k^A - h_i^0 \nabla^A V_i(q_{k+1})$
  end for
  Update state:
  Set next evaluation time $t_i := t_i + h_i^0$
  $k := k + 1$
  Reinsert i (and t_i^i) into priority queue.
end while
Drift all nodes to the final time:
for all nodes A **do**
  $q_{k+1}^A := q_k^A + h_k j_k^A / M_A, \quad h_k := T - \theta^A$
end for

5. Estimating the critical time step

A strict proof of stability criteria for AVI applied to multiple degree of freedom systems is not available at the moment. For traditional sub-cycling methods it is possible to give stability criteria

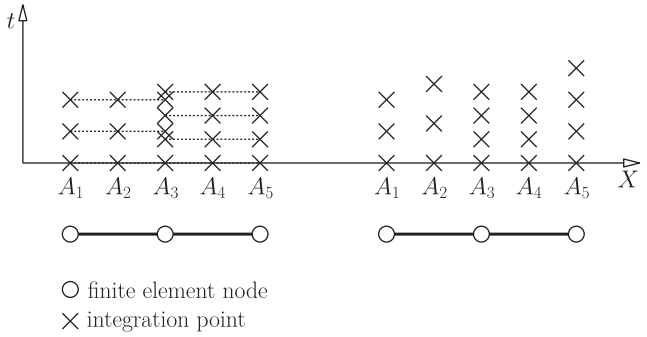

Fig. 4. Element partition vs. integration point partition. Two linear finite elements in 1d. Left: each V_i is one element; Right: each integration point A_i is one V_i. Standard finite elements have their integration points only in the interior, i.e. A_2 and A_4. CAG elements use integration points which are part of multiple elements, i.e. A_3. The number of shared elements can be numerous in 3d. These integration points are unnecessarily often evaluated if an element partition is used.

(e.g. [7]) because synchronous configurations in time exist. Then one can construct the transition matrix which maps one synchronous phase space configuration to the other. This is very different in AVI where generally no synchronous configurations appear after the initial time.

A deep discussion of stability limits of AVI was presented by Fong et al. [17] when considering a single degree of freedom

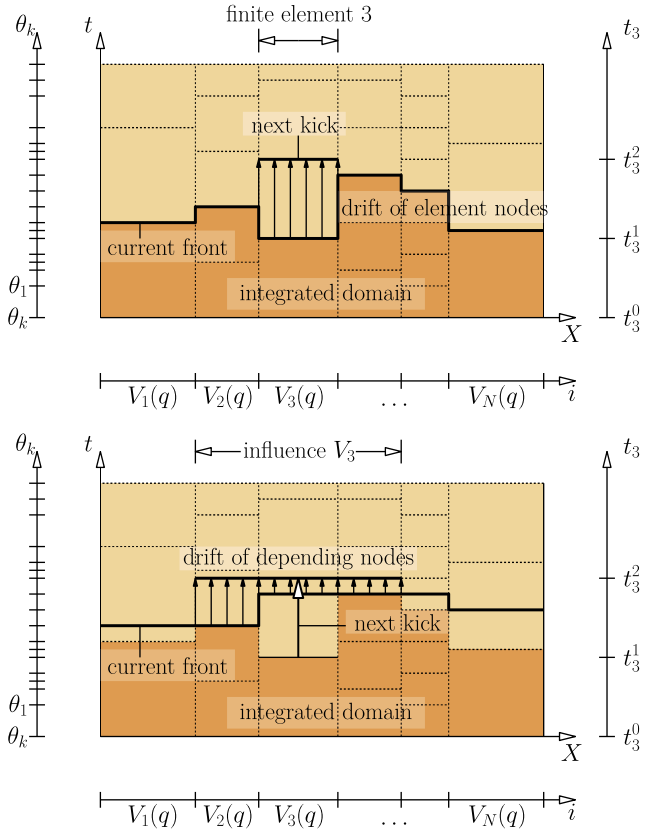

Fig. 5. Illustration of a space–time front in one spatial dimension [21]: The colored area is the space–time domain to be integrated (material space X × time t). The highlighted area is the domain which is already integrated. The deformed coordinates of all nodes are known at times θ^A defining the current front. For standard finite elements (top) the front is identical to the boundary of the already integrated domain. This is different for CAG elements (bottom): The most recently integrated space–time cells require the drift of all influencing nodes (even of adjacent finite elements). Potential V_3 is "kicked" next at time t_3^2 requiring the drift of surrounding nodes.

system with two potentials. The instabilities have the same nature as resonances appearing in r-RESPA [10,30].

An intuitive explanation for resonances is given in Fig. 6 using two potentials. Assume that the time step of the first potential Δt is exactly the half period of the second potential. Between two kicks of $V_1 = V^{slow}$ the vibration phase yields a trajectory with the natural frequency of $V_2 = V^{fast}$. Now consider a linear oscillator (representing the 'vibration' of V_2) with initial conditions $q = 1$ and $j = 0$. The kick of V_1 is then always applied when $q = 1,\ j \leqslant 0$ or $q = -1,\ j \geqslant 0$. In both cases, the sign of the force is such that the velocity will always increase leading to a monotonous increase of energy.

One of the unpleasant results of the analysis in [17] is that unstable combinations of arbitrarily small time steps may appear even though many of them are very mild which are hard to notice are which only play a role in long-term simulation. Furthermore, nearby any stable time step combination there is an arbitrarily close combination of times steps that could be unstable. Some of these instabilities may arise only for the single degree of freedom system under consideration, but may explain unexpected instabilities in finite element models. The authors notice, however, that for most finite element meshes these instabilities are less frequent or not dangerous.

A strategy to estimate the local time step in finite element meshes which usually leads to a stable solution relies on the CFL condition [31,32] being illustrated in Fig. 7. Using the solution of the system of partial differential equations in time and space, the state at point P' depends on the state of all points within the domain of dependence. This domain is a cone which slope is defined by the wave speed. Stability is then ensured if all grid points within the cone of dependence are actually used when computing the state in P' by the numerical method. The time step may be reduced to ensure stability in Fig. 7 right. The scheme using central differences is stable if the CFL condition

$$c\frac{\Delta t}{\Delta X} \leqslant C \tag{28}$$

is satisfied where C denotes the (dimension dependent) Courant number, ΔX and Δt the grid spacing, and c the velocity of wave propagation.

Given isoparametric finite elements the wave speed can be computed using elemental eigenvalues [11]. This strategy usually gives a stable time step limit. It should be noted, however, that this estimation is used without a rationale except an illustrative interpretation of the CFL condition, albeit it can be backed by numerical examples.

For general material laws and general finite element types a prediction of the wave speed and of the suitable grid size

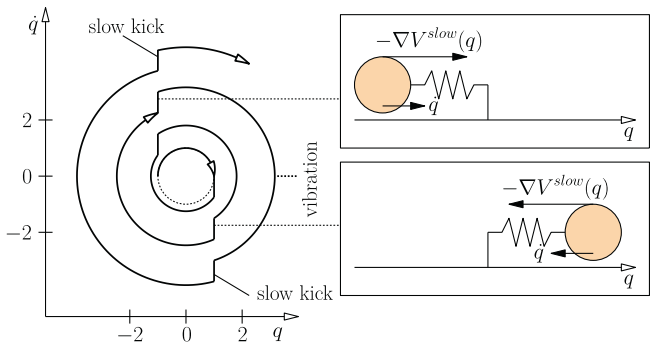

Fig. 6. Phase space diagram of a harmonic oscillator illustrating r-RESPA resonances [17]. The oscillator is hit by a velocity change every half period. Instead of forming a closed orbit, the kicks lead to monotonous energy growth. Assume that the oscillator is represented by V^{fast} with a very small time step.

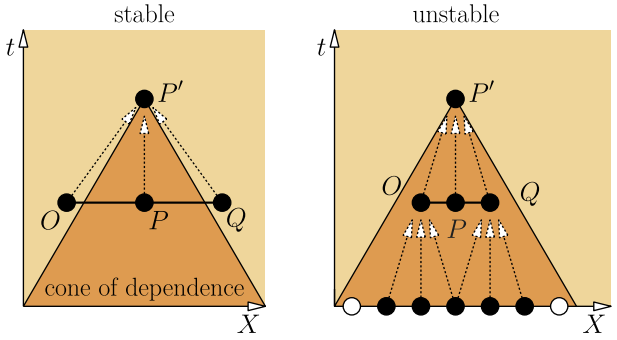

Fig. 7. Domain of dependence and stability for a rectangular space–time grid in one dimension.

parameter leads to numerous case studies. Thus, a general recipe is required. We propose a local eigenvalue problem for each interpolation point of the continuous assumed gradient field, i.e. at each spatial integration point.

Solve the local eigenvalue problem

$$\nabla^2 V_A(q_0)\mathbf{v} = \lambda m_A \mathbf{v} \tag{29}$$

at integration point A, wherein $\nabla^2 V_A$ denotes the local initial Hessian, $\mathbf{v}$ the eigenvector, λ the eigenvalue and m_A the mass of the integration point

$$m_A = \rho_A W_A \tag{30}$$

The Hessian $\nabla^2 V_A$ can be interpreted as a local stiffness matrix of the Ath support point.

The largest eigenvalue can be easily obtained by the power iteration. Notice, the Hessian matrix $\nabla^2 V_A$ is very sparse. It is usually stored in a column-wise layout. During the iteration, the matrix–vector product $\mathbf{y} = \nabla^2 V_A \mathbf{x}$ can be computed efficiently by assuming symmetry, i.e. one multiplies all nonzero components of the ith column with the jth element of $\mathbf{x}$ to obtain the ith element in $\mathbf{y}$. The vectors can be stored in dense format. Although many degrees of freedom may be allocated, the vector norm of $\mathbf{y}$ is computed efficiently since all non-zero components of $\mathbf{y}$ can be determined as the non-empty column indices of the local Hessian matrix.

The critical time step is then estimated from

$$\Delta t_A \leqslant \frac{2}{\sqrt{\max \lambda_A}} \tag{31}$$

This condition is identical to the stability condition of Velocity Verlet.

6. Example: asynchronous integration of a cantilever beam

6.1. Model problem

The following example presents the performance of AVI compared with Velocity Verlet. Furthermore, it illustrates the behavior of the distribution of the time step sizes within the structural model. The mechanical system is conservative, but nonlinear.

Consider a cantilever beam with square cross section as illustrated in Fig. 8. The geometry is defined by $L = 100,\ B = H = 10$. The material is a given by the nonlinear elastic St. Venant model with elastic modulus $E = 30 \times 10^3$, Poisson's ratio $\nu = 0.3$ and mass density 2400×10^{-9}.

The beam is discretized by $10n \times n \times n$ 8-noded brick elements where n is a mesh parameter. As finite element formulation the continuous assumed gradient element C3D_8N_27C is used. The element sizes are chosen to vary along the x-axis. The element

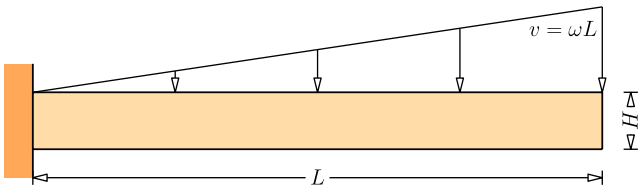

Fig. 8. Geometry of a cantilever beam.

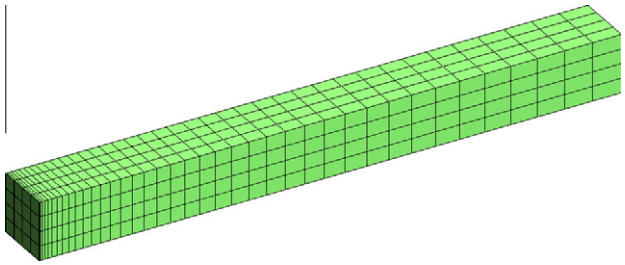

Fig. 9. Mesh of a cantilever beam.

sizes are small at the support and large at the beam's free side. The x-coordinates of the finite element nodes are given by $X_i = L(\frac{i}{10n})^2$ where i is a node index $i \in [0, 10n]$. The mesh is illustrated in Fig. 9.

The initial conditions are given by zero displacements $q_0 = 0$ and vertical velocity $v^y_{A,0} = -\omega X_A$ with node A, horizontal coordinate X_A and parameter $\omega = 180$. The time step is chosen to be $\Delta t = 0.5\Delta t_{crit}$. The simulation time interval is $T = 0.005$.

6.2. Benchmark against synchronous time stepping

Velocity Verlet is used as reference solution. Therein, only one restoring force evaluation is implemented (the end step force vector is temporarily stored and used as start step force in the subsequent time step). By doing so, the numerical efficiency is comparable with the asynchronous method with only one force evaluation per step.

Three methods are compared:

1. standard Velocity Verlet,
2. asynchronous scheme with constant step sizes.
3. Furthermore, the asynchronous algorithm is used to emulate a synchronous scheme by setting all time steps to the smallest time step found in the system. Therefore, the algorithm becomes synchronous. By comparing this strategy with standard Verlet one can measure the computational overhead introduced by the asynchronous procedure (maintaining the queue, numerous drifts, etc.).

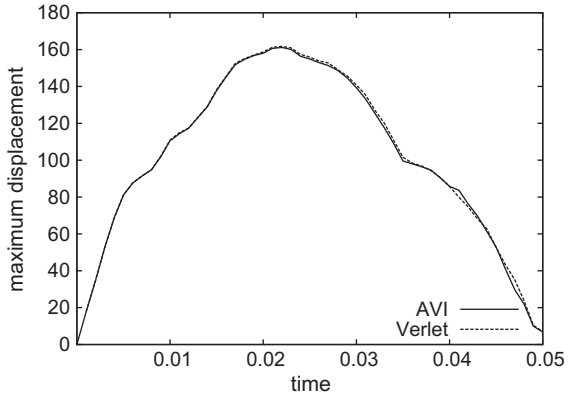

Fig. 11. Maximum displacement over time for Verlet and AVI.

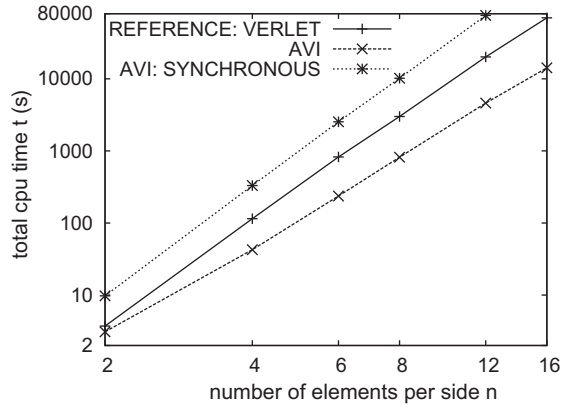

Fig. 12. Comparison of total cpu time using standard Verlet, AVI and AVI with synchronous time step for different mesh size parameters.

The energy balance for Velocity Verlet and the asynchronous procedure for the mesh size parameter $n = 4$ is given in Fig. 10. The maximum displacement over time for both methods is shown in Fig. 11. Both solutions are nearly identical.

The total cpu times for the three methods are illustrated in Fig. 12. The asynchronous scheme was faster than standard Verlet by a factor of 4,...,6 with increasing magnitude for larger meshes.

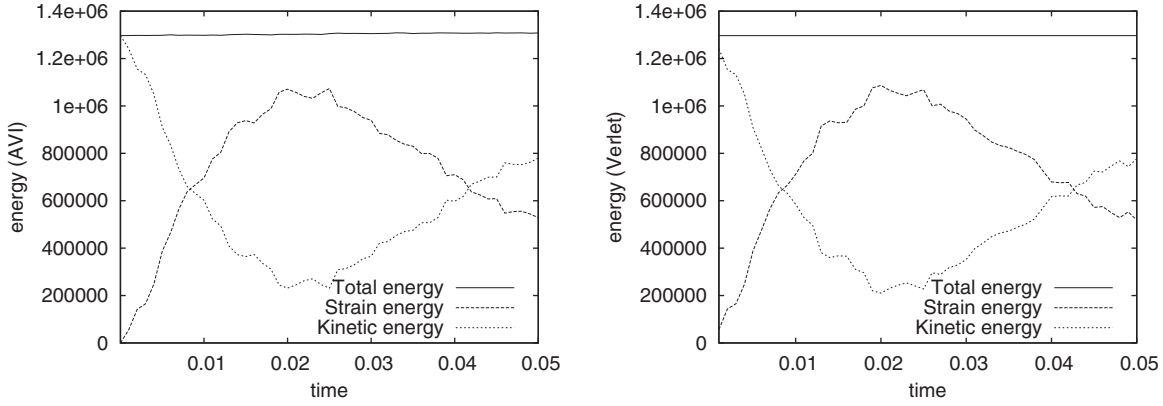

Fig. 10. Energy balance of AVI (left) and Verlet (right).

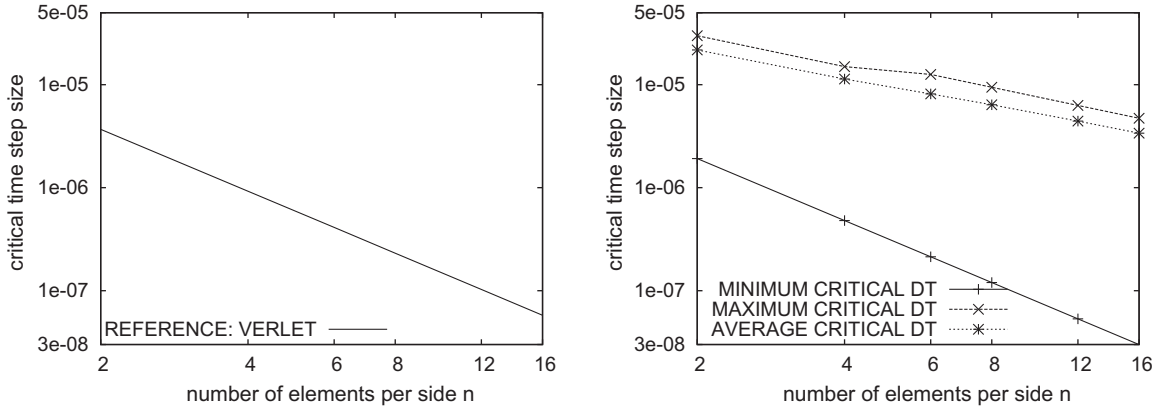

Fig. 13. Critical time steps for Verlet and AVI.

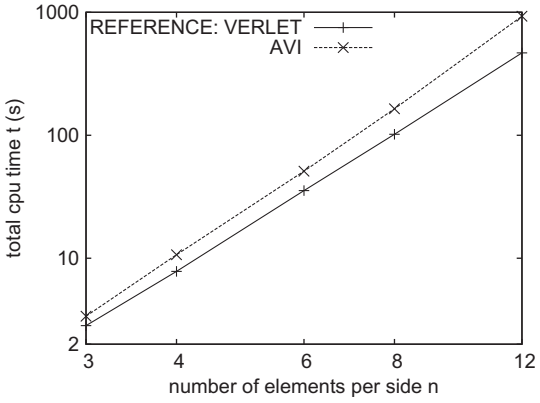

Fig. 14. Comparison of total cpu time using standard Verlet and AVI for different mesh size parameters with equidistant nodes (worst case).

The overhead of the asynchronous procedure is approximately given by a factor of eight compared with the standard scheme. This factor can be computed by relating the simulation times of both schemes to the different time step sizes being actually used.

The critical time steps are given in Fig. 13. The critical time step of standard Verlet is shown in the left subfigure. The time step of the synchronous setting of the asynchronous algorithm is computed from the minimum critical time step in the right subfigure.

Interestingly, the local estimation leads to a smallest time step which is approximately half as large as the time step used in the reference solution. Increasing the safety factor such that the smallest time step equals the one used in Verlet, however, may lead to an unstable solution.

6.3. Equally distant nodes

By changing the mesh generation scheme one may create a worst case scenario for the efficiency of asynchronous integrators. Now, each finite element is a cube with uniform edge length $10/n$. The computing times of standard Verlet and the asynchronous case are presented in Fig. 14 for different mesh size parameters. Obviously, the standard method outperforms the asynchronous scheme. This is because there is almost no benefit due to a broad deviation of the critical time step size. For example, given a mesh size parameter $n = 8$, the standard integrator obtains a critical time step of $h_{crit} = 7.016 \times 10^{-6}$. The asynchronous scheme obtains the interval $h_{crit} \in [6.406 \times 10^{-6}, 9.346 \times 10^{-6}]$ with an average of $h_{crit}^{avg} = 8.357 \times 10^{-6}$. Clearly, there is almost no deviation of the critical time step within the spatial domain. Its average value is very close to the one of the standard method. Considering the additional computational effort of the asynchronous algorithm, this results in a less efficient scheme.

This example shows that, although there may be great benefits in numerical efficiency when applying AVIs to certain models, the contrary may happen when applied to other models.

If isoparametric elements are used and if each element represents an individual potential V_i then Velocity Verlet and AVI would produce the same solution because both would use the same time step being applied to all elements. This is different in the present case, because even in the presence of homogenous element geometries slightly different eigenvalues are computed for the individual integration points.

7. Conclusions

Asynchronous variational integrators were presented in the context of CAG elements. The space–time discretization includes individual time step sizes per support point (or "smoothing cell") while the original AVIs associate individual time step sizes to each finite element.

Of particular interest was the estimation of the local time steps. When applied to CAG elements, it is not easy to determine the local time step. In the original AVI method this is done using the CFL condition which uses elemental stiffness and mass matrices which do not exist in the CAG context. The proposed strategy may be applied to other spatial discretization methods based on the weakened weak (W2) form, for example nodal integration and Smoothed FEM. An example illustrated that the proposed strategy renders the time stepping scheme sufficiently stable and efficient. Problems regarding resonances as observed in r-RESPA or in AVI applied to molecular dynamics did not appear in the numerical tests.

Acknowledgments

This research has been funded by the Austrian Science Fund (FWF) under project number P20419-N14.

All algorithms are implemented in the software package SLang-TNG (URL: tng.tuxfamily.org) and distributed under the BSD open source license.

References

[1] G.R. Liu, A G space theory and a weakened weak (W2) form for a unified formulation of compatible and incompatible methods. Part I: Theory, Int. J. Numer. Methods Engrg. 81 (2009) 1093–1126.
[2] S. Wolff, C. Bucher, A finite element method based on C0-continuous assumed gradients, Int. J. Numer. Methods Engrg. 86 (2011) 876–914.
[3] T. Belytschko, R. Mullen, Mesh partitions of explicit–implicit time integration, in: K. Mathe, J. Oden, W. Wunderlich (Eds.), Formulations Computational

Algorithms in Finite Element Analysis, MIT press, New York, pp. 673–690.
[4] T.J.R. Hughes, W. Liu, Implicit–explicit finite elements in transient analysis: stability theory, J. Appl. Mech. 45 (1978) 371–374.
[5] T. Belytschko, R. Mullen, Explicit integration of structural problems, Finite Elem. Nonlinear Mech. 2 (1977) 669–720.
[6] M.O. Neal, T. Belytschko, Explicit–explicit subcycling with non-integer time step ratios for structural dynamic systems, Comput. Struct. 31 (6) (1989) 871–880.
[7] P. Smolinski, Subcycling integration with non-integer time steps for structural dynamics problems, Comput. Struct. 59 (2) (1996) 273–281.
[8] P. Smolinski, S. Sleith, Explicit multi-time step methods for structural dynamics, in: New Method in Transient Analysis, second ed., ASME, 1992, pp. 1–4.
[9] W.J.T. Daniel, A study of the stability of subcycling algorithms in structural dynamics, Comput. Methods Appl. Mech. Engrg. 156 (1–4) (1998) 1–13.
[10] M. Tuckerman, B.J. Berne, G.J. Martyna, Reversible multiple time scale molecular dynamics, J. Chem. Phys. 97 (3) (1992) 1990–2001.
[11] A. Lew, M. Ortiz, Asynchronous variational integrators, in: Geometry, Mechanics and Dynamics, Springer, 2002, pp. 91–110.
[12] A. Lew, J.E. Marsden, M. Ortiz, M. West, Asynchronous variational integrators, Arch. Ration. Mech. Anal. 167 (2) (2003) 85–146.
[13] A. Lew, Variational time integrators in computational solid mechanics, Ph.D. Thesis, California Institute of Technology, 2003.
[14] A. Lew, J.E. Marsden, M. Ortiz, M. West, Variational time integrators, Int. J. Numer. Methods Engrg. 60 (1) (2004) 153–212.
[15] K. Kale, A. Lew, Parallel asynchronous variational integrators, Int. J. Numer. Methods Engrg. 70 (2007) 291–321.
[16] M. Focardi, P.M. Mariano, Convergence of asynchronous variational integrators in linear elastodynamics, Int. J. Numer. Methods Engrg. 75 (2008) 755–769.
[17] W. Fong, E. Darve, A. Lew, Stability of asynchronous variational integrators, J. Comput. Phys. 227 (18) (2008) 8367–8394.
[18] M. Gates, K. Matous, M.T. Heath, Asynchronous multi-domain variational integrators for non-linear problems, Int. J. Numer. Methods Engrg. 76 (29–32) (2008) 1353–1378.
[19] M. Benes, K. Matous, Asynchronous multi-domain variational integrators for nonlinear hyperelastic solids, Comput. Methods Appl. Mech. Engrg. 199 (29–32) (2010) 1992–2013.
[20] D. Harmon, E. Vouga, B. Smith, R. Tamstorf, E. Grinspun, Asynchronous contact mechanics, in: ACM SIGGRAPH 2009 papers, SIGGRAPH '09, ACM, New York, NY, USA, 2009, pp. 87:1–87:12.
[21] S. Wolff, C. Bucher, Asynchronous collision integrators: Explicit treatment of unilateral contact with friction and nodal restraints, submitted for publication.
[22] J. Bonet, H. Marriott, O. Hassan, An averaged nodal deformation gradient linear tetrahedral element for large strain explicit dynamic applications, Commun. Numer. Methods Engrg. 17 (8) (2001) 551–561.
[23] C. Dohrmann, M.W. Heinstein, J. Jung, S.W. Key, W.R. Witkowski, Node-based uniform strain elements for three-node triangular and four-node tetrahedral meshes, Int. J. Numer. Methods Engrg. 47 (9) (2000) 1549–1568.
[24] P. Krysl, B. Zhu, Locking-free continuum displacement finite elements with nodal integration, Int. J. Numer. Methods Engrg. 76 (2008) 1020–1043.
[25] G.R. Liu, K. Dai, T. Nguyen, A smoothed finite element method for mechanics problems, Comput. Mech. 39 (6) (2007) 859–877.
[26] G.R. Liu, T. Nguyen-Thoi, H. Nguyen-Xuan, K.Y. Lam, A node-based smoothed finite element method (NS-FEM) for upper bound solutions to solid mechanics problems, Comput. Struct. 87 (1–2) (2009) 14–26.
[27] E.P. Kasper, R.L. Taylor, A mixed-enhanced strain method. Part I: Geometrically linear problems, Comput. Struct. 75 (2000) 237–250.
[28] S. Wolff, C. Bucher, Some aspects when applying continuous assumed gradient methods to explicit dynamics, J. Eng. Mech 138 (7) (2012) 910–913.
[29] B.I. Wohlmuth, A mortar finite element method using dual spaces for the Lagrange multiplier, SIAM J. Numer. Anal. 38 (1998) 989–1012.
[30] J.A. Izaguirre, Longer time steps for molecular dynamics, Tech. Rep., University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1999.
[31] R. Courant, K. Friedrichs, H. Lewy, On the partial difference equations of mathematical physics, IBM J. Res. Dev. 11 (1967) 215–234.
[32] A. Üngör, A. Sheffer, Tent-Pitcher: a meshing algorithm for space–time discontinuous Galerkin methods, in: Proc. Ninth International Meshing Roundtable, Sandia National Laboratories, 2000, pp. 111–122.