INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2016; **107**:603–630
Published online 23 December 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.5183

An accelerated, convergent, and stable nodal integration in Galerkin meshfree methods for linear and nonlinear mechanics

Michael Hillman and Jiun-Shyan Chen*,†

Department of Structural Engineering, University of California, 9500 Gilman Drive, La Jolla, San Diego, CA 92093-0085, U.S.A.

SUMMARY

Convergent and stable domain integration that is also computationally efficient remains a challenge for Galerkin meshfree methods. High order quadrature can achieve stability and optimal convergence, but it is prohibitively expensive for practical use. On the other hand, low order quadrature consumes much less CPU but can yield non-convergent, unstable solutions. In this work, an accelerated, convergent, and stable nodal integration is developed for the reproducing kernel particle method. A stabilization scheme for nodal integration is proposed based on implicit gradients of the strains at the nodes that offers a computational cost similar to direct nodal integration. The method is also formulated in a variationally consistent manner, so that optimal convergence is achieved. A significant efficiency enhancement over a comparable stable and convergent nodal integration scheme is demonstrated in a complexity analysis and in CPU time studies. A stability analysis is also given, and several examples are provided to demonstrate the effectiveness of the proposed method for both linear and nonlinear problems. Copyright © 2015 John Wiley & Sons, Ltd.

Received 19 August 2015; Revised 11 November 2015; Accepted 27 November 2015

KEY WORDS: nodal integration; reproducing kernel particle method; stabilization; variational consistency; implicit gradient

1. INTRODUCTION

Galerkin meshfree methods [1–12] offer advantages over traditional finite element methods in classes of problems involving damage, fracture, fragmentation, material flow, and complex evolving multi-body contact, among others. There are also additional advantages such as straightforward adaptive h-refinement and p-refinement, arbitrary choice of continuity, embedding of the characteristic behavior of the solution at hand, and the method is not plagued by mesh alignment, mesh entanglement, time-consuming mesh generation, and other issues related to mesh-based methods.

An array of considerations, however, come with the employment of these methods, and in particular, domain integration needs careful attention. Meshfree shape functions are in general rational and their overlapping supports may form complicated structures, both of which contribute to difficulty in domain integration, and the choice of quadrature greatly affects the stability and convergence of the numerical solution. High order quadrature can offer stability and optimal convergence, but it is prohibitively expensive for practical use. On the other hand, low order quadrature consumes less CPU but can yield non-convergent, unstable solutions [13–17]. In order to make effective large-scale simulations feasible, numerical integration must be efficient and yield convergent, stable solutions.

The importance of quadrature for Galerkin meshfree methods has been the subject of many papers since their inception (cf. [13–27]). Nodal integration is an attractive choice because of its simplicity, efficiency, and the ability to maintain meshfree characteristics as a whole. However,

*Correspondence to: Jiun-Shyan Chen, Department of Structural Engineering, University of California, 9500 Gilman Drive, San Diego, La Jolla, CA 92093-0085, U.S.A.
†E-mail: js-chen@ucsd.edu

Copyright © 2015 John Wiley & Sons, Ltd.

directly integrating at nodes results in stability issues [13–15] as well as sub-optimal convergence [13, 15, 17], and requires special techniques in order to overcome these difficulties. Several nodal integration methods have been developed that are convergent and stable [13, 25, 28, 29], but in the end sacrifice efficiency because of the approaches taken to ensure accuracy and stability. Another issue with these methods is the tuning of stabilization parameters for optimal accuracy and convergence.

The instability in direct nodal integration is due to the fact that for discretizations with spacing h, oscillating modes of wavelength $2h$ are admitted in the solution with little or no energy due to gradients being sampled only at the nodes [13–15]. Least-squares stabilization [13] and Taylor series expansions [28, 30] alleviate the instability by including second order derivatives that are non-zero at nodes, although this places a computational burden on the method because of the expensive high order derivatives involved. Stress points have also been utilized to avoid the instability [21, 22], but this technique requires additional stabilization such as least-squares to ensure stability and accuracy [31]. A stabilized conforming nodal integration (SCNI) has been proposed [15, 23, 32] where derivatives are not directly evaluated at nodes, which circumvents the instability without the use of any derivatives or additional sampling points. This method, however, also requires additional stabilization to ensure reliable solutions in all situations [25], which involves evaluating derivatives in nodal sub-cells and requires high computational effort.

Apart from stability, improper selection of quadrature schemes can also result in non-convergent solutions in meshfree methods [15–17]. Galerkin orthogonality is violated with the employment of insufficient quadrature in the discrete form, and the Galerkin method loses the best approximation property according to Strang's first lemma [33]. A general framework of variational consistency for arbitrary order Galerkin exactness has been introduced in [17] where a restoration of Galerkin orthogonality for a given polynomial order of solution can be achieved. Test and trial functions can be constructed under this framework such that accuracy and optimal convergence are attained with lower-order quadrature [17]. The earliest example of this technique for meshfree methods is SCNI [15], which satisfies the first order variational consistency condition and exhibits optimal convergence for approximation spaces with linear completeness. This method was later extended in [34, 35] to higher order schemes, which also have been shown to be very effective when compared to standard integration techniques.

Lowering both the number of evaluation points and the cost at each point can greatly lower computational cost. One approach to accomplish the latter is to employ implicit gradients, which were first introduced for synchronized convergence in [36, 37]. Using this approach, the completeness of gradients can be embedded into the meshfree approximation, and this avoids the complexity of direct differentiation of meshfree shape functions. Implicit gradients have also been utilized in the regularization of strain localization [38] to avoid ambiguous boundary conditions associated with the direct gradient methods. In [39], an implicit gradient method was introduced to reduce the cost of meshfree collocation methods that require higher order derivatives of the approximation functions.

The main burden of current stabilized nodal integration is either the employment of additional evaluation points or stabilizing terms that involve higher order derivatives, both of which are CPU demanding for meshfree methods. Another issue with these methods is the choice of numerical parameters associated with the stabilization. The objective at hand then is to achieve stability in nodal integration without additional sampling points or high order derivatives and ideally have no parameters to tune. In this work, a stabilization scheme for nodal integration is proposed for the reproducing kernel particle method (RKPM) based on extrapolation of the approximation at nodes using implicit gradients. The method is devoid of tunable parameters and offers a computational cost similar to direct nodal integration. The method is also formulated in a variationally consistent manner so that optimal convergence is achieved. A significant efficiency enhancement over a comparable stable and convergent nodal integration scheme is demonstrated in a complexity analysis and in CPU time studies. A stability analysis is also given, and several examples are provided to demonstrate the effectiveness of the proposed method for both linear and nonlinear problems.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

The remainder of this paper is organized as follows. In Section 2, various nodal integration schemes are discussed along with the difficulties associated with them. Section 3 introduces a new stabilization for nodal integration, where its stability is demonstrated, complexity comparisons are made, and variationally consistent test functions are introduced for the method. Several numerical examples are given in Section 4, and concluding remarks are given in Section 5.

2. NODAL INTEGRATION IN MESHFREE METHODS

2.1. Reproducing kernel approximation

The reproducing kernel (RK) approximation $u^h(\mathbf{x})$ of a function $u(\mathbf{x})$ using a set of NP nodes is constructed by the product of a kernel function $\Phi_a(\mathbf{x}-\mathbf{x}_I)$ with compact support measure a and a correction function [3]:

$$u^h(\mathbf{x}) = \sum_{I=1}^{NP} \left\{ \sum_{|\alpha|\leqslant n} b_\alpha(\mathbf{x})(\mathbf{x}-\mathbf{x}_I)^\alpha \right\} \Phi_a(\mathbf{x}-\mathbf{x}_I)u_I \equiv \sum_{I=1}^{NP} \Psi_I(\mathbf{x})u_I. \tag{1}$$

In the preceding equation, the multi-index notation $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d)$ with dimension d has been introduced with $|\alpha| = \sum_{i=1}^{d} \alpha_i$, $\mathbf{x}^\alpha \equiv x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \ldots \cdot x_d^{\alpha_d}$, $\mathbf{x}_I^\alpha \equiv x_{I1}^{\alpha_1} \cdot x_{I2}^{\alpha_2} \cdot \ldots \cdot x_{Id}^{\alpha_d}$, $b_\alpha = b_{\alpha_1\alpha_2\cdots\alpha_d}$, the term $\{(\mathbf{x}-\mathbf{x}_I)^\alpha\}_{|\alpha|\leqslant n}$ is the set of basis functions, $\{b_\alpha(\mathbf{x})\}_{|\alpha|\leqslant n}$ are coefficients of those bases, $\Psi_I(\mathbf{x})$ is the RK shape function, and u_I is the associated coefficient. The kernel function $\Phi_a(\mathbf{x}-\mathbf{x}_I)$ determines the smoothness of the approximation function, for example, a cubic B-spline function gives C^2 continuity.

The set of coefficients $\{b_\alpha(\mathbf{x})\}_{|\alpha|\leqslant n}$ is determined by enforcing the following reproducing conditions:

$$\sum_{I=1}^{NP} \Psi_I(\mathbf{x})\mathbf{x}_I^\alpha = \mathbf{x}^\alpha, \quad |\alpha| \leqslant n. \tag{2}$$

With $\{b_\alpha(\mathbf{x})\}_{|\alpha|\leqslant n}$ obtained from (2), the RK shape functions are constructed as

$$\Psi_I(\mathbf{x}) = \mathbf{H}^T(\mathbf{0})\mathbf{M}^{-1}(\mathbf{x})\mathbf{H}(\mathbf{x}-\mathbf{x}_I)\Phi_a(\mathbf{x}-\mathbf{x}_I) \tag{3}$$

where

$$\mathbf{M}(\mathbf{x}) = \sum_{I=1}^{NP} \mathbf{H}(\mathbf{x}-\mathbf{x}_I)\mathbf{H}^T(\mathbf{x}-\mathbf{x}_I)\Phi_a(\mathbf{x}-\mathbf{x}_I), \tag{4}$$

which is called the *moment matrix*, and the vector $\mathbf{H}(\mathbf{x}-\mathbf{x}_I)$ is the column vector of $\{(\mathbf{x}-\mathbf{x}_I)^\alpha\}_{|\alpha|\leqslant n}$, for example, with linear basis in two dimensions

$$\mathbf{H}(\mathbf{x}-\mathbf{x}_I) = \left[\, 1,\ x-x_I,\ y-y_I \,\right]^T, \tag{5}$$

and for linear basis in three dimensions, we have

$$\mathbf{H}(\mathbf{x}-\mathbf{x}_I) = \left[\, 1,\ x-x_I,\ y-y_I,\ z-z_I \,\right]^T. \tag{6}$$

The reproducing conditions (2) are met, provided the moment matrix (4) is invertible, and this requires a sufficient number of non-coplanar nodes covering the evaluation point $\mathbf{x}$ [4].

Note that the shape functions in (3) are rational functions, and the overlapping supports form complicated structures that need to be integrated. The result of these two issues is either very high order quadrature or special techniques such as those introduced in Section 3.4 must be used in order

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

to obtain accurate and convergent solutions. In addition, the employment of nodal integration with these shape functions can also result in instability due to the possibility of sampling gradients of zero at nodal points, as discussed in the following sections.

2.2. Model problem

The governing equations for elasticity in d-dimensions are considered as a model problem for numerical integration of Galerkin meshfree methods:

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} + \mathbf{b} &= \mathbf{0} \text{ on } \Omega \\ \mathbf{n} \cdot \boldsymbol{\sigma} &= \mathbf{h} \text{ on } \partial\Omega_h \\ \mathbf{u} &= \mathbf{g} \text{ on } \partial\Omega_g \end{aligned} \tag{7}$$

where $\mathbf{u}$ is the displacement field, $\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u})$ is the Cauchy stress tensor, $\mathbf{C}$ is the elasticity tensor, $\boldsymbol{\varepsilon}(\mathbf{u}) = 1/2(\nabla \otimes \mathbf{u} + \mathbf{u} \otimes \nabla)$ is the strain tensor, $\mathbf{n}$ is the surface normal on $\partial\Omega$, $\mathbf{b}$ is the body force, $\mathbf{h}$ is the prescribed traction on $\partial\Omega_h$, and $\mathbf{g}$ is the prescribed displacement on $\partial\Omega_g$.

The weak form of (7) asks to find $\mathbf{u} \in \left[H_g^1\right]^d$, such that for all kinematically admissible test functions $\mathbf{v} \in \left[H_0^1\right]^d$ the following holds:

$$a(\mathbf{v}, \mathbf{u}) = (\mathbf{v}, \mathbf{b})_\Omega + (\mathbf{v}, \mathbf{h})_{\partial\Omega_h} \tag{8}$$

where the linear and bilinear forms for the above are defined as

$$a(\mathbf{v}, \mathbf{u}) = \int_\Omega \boldsymbol{\varepsilon}(\mathbf{v}) : \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u}) \mathrm{d}\Omega \tag{9}$$

$$(\mathbf{v}, \mathbf{b})_\Omega = \int_\Omega \mathbf{v} \cdot \mathbf{b} \mathrm{d}\Omega \tag{10}$$

$$(\mathbf{v}, \mathbf{h})_{\partial\Omega_h} = \int_{\partial\Omega_h} \mathbf{v} \cdot \mathbf{h} \mathrm{d}\Gamma. \tag{11}$$

Using finite dimensional approximations for $\mathbf{u}$ and $\mathbf{v}$, the discrete form of (8) asks to find $\mathbf{u}^h \in [H_g^1]^d$ such that the following equation holds for all kinematically admissible $\mathbf{v}^h \in \left[H_0^1\right]^d$:

$$a(\mathbf{v}^h, \mathbf{u}^h) = (\mathbf{v}^h, \mathbf{b})_\Omega + (\mathbf{v}^h, \mathbf{h})_{\partial\Omega_h}. \tag{12}$$

2.3. Nodal integration methods

2.3.1. Direct nodal integration. The discrete form in the previous section necessitates numerical quadrature. Perhaps the simplest quadrature rule for meshfree methods is nodal integration, where shape functions are directly evaluated at the nodes, which has been termed *direct nodal integration* (DNI) in the literature. Using this technique, the bilinear form in (12) is evaluated as

$$a_D\langle \mathbf{v}^h, \mathbf{u}^h \rangle = \sum_{L=1}^{NP} \boldsymbol{\varepsilon}_L(\mathbf{v}^h) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\mathbf{u}^h) W_L \tag{13}$$

where $\boldsymbol{\varepsilon}_L(\mathbf{u}^h) = \boldsymbol{\varepsilon}\left(\mathbf{u}^h(\mathbf{x}_L)\right)$, $a_D\langle \cdot, \cdot \rangle$ is the DNI quadrature version of the bilinear form in (12) and W_L is the integration weight associated with node L.

For DNI and discretizations with spacing h, oscillating modes of wavelength $2h$ are admitted in the solution of the discrete problem with little or no energy due to gradients being evaluated directly at the nodes [13–15].

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

2.3.2. Stabilized conforming nodal integration. SCNI has been introduced in [15, 23] to remedy the instability in DNI. In this method, gradients are smoothed over conforming cells that partition the domain as shown in Figure 1(a), and the smoothing operation is converted to a cell boundary integral using the divergence theorem. In this way, gradients are not directly evaluated at the nodes and the oscillating modes in DNI are precluded. The cells can be generated by, for example, Voronoi diagrams.

The smoothing zones are chosen as conforming so that the first order integration constraint (for Galerkin linear exactness) is satisfied [15]. This condition is the requirement to satisfy the following divergence equality with the set of test functions and the chosen numerical integration:

$$\overset{\wedge}{\int_{\Omega}} \nabla \Psi_I \, \mathrm{d}\Omega = \overset{\wedge}{\int_{\partial\Omega}} \Psi_I \mathbf{n} \mathrm{d}\Gamma \tag{14}$$

where '^' denotes numerical integration. SCNI employs divergence with a smoothed gradient $\tilde{\nabla}$ in each nodal representative domain Ω_L by

$$\tilde{\nabla} \Psi_I(\mathbf{x}_L) = \frac{1}{W_L} \int_{\Omega_L} \nabla \Psi_I \, \mathrm{d}\Omega = \frac{1}{W_L} \int_{\partial\Omega_L} \Psi_I \mathbf{n} \mathrm{d}\Gamma. \tag{15}$$

If the smoothing domains $\{\Omega_L\}_{L=1}^{NP}$ are conforming such as those in Figure 1(a), nodal integration with smoothed gradients $\tilde{\nabla}$ meets the condition (14) when the boundary integration for smoothing (15) and boundary conditions in (12) use the same quadrature rule [15]:

$$\overset{\wedge}{\int_{\Omega}} \tilde{\nabla} \Psi_I \, \mathrm{d}\Omega = \sum_{L=1}^{NP} \tilde{\nabla} \Psi_I(\mathbf{x}_L) W_L = \sum_{L=1}^{NP} \overset{\wedge}{\int_{\partial\Omega_L}} \Psi_I \mathbf{n} \mathrm{d}\Gamma = \overset{\wedge}{\int_{\partial\Omega}} \Psi_I \mathbf{n} \mathrm{d}\Gamma. \tag{16}$$

With the smoothed gradient operator (15), the bilinear form in (12) is evaluated with nodal integration as

$$a_S \langle \mathbf{v}^h, \mathbf{u}^h \rangle = \sum_{L=1}^{NP} \tilde{\boldsymbol{\varepsilon}}_L(\mathbf{v}^h) : \mathbf{C} : \tilde{\boldsymbol{\varepsilon}}_L(\mathbf{u}^h) W_L \tag{17}$$

where $a_S \langle \cdot, \cdot \rangle$ is the bilinear form for SCNI, and $\tilde{\boldsymbol{\varepsilon}}_L \left(\mathbf{u}^h\right) = 1/2(\tilde{\nabla} \otimes \mathbf{u}(\mathbf{x}_L) + \mathbf{u}(\mathbf{x}_L) \otimes \tilde{\nabla})$ is the smoothed strain evaluated at $\mathbf{x}_L$.

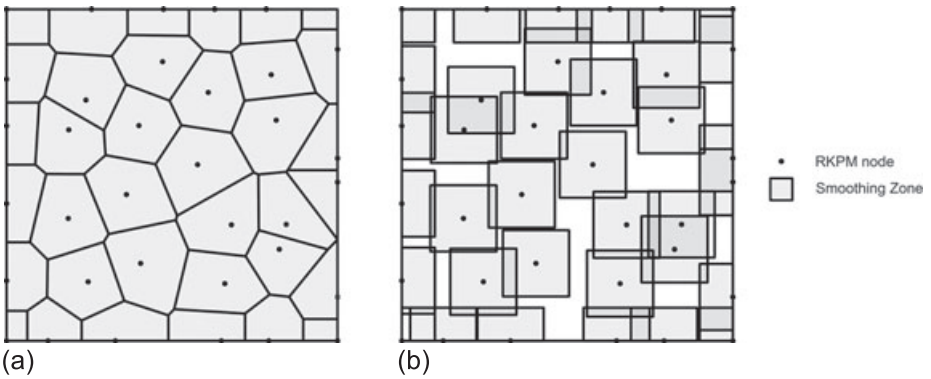

Figure 1. Nodal smoothing zones for (a) stabilized conforming nodal integration (SCNI) and (b) stabilized non-conforming nodal integration (SNNI).

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

2.3.3. Stabilized non-conforming nodal integration. For classes of problems where material separation occurs, shape functions can be reconstructed so that nodes reassociate. This formulation is termed *semi-Lagrangian* as nodes are located at material points, while kernels are defined in the current configuration [40, 41]. Reconstructing conforming representative nodal domains necessary for SCNI, however, is extremely tedious for this class of problems, and also prohibitively expensive in three dimensions. In this case, relaxation of the conforming condition on the smoothing zones, as shown in Figure 1(b), is a necessary simplification, and results in stabilized non-conforming nodal integration (SNNI) [40, 41], which employs the smoothed gradient

$$\tilde{\nabla}\Psi_I(\mathbf{x}_L) = \frac{1}{\bar{W}_L}\int_{\partial\bar{\Omega}_L} \Psi_I \mathbf{n}\mathrm{d}\Gamma \tag{18}$$

where $\bar{W}_L = \int_{\bar{\Omega}_L} \mathrm{d}\Omega$ and $\bar{\Omega}_L$ is a non-conforming nodal representative domain, for example, box shaped. SNNI also possesses superior stability over DNI but can yield non-convergent solutions under certain discretizations because of the relaxation of the conforming conditions and subsequent failure to meet the integration constraint [17]. In this paper, SNNI and its stabilized and convergent variants are considered as reference methods for comparison, because they are often employed in problems where SCNI is not possible.

2.4. Instability in nodal integration

With limited boundary influence of the discretization at hand, that is, when the surface to volume ratio is small, or for sufficiently fine discretizations, the instability in nodal integration methods can be examined in a unified fashion as follows.

Consider the calculation of the internal energy by nodal integration for the one-dimensional elasticity problem in a uniform discretization. For an interior node (with no influence from the boundary), consider m nodes covering the nodal location $\mathbf{x}_L$ on each side of the node, giving a total of $2m + 1$ nodes covering the location. A gradient is computed at this node as

$$u^h_{,x}(x_L) = \Psi_{L-m,x}(x_L)u_{L-m} + \ldots + \Psi_{L,x}(x_L)u_L + \ldots + \Psi_{L+m,x}(x_L)u_{L+m}. \tag{19}$$

When sufficiently smooth (at least C^1), symmetric kernels are employed, the term $\Psi_{L,x}(\mathbf{x}_L)$ can be shown to be zero when computed directly, and also when using averaging and smoothing operations. The remaining terms, using the uniformity and symmetry of the discretization are

$$\begin{aligned} u^h_{,x}(x_L) = {} & 2\Psi_{L-m,x}(x_L)\,(u_{L-m} - u_{L+m}) + 2\Psi_{L-m+1,x}(x_L)\,(u_{L-m+1} - u_{L+m-1}) + \ldots \\ & + 2\Psi_{L-1,x}(x_L)\,(u_{L-1} - u_{L+1})\,. \end{aligned} \tag{20}$$

For modes of alternating displacement of unity at each node $u_{L-p} = u_{L+p}$ for any p, the gradient is identically zero, and interior nodes have no contribution to the internal energy of the system. This is the essential difficulty of undersampling using only nodes.

Note that no particular property of the shape functions other than smoothness and symmetry were invoked. Thus, regardless of whether the gradients are smoothed, direct, or averaged, the results still hold and spurious oscillatory zero-energy or low-energy modes can appear in the solution with any of these nodal integrations. The result also suggests that only energy from the boundary can be expected to keep the mode in check, which is similar to the discussions in the literature [25]. The consequence is, for discretizations that have sufficiently small surface to volume ratio or for sufficiently fine discretizations, the modes are expected to be present in the solution.

To illustrate this instability, consider the node distribution in Figure 2 for construction of a linear elastic stiffness matrix, discretized with the fully integrated bilinear finite element method (FEM) and nodally integrated linear RKPM with quartic B-spline kernels and a normalized support of 2.0. For two-dimensional elasticity, there are three proper zero-eigenvalue modes for translation and rotation. The fourth lowest eigenvalue mode for the fully integrated bilinear finite element method

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

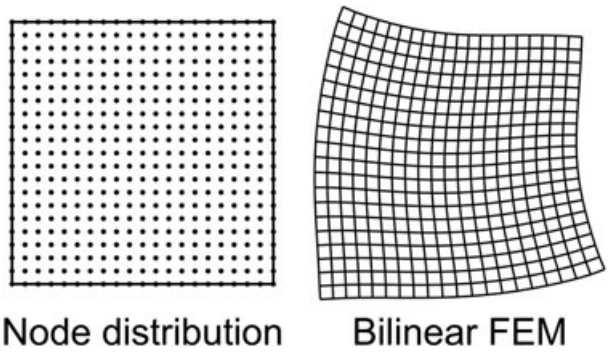

Figure 2. Discretization and fourth eigenvalue mode for the stiffness matrix for fully integrated finite elements.

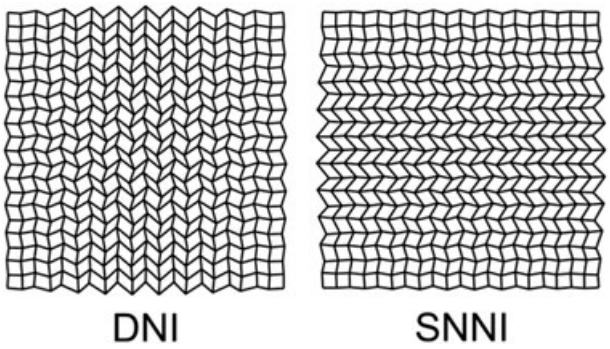

Figure 3. Fourth eigenvalue mode of the stiffness matrix for nodal integration methods.

shown in Figure 2 is a stable mode of deformation. In contrast, for DNI and SNNI, the first non-zero eigenvalue mode is a spurious, oscillatory mode as shown in Figure 3, where the node-to-node oscillations are clearly visible for both methods.

Because the cause of these unstable modes is the trial function gradients evaluating as zero, the choice of the test functions is irrelevant, and introducing different Petrov-Galerkin test functions cannot be expected to preclude the modes. Thus, modifying test functions for variational consistency discussed in Section 3.4 alone will not suppress these modes for uniform discretizations; although for non-uniform discretizations, it has been observed in [29] that the stability is improved.

2.5. *Additional stabilization for stabilized conforming nodal integration and stabilized non-conforming nodal integration*

The stabilization of these low energy modes in SCNI and SNNI can be accomplished by including strain averaging over subdivisions of the smoothing cells. For SCNI, each nodal representative domain can be triangulated, and for SNNI, the smoothing zones can simply be divided into equal parts, as shown in Figure 4. Strains are evaluated at the centroid $\mathbf{x}_L^K$ of NS sub-cells in order to provide additional coercivity and avoid undersampling of the saw-tooth modes. The form employed is based on maintaining satisfaction of the linear patch test for SCNI [25]:

$$a_{MS}\langle \mathbf{v}^h, \mathbf{u}^h \rangle = a_S\langle \mathbf{v}^h, \mathbf{u}^h \rangle + a_M\langle \mathbf{v}^h, \mathbf{u}^h \rangle \tag{21}$$

where $a_S\langle \mathbf{v}^h, \mathbf{u}^h \rangle$ is the bilinear form with SCNI or SNNI smoothing, $a_{MS}\langle \mathbf{v}^h, \mathbf{u}^h \rangle$ is the bilinear form for modified stabilized conforming nodal integration or modified stabilized non-conforming nodal integration (MSNNI) and

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

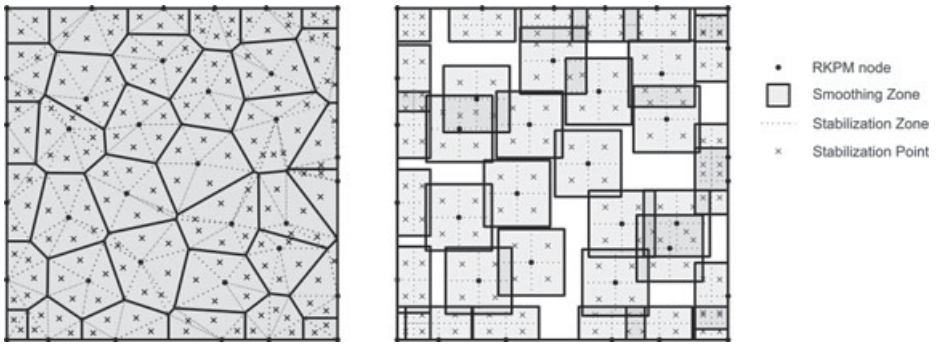

Figure 4. Stabilization schemes for (a) modified stabilized conforming nodal integration (MSCNI) and (b) modified stabilized non-conforming nodal integration (MSNNI).

$$a_M\langle \mathbf{v}^h, \mathbf{u}^h\rangle = \sum_{L=1}^{NP}\sum_{K=1}^{NS} c\left[\left(\tilde{\boldsymbol{\varepsilon}}_L(\mathbf{v}^h) - \boldsymbol{\varepsilon}_L^K(\mathbf{v}^h)\right) : \mathbf{C} : \left(\tilde{\boldsymbol{\varepsilon}}_L(\mathbf{u}^h) - \boldsymbol{\varepsilon}_L^K(\mathbf{u}^h)\right) W_K^L\right] \tag{22}$$

is the additional stabilization where $0.0 \leqslant c \leqslant 1.0$, $\boldsymbol{\varepsilon}_L^K\left(\mathbf{u}^h\right) = \boldsymbol{\varepsilon}\left(\mathbf{u}^h\left(\mathbf{x}_L^K\right)\right)$ is the strain evaluated at the centroid of the sub-cell, and W_L^K is the weight of the sub-cell calculated from the weight W_L. It is clear from (22) that for $c > 0.0$ additional coercivity is added to the solution. Herein, the value of $c = 0.2$ is chosen, as larger values have been shown to reduce solution accuracy [42]. Thus, a drawback of this stabilization is that an artificial parameter is introduced, which needs to be selected carefully to ensure stability and also avoid destroying the accuracy of the smoothed integration methods. The other drawback of this method is that many additional evaluation points are needed to provide stabilization, which is detrimental to efficiency.

3. ACCELERATED STABILIZED AND CONVERGENT NODAL INTEGRATION

3.1. Implicit gradient expansion

Consider the following extrapolation of an RK approximation $u^h(\mathbf{x})$ from a point $\bar{\mathbf{x}}$:

$$u^h(\mathbf{x}) \approx \hat{u}^h(\mathbf{x}) = \sum_{I=1}^{NP} \Psi_I(\bar{\mathbf{x}}) u_I + \sum_{I=1}^{NP} \boldsymbol{\Psi}_I^{\nabla}(\bar{\mathbf{x}}) \cdot (\mathbf{x} - \bar{\mathbf{x}}) u_I \tag{23}$$

where '$\cdot$' denotes an inner product and $\boldsymbol{\Psi}_I^{\nabla}(\mathbf{x})$ is a vector of implicit gradient approximations to be defined. The functions $\{\boldsymbol{\Psi}_I^{\nabla}(\mathbf{x})\}_{I=1}^{NP}$ are constructed such that the following equivalence is satisfied:

$$\sum_{I=1}^{NP} \boldsymbol{\Psi}_I^{\nabla}(\bar{\mathbf{x}}) \cdot (\mathbf{x} - \bar{\mathbf{x}}) \mathbf{x}_I^{\alpha} = \sum_{I=1}^{NP} \nabla \Psi_I(\bar{\mathbf{x}}) \cdot (\mathbf{x} - \bar{\mathbf{x}}) \mathbf{x}_I^{\alpha}, \quad |\alpha| \leqslant n. \tag{24}$$

It can be seen that the expansion in (23) is equivalent to the first order Taylor expansion when $u^h(\mathbf{x})$ is of order $|\alpha| \leqslant n$, and is termed the *implicit gradient expansion* herein.

Utilizing (2), the condition (24) can be rephrased into the following gradient reproducing conditions:

$$\sum_{I=1}^{NP} \boldsymbol{\Psi}_I^{\nabla}(\mathbf{x})\, \mathbf{x}_I^{\alpha} = \nabla \mathbf{x}^{\alpha}, \quad |\alpha| \leqslant n. \tag{25}$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

Construction of the functions $\boldsymbol{\Psi}_I^{\nabla}(\mathbf{x})$ can be performed by employing the same form as $\Psi_I(\mathbf{x})$ in (1):

$$\Psi_{Ii}^{\nabla}(\mathbf{x}) = \left\{ \sum_{|\alpha|\leq n} (\mathbf{x}-\mathbf{x}_I)^{\alpha} b_{\alpha}^{i}(x) \right\} \Phi_a(\mathbf{x}-\mathbf{x}_I). \tag{26}$$

The unknown coefficients $\{b_{\alpha}^{i}(\mathbf{x})\}_{|\alpha|\leq n}$ are determined by enforcing (25), consequently, the functions in (26) are constructed as [38]:

$$\Psi_{Ii}^{\nabla}(\mathbf{x}) = \mathbf{H}_i^T \mathbf{M}^{-1}(\mathbf{x}) \mathbf{H}(\mathbf{x}-\mathbf{x}_I) \Phi_a(\mathbf{x}-\mathbf{x}_I) \tag{27}$$

where $\mathbf{H}_i$ is a column vector of the form:

$$\mathbf{H}_i = [0, \ \ldots, \ 0, \ \underset{\substack{\uparrow \\ (i+1)^{th} \text{ entry}}}{-1}, \ 0, \ \ldots, \ 0]^T . \tag{28}$$

For example, in two dimensions with linear basis, the vector $\mathbf{H}_i$ takes on the following values:

$$\begin{aligned} \mathbf{H}_1 &= [\ 0, -1, \ 0 \ ]^T \\ \mathbf{H}_2 &= [\ 0, \ 0, \ -1 \ ]^T. \end{aligned} \tag{29}$$

In three dimensions with linear basis, the vector $\mathbf{H}_i$ takes on the values:

$$\begin{aligned} \mathbf{H}_1 &= [\ 0, -1, \ 0, \ \ 0 \ ]^T \\ \mathbf{H}_2 &= [\ 0, \ 0, \ -1, \ 0 \ ]^T \\ \mathbf{H}_3 &= [\ 0, \ 0, \ \ 0, \ -1 \ ]^T. \end{aligned} \tag{30}$$

One can observe that the construction of (27) has the same complexity as (3), and the resulting computational cost of the stabilization introduced by the expansion (23) described in the next subsection is similar to direct nodal integration.

3.2. Naturally stabilized nodal integration

The implicit gradient expansion introduced in the previous section leads naturally to a stabilized integration when employed for the strain at each node. The form of stabilization is similar to that introduced in the finite element context in [43], but here introduced in the Cartesian coordinates and instead uses an extrapolation of the type (23). The ideas here are presented in three dimensions, but the application to lower dimensions is straightforward.

First, consider the decomposition of the bilinear form in (12) with nodal integration zones Ω_L:

$$a(\mathbf{v}^h, \mathbf{u}^h) = \sum_{L=1}^{NP} \left\{ \int_{\Omega_L} \boldsymbol{\varepsilon}(\mathbf{v}^h) : \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u}^h) \mathrm{d}\Omega \right\}. \tag{31}$$

Now consider an implicit gradient expansion of the form in (23) for the strains in Ω_L with $\bar{\mathbf{x}} = \mathbf{x}_L$:

$$\boldsymbol{\varepsilon}(\mathbf{u}^h(\mathbf{x})) \approx \boldsymbol{\varepsilon}_L(\mathbf{u}^h) + (x-x_L)\boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}_x^h) + (y-y_L)\boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}_y^h) + (z-z_L)\boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}_z^h) \equiv \hat{\boldsymbol{\varepsilon}}_L(\mathbf{u}^h) \tag{32}$$

where $\boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}_x^h) \equiv \boldsymbol{\varepsilon}\left(\hat{\mathbf{u}}_x^h(\mathbf{x}_L)\right)$, $\boldsymbol{\varepsilon}_L\left(\hat{\mathbf{u}}_y^h\right) \equiv \boldsymbol{\varepsilon}\left(\hat{\mathbf{u}}_y^h(\mathbf{x}_L)\right)$, $\boldsymbol{\varepsilon}_L\left(\hat{\mathbf{u}}_z^h\right) \equiv \boldsymbol{\varepsilon}\left(\hat{\mathbf{u}}_z^h(\mathbf{x}_L)\right)$, and

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

$$\begin{aligned}\hat{\mathbf{u}}^h_x(\mathbf{x}_L) &= \sum_{I=1}^{NP} \Psi^{\nabla}_{I1}(\mathbf{x}_L)\mathbf{u}_I \\ \hat{\mathbf{u}}^h_y(\mathbf{x}_L) &= \sum_{I=1}^{NP} \Psi^{\nabla}_{I2}(\mathbf{x}_L)\mathbf{u}_I \\ \hat{\mathbf{u}}^h_z(\mathbf{x}_L) &= \sum_{I=1}^{NP} \Psi^{\nabla}_{I3}(\mathbf{x}_L)\mathbf{u}_I\,.\end{aligned} \tag{33}$$

Substituting (32) into (31) for each of the strains, one obtains, for each nodal domain Ω_L:

$$\begin{aligned}\int_{\Omega_L} \hat{\boldsymbol{\varepsilon}}_L(\mathbf{v}^h) : \mathbf{C} : \hat{\boldsymbol{\varepsilon}}_L(\mathbf{u}^h)\mathrm{d}\Omega = &\int_{\Omega_L} \boldsymbol{\varepsilon}_L(\mathbf{v}^h) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\mathbf{u}^h)\mathrm{d}\Omega \\ &+ \int_{\Omega_L} (x - x_L)^2\, \boldsymbol{\varepsilon}_L(\hat{\mathbf{v}}^h_x) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}^h_x)\mathrm{d}\Omega \\ &+ \int_{\Omega_L} (y - y_L)^2\, \boldsymbol{\varepsilon}_L(\hat{\mathbf{v}}^h_y) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}^h_y)\mathrm{d}\Omega \\ &+ \int_{\Omega_L} (z - z_L)^2\, \boldsymbol{\varepsilon}_L(\hat{\mathbf{v}}^h_z) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}^h_z)\mathrm{d}\Omega\,.\end{aligned} \tag{34}$$

In arriving at (34), the following assumption has been utilized:

$$\int_{\Omega_L} (x - x_L)\,\mathrm{d}\Omega = \int_{\Omega_L} (y - y_L)\,\mathrm{d}\Omega = \int_{\Omega_L} (z - z_L)\,\mathrm{d}\Omega = 0. \tag{35}$$

The effect of the assumption (35) is expected to be negligible, because in general, nodes are located at or near the centroid of the domain Ω_L. Moreover, for nodes located on the edges of the domain, the condition (35) will, in general, be satisfied. In numerical experiments, it has been observed that these terms have very little effect and only increase computational cost.

Moving the constant quantities in (34) out of the integrals, it is possible to evaluate the non-constant terms analytically via

$$\begin{aligned}\int_{\Omega_L} \hat{\boldsymbol{\varepsilon}}_L(\mathbf{v}^h) : \mathbf{C} : \hat{\boldsymbol{\varepsilon}}_L(\mathbf{u}^h)\mathrm{d}\Omega = &\boldsymbol{\varepsilon}_L(\mathbf{v}^h) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\mathbf{u}^h)W_L + \boldsymbol{\varepsilon}_L(\hat{\mathbf{v}}^h_x) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}^h_x)M_{Lx} \\ &+ \boldsymbol{\varepsilon}_L(\hat{\mathbf{v}}^h_y) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}^h_y)M_{Ly} + \boldsymbol{\varepsilon}_L(\hat{\mathbf{v}}^h_z) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}^h_z)M_{Lz}\end{aligned} \tag{36}$$

where

$$\begin{aligned}M_{Lx} &= \int_{\Omega_L} (x - x_L)^2\,\mathrm{d}\Omega \\ M_{Ly} &= \int_{\Omega_L} (y - y_L)^2\,\mathrm{d}\Omega \\ M_{Lz} &= \int_{\Omega_L} (z - z_L)^2\,\mathrm{d}\Omega\,,\end{aligned} \tag{37}$$

which are the second moments of inertia of each nodal domain. Employing (36) in (31), the numerical integration of the internal force results in a stabilized direct nodal integration:

$$a_{ND}\langle \mathbf{v}^h, \mathbf{u}^h\rangle = a_D\langle \mathbf{v}^h, \mathbf{u}^h\rangle + a_N\langle \mathbf{v}^h, \mathbf{u}^h\rangle \tag{38}$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

where $a_D\langle \mathbf{v}^h, \mathbf{u}^h\rangle$ is the quadrature version of the bilinear form with DNI (13), and

$$a_N\langle \mathbf{v}^h, \mathbf{u}^h\rangle = \sum_{L=1}^{NP} \Big\{ \boldsymbol{\varepsilon}_L(\hat{\mathbf{v}}_x^h) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}_x^h) M_{Lx} + \boldsymbol{\varepsilon}_L(\hat{\mathbf{v}}_y^h) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}_y^h) M_{Ly} + \boldsymbol{\varepsilon}_L(\hat{\mathbf{v}}_z^h) : \mathbf{C} : \boldsymbol{\varepsilon}_L(\hat{\mathbf{u}}_z^h) M_{Lz} \Big\} \tag{39}$$

is the stabilization introduced naturally by the implicit gradient expansion of the strains at each node. The additional terms in the stabilization *are necessarily positive* for non-zero strains and quadratic forms of $a(\cdot,\cdot)$ and thus provide additional coercivity. The constants associated with the additional terms occur completely naturally, and thus no tuning of any parameters is required, which is in contrast to other stabilized methods [13, 25]. Stabilization with (38) is termed *naturally stabilized nodal integration (NSNI)* herein. For nonlinear problems, we propose a stress update analogous to the form (39) for facilitation of stabilization in explicit dynamic formulations, which is given in Appendix A.

Let us reexamine the calculation of the internal energy in a uniform discretization of a one-dimensional elasticity problem, problematic for nodal integration. Again, considering a node with $2m+1$ nodes covering the nodal point, the computation of the stabilizing term $\varepsilon_L(\hat{u}_x^h) \equiv \hat{u}_{x,x}^h(x_L)$ for its contribution to the internal energy in one dimension is

$$\begin{aligned} \hat{u}_{x,x}^h(x_L) = {} & \Psi^{\nabla}_{L-m,x}(x_L) u_{L-m} + \Psi^{\nabla}_{L-m+1,x}(x_L) u_{L-m+1} + \ldots + \Psi^{\nabla}_{L,x}(x_L) u_L + \ldots \\ & + \Psi^{\nabla}_{L+m-1,x}(x_L) u_{L+m-1} + \Psi^{\nabla}_{L+m,x}(x_L) u_{L+m} \end{aligned} \tag{40}$$

where $\Psi^{\nabla}_{I,x} \equiv \Psi^{\nabla}_{I1,x}$. Using the symmetry and uniformity of the discretization one obtains

$$\hat{u}_{x,x}^h(x_L) = \Psi^{\nabla}_{L-m,x}(x_L)(u_{L-m}+u_{L+m}) + \Psi^{\nabla}_{L-m+1,x}(x_L)(u_{L-m+1}+u_{L+m-1}) + \ldots + \Psi^{\nabla}_{L,x}(x_L) u_L. \tag{41}$$

Modes of alternating displacement of unity at each node leads to

$$\hat{u}_{x,x}^h(x_L) = \pm \left(2\Psi^{\nabla}_{L-m,x}(x_L) - 2\Psi^{\nabla}_{L-m+1,x}(x_L) + \ldots + (-1)^m \Psi^{\nabla}_{L,x}(x_L) \right), \tag{42}$$

which is *non-zero* for sufficiently smooth kernels, in contrast to (20), and thus the oscillatory mode is unlikely to be admitted due to the associated additional energy. It is interesting to note the implicit gradient expansion introduces terms that are not zero at nodal locations, and undersampling of saw-tooth modes is avoided.

Figure 5 shows the fourth eigenvalue mode of the previously employed RK discretization for stability analysis using MSNNI and NSNI, both showing very good agreement with the mode shape of FEM in Figure 2. The corresponding eigenvalues are 1.28 and 1.32 for MSNNI and NSNI, respectively, which are also in agreement of the value of 1.30 for fully integrated FEM.

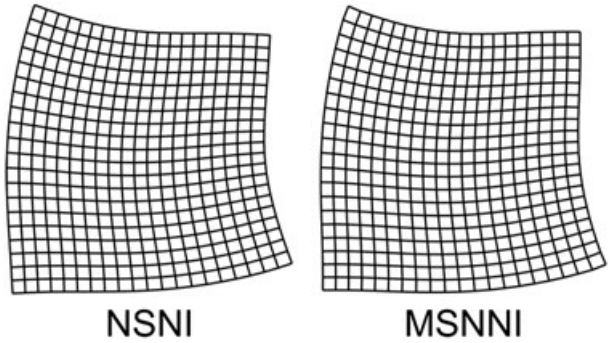

Figure 5. Fourth eigenvalue mode of the stiffness matrix for stabilized nodal integration methods.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

3.3. *Complexity analysis of modified stabilized non-conforming nodal integration and naturally stabilized nodal integration*

To estimate the operation counts, multiplication and division (M/D), and addition and subtraction (A/S) are considered. For MSNNI in d-dimensions, for each nodal integration point the functions $\Psi_I(\mathbf{x})$ must be calculated $2d+1$ times in order to construct smoothed gradients with box smoothing zones and evaluate the shape functions at the node itself. In addition, each gradient $\nabla\Psi_I(\mathbf{x})$ must be calculated in the center of the sub-cells. Consequently, the total operation count for one node, for evaluating all shape functions necessary for forming the stiffness matrix, body force, and mass matrix is

$$\begin{aligned}
\text{M/D: } &\Big((2^d+1)(2d+1)\Big)s^3 + \Big((2d+1)(2k+1) + 2^d(6dk+2d+2k+1)\Big)s^2 \\
&+ \Big(k(2d+1) + 2^d(2dk+k)\Big)s + k(2d+1) + 2^{d+1}dk \\
\text{A/S: } &\Big((2^d+1)(2d+1)\Big)s^3 + \Big((2d+1)(k-2) + 2^d(3dk-4d+k-2)\Big)s^2 \\
&+ \Big(k(2d+1) + 2^d(2dk-d+k)\Big)s - k(2d+1) - 2^d(2d-k)
\end{aligned} \tag{43}$$

where $s = (n+d)!/n!d!$, and k is the number of shape functions covering a given node.

In contrast to MSNNI, using NSNI, the functions $\Psi_I(\mathbf{x})$, $\nabla\Psi_I(\mathbf{x})$, $\Psi^{\nabla}_{Ii}(x)$, and $\nabla\Psi^{\nabla}_{Ii}(x)$ need only be calculated once for each node, which gives substantially less CPU burden. Consequently, the total operation count for evaluating all shape functions for one node necessary for forming the stiffness matrix, body force, and mass matrix is

$$\begin{aligned}
\text{M/D: } &(2d+1)s^3 + (d^2+6dk+3d+2k+1)s^2 + (2d^2k+k+3dk)s \\
&+ dk(d+3) + k \\
\text{A/S: } &(2d+1)s^3 + (d^2+3dk-3d+k-2)s^2 + (2d^2k+3dk-d^2+k-2d)s \\
&- dk(d+1) + 2d - k.
\end{aligned} \tag{44}$$

For uniform discretizations with $n = 1$, the cost ratios of (43) versus (44) are presented in Tables I and II, for two and three dimensions, respectively, for various normalized support sizes. It can be seen that in both cases, the cost savings in constructing the discrete equations is significant. For two dimensions, 5–7 times cost savings can be observed, and for three dimensions, a cost savings of an order of magnitude can be achieved (12–18 times). The details of the complexity analysis for the two methods can be found in Appendix B, following [44].

Table I. Operation count ratios of MSNNI versus NSNI in 2-D.

	$1 < a < 1.5$	$1.5 \leqslant a < 2.5$	$2.5 \leqslant a < 3.5$
M/D ratio	7.39	6.38	5.77
A/S ratio	7.21	6.10	5.48

Table II. Operation count ratios of MSNNI versus NSNI in 3-D.

	$1 < a < 1.5$	$1.5 \leqslant a < 2.5$	$2.5 \leqslant a < 3.5$
M/D ratio	20.30	15.69	13.22
A/S ratio	18.88	14.44	12.14

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

3.4. Variationally consistent naturally stabilized nodal integration

Variational consistency plays an important role in quadrature for meshfree methods for attaining optimal convergence in an efficient manner. To achieve high accuracy and optimal convergence using conventional integration methods such as Gaussian integration, very high order quadrature is required, which is prohibitively expensive. A way to circumvent this problem is to satisfy the variational consistency conditions, which are the requirements to satisfy an n^{th} order patch test.

The two requirements for n^{th} order variational consistency are n^{th} order reproducing conditions of the trial functions, and satisfaction of the n^{th} order integration constraints by the test functions [17]. It is possible to construct a set of trial functions that satisfy the reproducing conditions using shape functions such as those in (1) and a separate set of test functions that satisfy the integration constraints.

Let us reexamine this condition for the proposed stabilization. To arrive at (14), a problem with a linear solution is considered for examining linear Galerkin exactness with quadrature, leading to nodal coefficients

$$\begin{aligned} u_{I1} &= c_{10} + c_{11}x_I + c_{12}y_I + c_{13}z_I \\ u_{I2} &= c_{20} + c_{21}x_I + c_{22}y_I + c_{23}z_I \\ u_{I3} &= c_{30} + c_{31}x_I + c_{32}y_I + c_{33}z_I \,. \end{aligned} \tag{45}$$

Utilizing the gradient reproducing conditions (25), the terms $\hat{\mathbf{u}}^h_x$, $\hat{\mathbf{u}}^h_y$, and $\hat{\mathbf{u}}^h_z$ in (33) are constant, thus $\boldsymbol{\varepsilon}_L\left(\hat{\mathbf{u}}^h_x\right) = \boldsymbol{\varepsilon}_L\left(\hat{\mathbf{u}}^h_y\right) = \varepsilon_L\left(\hat{\mathbf{u}}^h_z\right) = \mathbf{0}$ and the additional stabilization (39) vanishes, that is,

$$a_{ND}\langle \mathbf{v}^h, \mathbf{u}^h\rangle = a_D\langle \mathbf{v}^h, \mathbf{u}^h\rangle. \tag{46}$$

Using the above, it is easily shown following [17] that the condition for first order variational consistency for NSNI is the constraint (14).

Satisfaction of the constraint can be straightforwardly accomplished by introducing assumed gradients for the test functions as in [17], while leaving the additional terms in (39) unmodified:

$$\bar{\Psi}_{I,i}(\mathbf{x}) = \Psi_{I,i}(\mathbf{x}) + R_I(\mathbf{x})\xi_{Ii} \tag{47}$$

where $(\cdot)_{,i} \equiv \partial(\cdot)/\partial x_i$ and ξ_{Ii} is a constant coefficient and

$$R_I(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \text{ supp } (\Psi_I(\mathbf{x})) \\ 0 & \text{if } \mathbf{x} \notin \text{ supp } (\Psi_I(\mathbf{x})) \end{cases} . \tag{48}$$

Substitution of (47) into (14) yields the coefficients

$$\xi_{Ii} = \left(\int_{\partial\Omega}^{\wedge} \Psi_I n_i \mathrm{d}\Gamma - \int_{\Omega}^{\wedge} \Psi_{I,i}\mathrm{d}\Omega \right) \Big/ \int_{\Omega}^{\wedge} R_I \mathrm{d}\Omega \,. \tag{49}$$

Note that no particular type of numerical integration has been specified in the introduction of the test function gradients. The coefficients in (49) make any numerical integration first order variationally consistent, and thus applies to Gaussian integration, DNI, SNNI, and other variationally inconsistent methods. For SCNI which satisfies (14), the formulation leads to coefficients ξ_{Ii} of zero, and no correction is introduced. The NSNI approach with the variationally consistent correction (47)–(49) is termed Variationally consistent naturally stabilized nodal integration (VC-NSNI) herein.

The fourth lowest eigenvalue modes are shown in Figure 6 for variationally consistent DNI and SNNI (denoted herein with the prefix 'VC-') and their stabilized counterparts VC-NSNI and VC-MSNNI, respectively, using the previously employed RK discretization. It can be seen that employment of variationally consistent test function gradients alone cannot preclude the oscillatory modes, because of the instability resulting from the choice of trial functions, which agrees with the discussion in Section 2.4. It can also be seen in Figure 6 that for VC-MSNNI and VC-NSNI, the introduction of test function gradients (47) in the variationally consistent correction does not degrade the stability of the already stable methods MSNNI and NSNI, and have nearly the exact same eigenvalues for this mode. Lastly, the mode shape for VC-MSNNI and VC-NSNI both agree well with the shape for fully integrated FEM.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

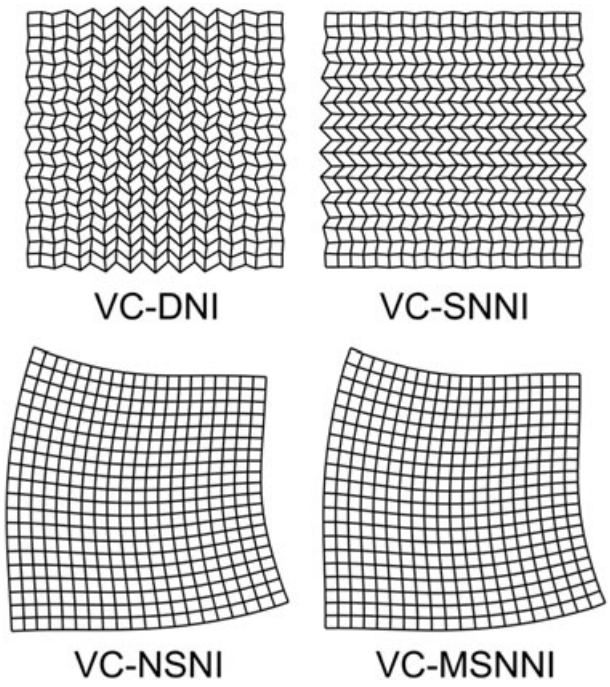

Figure 6. Fourth lowest eigenvalue mode of the stiffness matrix for variationally consistent integration methods.

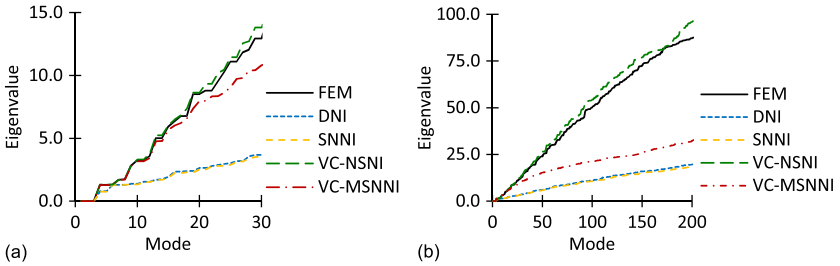

Figure 7. Eigenvalues of stiffness matrices: (a) 30 lowest modes and (b) 200 lowest modes.

The two stable methods VC-MSNNI and VC-NSNI differ however when the spectrum of eigenvalues is examined. The first lowest 30 modes agree reasonably well with fully integrated FEM for both cases, as seen in Figure 7(a), but when the first lowest 200 modes are examined, VC-NSNI shows much better agreement, as seen in Figure 7(b).

Because several techniques are employed to arrive at the present method VC-NSNI, matrix forms and implementation details are given in Appendix C.

Remark

Whenever a Petrov-Galerkin formulation is employed, the stability of the discrete system requires careful investigation. For the variationally consistent formulation given by (47)–(49), the stability requires the coercivity of the bilinear form, which hinges on the coefficients of the correction being small [17]. This, in turn, requires adequate quadrature accuracy to yield a small enough residual of the integration constraint. In reference [17], it was demonstrated that several nodal integration

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

schemes, including DNI, are adequate for effective corrections when reproducing kernel approximation functions with linear bases are used. If the bilinear form is coercive, an important implication for both static and dynamic problems is that the system matrix will also be positive definite [45].

4. NUMERICAL EXAMPLES

4.1. Patch test

In order to verify the variational consistency of VC-NSNI, consider a linear patch test for the Poisson problem in two dimensions with the solution $u = x + 2y$. The corresponding boundary value problem is

$$\begin{array}{lll} \nabla^2 u = 0 & \text{in} & \Omega \\ \nabla u \cdot \mathbf{n} = n_1 + 2n_2 & \text{on} & \partial\Omega_h \\ u = x + 2y & \text{on} & \partial\Omega_g \end{array} \tag{50}$$

where $\Omega : (-1,1) \times (-1,1)$, $\partial\Omega_h : -1 \leqslant x \leqslant 1,\ y = 1;\ x = 1,\ -1 \leqslant y \leqslant 1$, and $\partial\Omega_g = \partial\Omega \backslash \partial\Omega_h$. The problem is solved with NSNI and VC-NSNI to demonstrate that first order variational consistency is violated and satisfied for these methods, respectively. A non-uniform discretization shown in Figure 8(a) is considered, with linear RKPM using cubic B-spline kernels with a normalized support of 1.5. The L^2 norms and H^1 semi-norms of the solution error are shown in Table III, where it can be seen that the linear patch test is passed for the variationally consistent proposed stabilized nodal integration. It is also seen that NSNI does not satisfy variational consistency. The solutions for the two methods are shown in Figure 8(b) and Figure 8(c), where NSNI exhibits visible solution error.

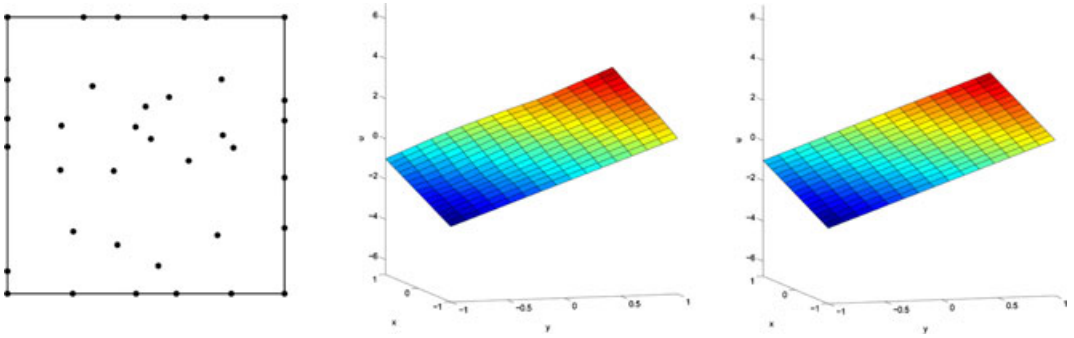

Figure 8. Linear patch test: (a) reproducing kernel particle method discretization, (b) solution by naturally stabilized nodal integration, and (c) solution by variationally consistent naturally stabilized nodal integration (exact).

Table III. Norms of error in the linear patch test.

	L^2 norm	H^1 semi-norm
NSNI	1.67E-01	4.92E-01
VC-NSNI	2.02E-14	6.05E-14

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

4.2. *Convergence in Poisson equation*

To examine the convergence and efficiency of the various quadrature formulations discussed herein, consider the following Poisson problem:

$$\begin{aligned} \nabla^2 u &= \sin(\pi x)\sin(\pi y) \quad \text{in} \quad \Omega \\ u &= 0 \qquad\qquad\qquad\quad \text{on} \quad \partial\Omega \end{aligned} \tag{51}$$

where $\Omega : (-1,1)\times(-1,1)$. The solution to the problem is

$$u = -\frac{1}{2\pi^2}\sin(\pi x)\sin(\pi y). \tag{52}$$

The domain is discretized non-uniformly and refined uniformly as shown in Figure 9, using 121, 441, 1681, and 6561 nodes. Linear RKPM using cubic B-spline kernels with a normalized support of 2.0 is employed for the study. Convergence of the error is shown in Figure 10 for the various

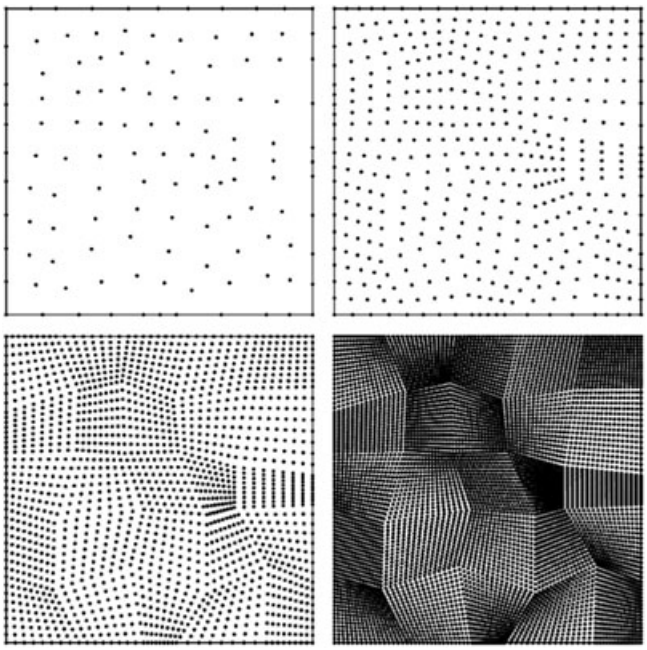

Figure 9. Discretizations used for convergence in Poisson problem.

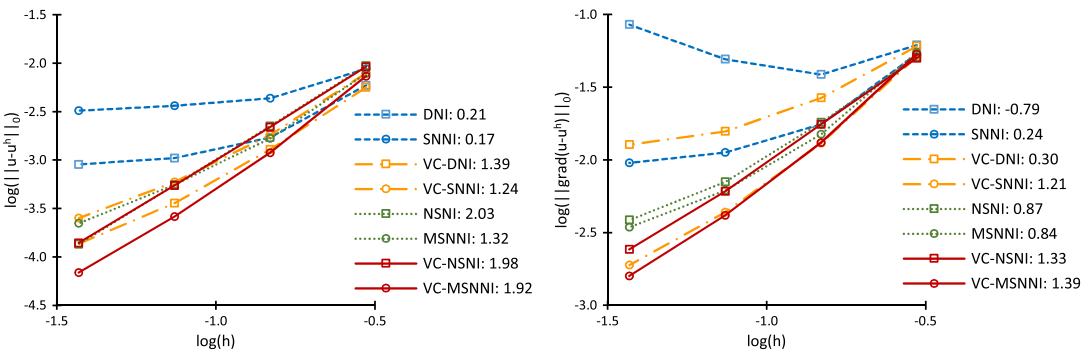

Figure 10. Convergence of nodal integration methods (rates indicated in legend).

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

methods discussed. Here it can be seen that only the stabilized and variationally consistent methods (VC-NSNI and VC-MSNNI) give the optimal rate of convergence in both norms. VC integration alone does not give optimal convergence because of the unstable modes that are not kept in check. It can also be seen that stabilization alone also does not give optimal rates because of violation of variational consistency. Thus, it appears that both stabilization and variational consistency are necessary to ensure optimal convergence with nodal integration.

Comparing the two stable and convergent methods, over an order of magnitude of CPU time (about 16 times for the finest) is saved using VC-NSNI versus VC-MSNNI, as seen in Figure 11. Here, the CPU time includes only shape function construction and stiffness matrix assembly. From Table I, for the kernel support employed, it is expected that the shape function cost should be decreased by roughly six times. The difference may be caused by the increased assembly time because of the increased number of neighbors in MSNNI, which employs smoothed integration or differences caused by implementation. Lastly, from Figure 11, it can be seen that the increase in CPU time for constructing variationally consistent test functions for all methods is marginal.

The CPU time versus error is shown in Figure 12 for the various methods. Lines that are closer to the origin are more effective because they give the least error and CPU time. Because VC-NSNI and VC-MSNNI are the only stable methods that converge optimally, the comparison of these is the primary case of interest. From the figure, it is clear that VC-NSNI is the more effective method for both norms, and that for a given level of error, roughly an order of magnitude less CPU time is required compared to VC-MSNNI. Moreover, it is arguable that VC-NSNI is the most effective method of all, because, while VC-DNI and NSNI provide similar effectiveness in the L^2 norm, for the H^1 semi-norm, VC-DNI is far worse, and NSNI has reduced effectiveness as the level of refinement increases.

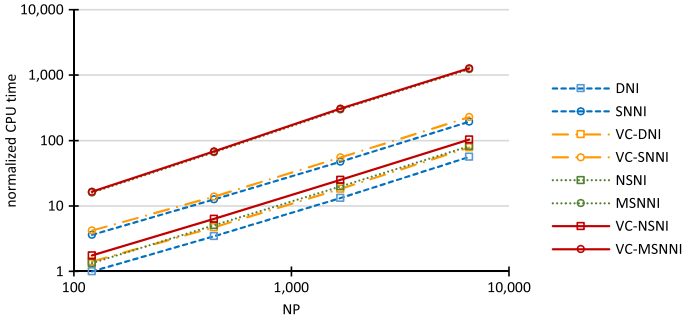

Figure 11. Scaling of CPU time with increase in degrees of freedom.

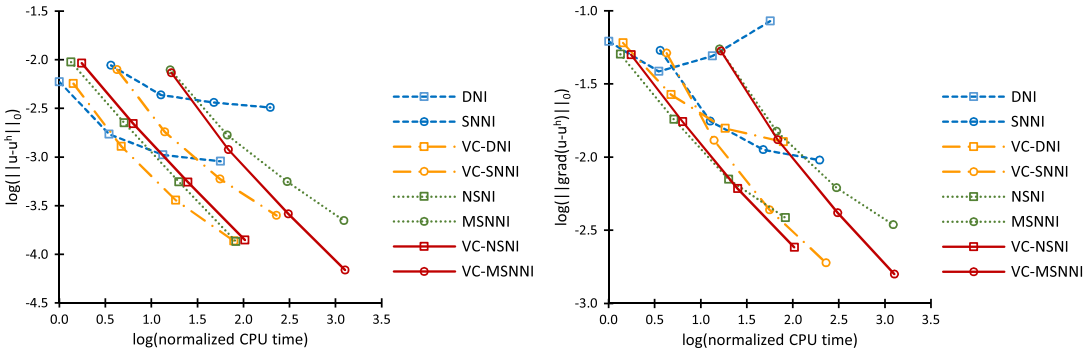

Figure 12. Effectiveness of nodal integration methods.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

4.3. Tension test

Consider the necking of a steel bar undergoing a large axial displacement. As shown in Figure 13, the model of the bar is slightly tapered in the middle. The dimensions of the bar indicated in the figure are L_0 =5.3334 cm, a_0 =0.6298 cm, and a_1 =0.6413 cm. This experiment has been carried out in [46] with data obtained for the force-displacement relationship and necking at the center of the bar. The material is modeled with J_2 plasticity and isotropic hardening. The elastic properties of the bar are Young's modulus E =206.9 GPa and Poisson's ratio ν =0.29 [46], and the yield stress is taken as

$$K(\bar{e}_p) = \sigma_y^0 + \alpha \bar{e}^p + \left(\sigma_y^\infty - \sigma_y^0\right)\left(1 - e^{-\beta \bar{e}^p}\right) \tag{53}$$

where $\bar{e}$ is the equivalent plastic strain, σ_y^0 =0.45 GPa, α =0.12924 GPa, σ_y^∞ =0.715 GPa , and β =16.93.

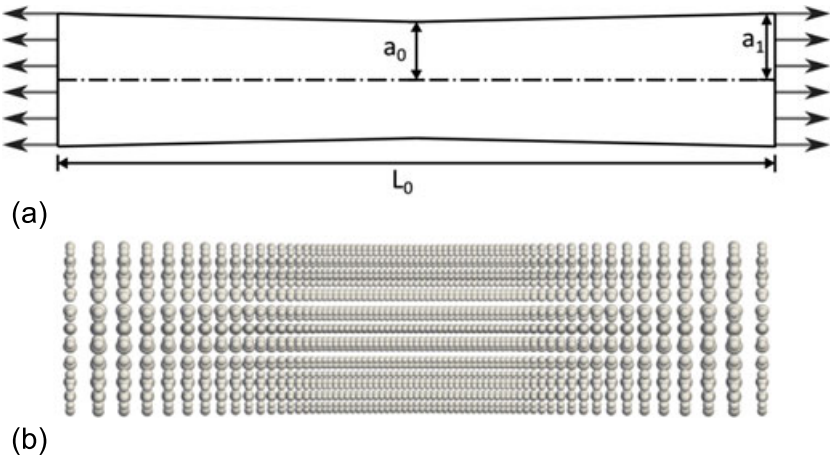

Figure 13. Tension test: (a) depiction of problem with force F and dimensions indicated and (b) RKPM discretization.

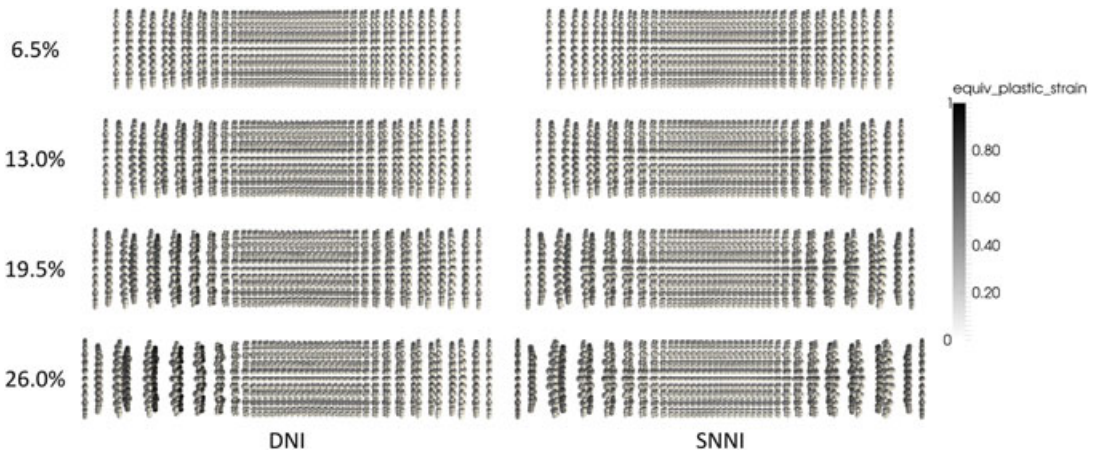

Figure 14. Tension test: progressive deformation with nodal integration at various levels of elongation with equivalent plastic strain values indicated in the legend.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

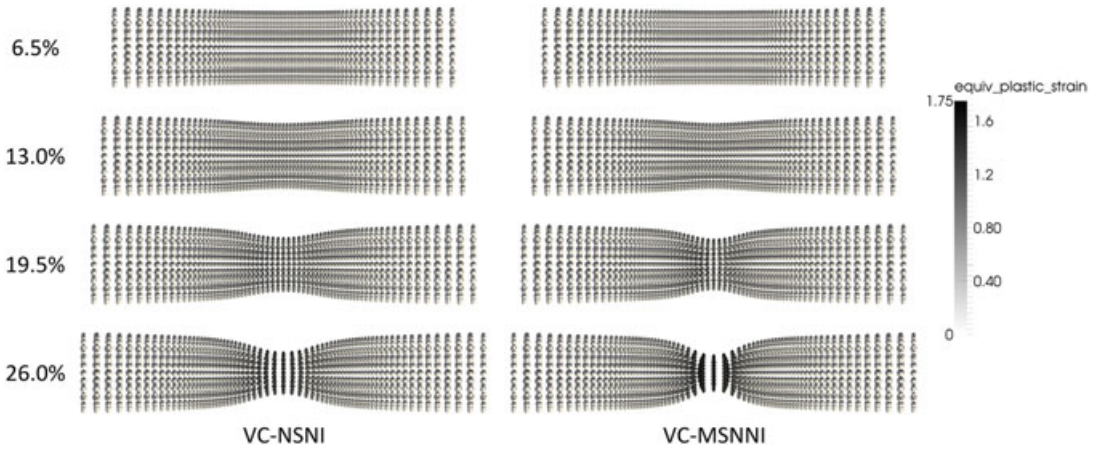

Figure 15. Tension test: progressive deformation with stabilized nodal integration at various levels of elongation with equivalent plastic strain values indicated in the legend.

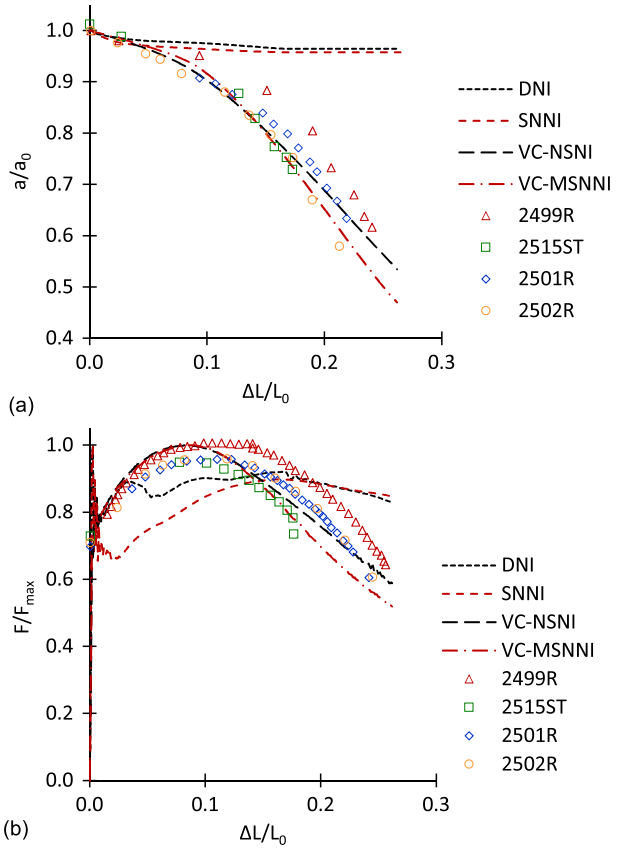

Figure 16. Comparison of various methods with experimental data from [46] in the tension test: (a) reduction of radius in the necking region and (b) load displacement curve.

The semi-Lagrangian formulation [40, 41] is considered where shape functions are continually re-constructed due to employment of semi-Lagrangian kernels. This type of formulation is a primary case of interest for this paper aside from linear problems, because the re-construction of shape functions constitutes a large portion of overall CPU time. Linear basis with a normalized support of 1.6

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

is employed with a quartic B-spline kernel in the RK approximation, and the bar is discretized using 6693 nodes. Based on the previous examples, four methods of interest are selected for comparison: SNNI, DNI, VC-MSNNI, and VC-NSNI.

The progressive deformation of the bar for the unstabilized nodal integration methods DNI and SNNI are shown in Figure 14 at various levels of elongation. The spurious modes of alternating displacement are clearly seen near the ends of the bar, and instead of the strain localizing in the center of the bar, the strain localizes at the location of these instabilities. In contrast, these modes are completely absent in the solutions obtained from VC-NSNI and VC-MSNNI as shown in Figure 15, where strain is seen to localize at the correct location, at the center of the bar. Because of the spurious modes in the unstabilized nodal integrations, they are seen to give very poor results when compared with the experimental data, as shown in Figure 16, while the stabilized nodal integrations VC-NSNI and VC-MSNNI agree well with the experimental data.

4.4. Taylor bar impact

The Taylor bar impact problem [47] was first performed by Wilkins and Guinan in [48] and subsequently by many others. An aluminum bar with an initial height and radius of 2.346 cm and 0.391 cm, respectively, impacts a rigid wall with an initial velocity of 373.0 m/s. For the aluminum material, J_2 plasticity with isotropic hardening is considered with material properties Young's modulus $E = 78.2$ GPa, Poisson's ratio $\nu = 0.30$, and density $\rho = 2700$ kg/m^3, and the yield stress is taken as

$$K(\bar{e}_p) = \sigma_Y(1 + 125\bar{e}_p)^{0.1} \tag{54}$$

where $\sigma_Y = 0.29$ GPa.

The semi-Lagrangian formulation is again considered, with linear basis and quartic B-spline kernel functions with a normalized support of 2.8, and 29,637 nodes discretize the bar. The wall is also modeled and is considered frictionless, and a kernel contact algorithm [41] is employed for the bar-wall interaction. The four integration methods of interest used in the previous example are again employed.

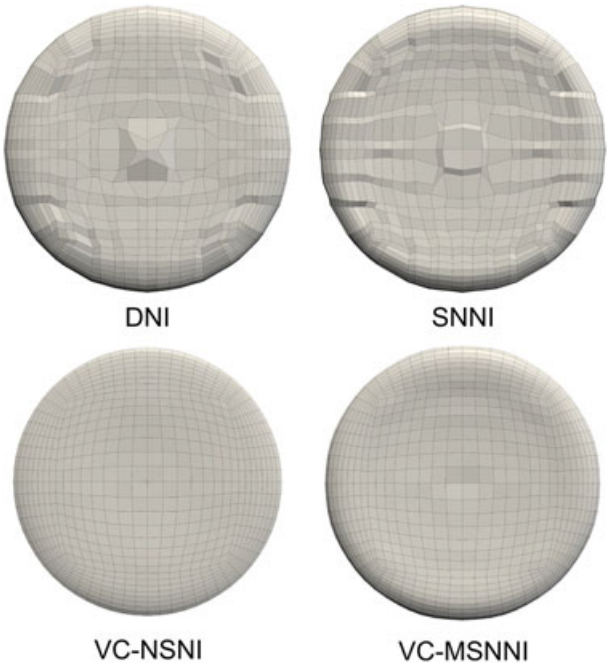

Figure 17. Final deformation on the face of the Taylor bar for various nodal integration methods.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

Table IV. Dimensions of deformed Taylor bar and CPU times.

	Radius (cm)	Height (cm)	Normalized CPU
SNNI	0.839	1.649	2.46
DNI	0.838	1.660	0.70
VC-MSNNI	0.801	1.649	8.15
VC-NSNI	0.775	1.651	1.00
RKPM [4]	0.827	1.645	-
Particle in cell [49]	0.78	1.65	-
Finite element [48]	0.742	1.652	-
Experimental [48]	-	1.651	-

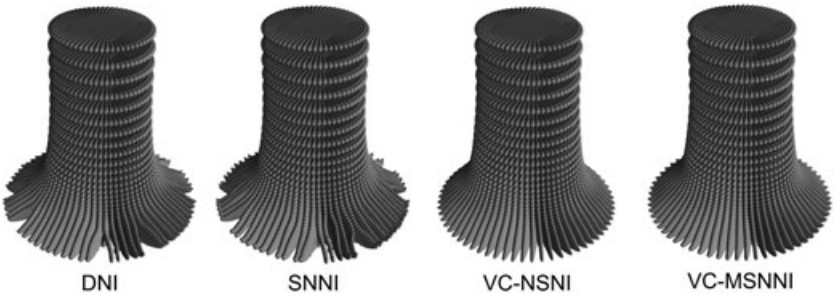

Figure 18. Final deformation of the Taylor bar for various nodal integration methods.

The deformed shape of the face of the bar is shown in Figure 17 with material deformation plotted using the connectivity of the preprocessing mesh used to generate the RKPM discretization. Here, the difference in solutions is quite dramatic; DNI and SNNI both clearly show the spurious oscillatory modes in the solution, while the stabilized methods do not.

The deformed height and radius of the bar, and normalized CPU times for the four methods are shown in Table IV, along with several reference solutions and the experimental deformed height. First, it is seen that a speed-up factor of over eight for VC-NSNI versus VC-MSNNI is observed. The reason the range of speed up shown in Table II is not observed is because of the additional calculations performed for both methods, for example, node neighbor search and stress update.

For the deformed dimensions in Table IV, it can be seen that DNI and SNNI predict a very large radius compared with the reference solutions, likely because of the very little resistance to the oscillatory mode of deformation. Another explanation is the fact that the Taylor bar actually splits because of these modes as seen in Figure 18, while VC-NSNI and VC-MSNNI give contiguous solutions. For the deformed height of the bar, all methods give reasonable heights compared with the reference solutions except for DNI, and VC-NSNI is closest to the experimental data.

5. CONCLUSION

In this paper, a new stabilized nodal integration method is introduced for meshfree methods under the RKPM framework. The method employs an implicit gradient expansion of the strains at each node that naturally results in a stabilized nodal integration and is devoid of tunable parameters. Because of the implicit construction of the gradient, high efficiency is attained, and the complexity of the method is similar to direct nodal integration.

Variationally consistent integration was also introduced for the proposed stabilization. The combined method is able to pass the patch test, and studies demonstrated optimal convergence of the

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

method. One interesting result of the convergence study was that both stabilization and variationally consistent integration were shown to be necessary to attain optimal convergence for nodal integration. In addition, the eigenvalue analysis showed that stability was not sacrificed by the introduction of variationally consistent test functions, and CPU time studies showed that efficiency was not strongly affected by introducing variationally consistent integration. The proposed method was also shown in several examples to be effective in providing stable solutions for large deformation problems.

The VC-MSNNI method was the primary basis for comparison because it offers both stability and optimal convergence in nodal integration, and is also suitable for problems with material separation. Compared with this method, the complexity was shown to be between 5–20 times lower for the proposed VC-NSNI method depending on support and spatial dimension. In addition, the method was demonstrated to provide superior stability over VC-MSNNI in an eigenvalue analysis.

The method presented herein shows good performance and high effectiveness, and can be used to significantly decrease CPU time in problems where a large portion of computation time is spent on shape function construction, for example, in linear problems, or when semi-Lagrangian kernels are employed in nonlinear problems. In addition, for Lagrangian formulations, the method is expected to provide similar efficiency because state and field variables are only needed at nodes.

APPENDIX A: NONLINEAR IMPLEMENTATION OF NATURALLY STABILIZED NODAL INTEGRATION

For objective integration of stress rates, we consider the procedure given in [50]. From time n to time $n+1$ with generalized displacement increment $\Delta\mathbf{d}^n = \mathbf{d}^{n+1} - \mathbf{d}^n$, for the calculation of stress $\boldsymbol{\sigma}_I^{n+1}$ at node I the trial stress $\boldsymbol{\sigma}_{I,\text{trial}}^{n+1}$ is first calculated as

$$\begin{aligned}
\boldsymbol{\sigma}_{I,\text{trial}}^{n+1} &= \mathbf{Q}_I \boldsymbol{\sigma}_I^n \mathbf{Q}_I^T + \mathbf{D}^{\text{e}} \Delta\boldsymbol{\varepsilon}_I \\
\mathbf{Q}_I &= \mathbf{I} + \left(\mathbf{I} - \tfrac{1}{2}\boldsymbol{\omega}_I\right)^{-1} \boldsymbol{\omega}_I \\
\boldsymbol{\omega}_\mathbf{I} &= \tfrac{1}{2}\left(\mathbf{G}_I - \mathbf{G}_I^T\right),
\end{aligned} \tag{A.1}$$

where

$$\begin{aligned}
\mathbf{G}_I &= \mathbf{L}_I \left(\mathbf{I} + \tfrac{1}{2}\mathbf{L}_I\right)^{-1} \\
(\mathbf{L}_I)_{ij} &= \sum_{J=1}^{NP} \Psi_{J,j}(\mathbf{x}_I) \Delta d_{Ji} \\
\boldsymbol{\Delta\varepsilon}_I &= \tfrac{1}{2}\left(\mathbf{G}_I + \mathbf{G}_I^T\right),
\end{aligned} \tag{A.2}$$

and $\mathbf{D}^{\text{e}}$ is the elastic co-rotational material response tensor. The Cauchy stress $\boldsymbol{\sigma}_I^{n+1}$ is then computed, by, for example for plasticity, return mapping.

For stabilization, denote the stress gradient at the nodes as $\boldsymbol{\sigma}_{I,i}$ with increment $\Delta\boldsymbol{\sigma}_{I,i}$. The stress gradient update is performed as follows:

$$\begin{aligned}
\boldsymbol{\sigma}_{I,i}^{n+1} &= \mathbf{Q}_I \boldsymbol{\sigma}_{I,i}^n \mathbf{Q}_I^T + \boldsymbol{\Delta\sigma}_{I,i} \\
\boldsymbol{\Delta\sigma}_{I,i} &= \mathbf{D}_I^{\text{ep}} \boldsymbol{\Delta\varepsilon}_{I,i} \\
\boldsymbol{\Delta\varepsilon}_{I,i} &= \frac{1}{2}\left(\mathbf{G}_{I,i} + \mathbf{G}_{I,i}^T\right),
\end{aligned} \tag{A.3}$$

where $\mathbf{D}_I^{\text{ep}}$ is the consistent tangent moduli computed at node I and

$$(L_{I,i})_{jk} = \sum_{J=1}^{NP} \Psi_{Ji,k}^{\nabla}(\mathbf{x}_I) \Delta d_{Jj}. \tag{A.4}$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

APPENDIX B: COMPLEXITY ANALYSIS

B.1. Operation counts for elementary operations

The bulk of the cost in forming reproducing kernel (RK) shape functions and their derivatives are the matrix operations involved. The size of matrices in forming RK shape functions are the size of the basis vector s. Operation counts for elementary operations involved in terms of multiplication and division, and addition and subtraction are shown in Table B.1.

B.2. Operation counts for reproducing kernel shape functions

For complexity analysis, it is convenient to express the RK approximation function $\Psi_I(\mathbf{x})$ as the product of the kernel function $\Phi_a(\mathbf{x}-\mathbf{x}_I)$ and a correction function $C(\mathbf{x};\mathbf{x}-\mathbf{x}_I)$:

$$\Psi_I(\mathbf{x}) = C(\mathbf{x};\mathbf{x}-\mathbf{x}_I)\Phi_a(\mathbf{x}-\mathbf{x}_I) \tag{B.1}$$

where

$$C(\mathbf{x};\mathbf{x}-\mathbf{x}_I) = \mathbf{b}(\mathbf{x})\mathbf{H}(\mathbf{x}-\mathbf{x}_I) \tag{B.2}$$

$$\mathbf{b}(\mathbf{x}) = \mathbf{H}^T(\mathbf{0})\mathbf{M}^{-1}(\mathbf{x}) \tag{B.3}$$

$$\mathbf{M}(\mathbf{x}) = \sum_{I=1}^{NP} \mathbf{H}(\mathbf{x}-\mathbf{x}_I)\mathbf{H}^T(\mathbf{x}-\mathbf{x}_I)\Phi_a(\mathbf{x}-\mathbf{x}_I). \tag{B.4}$$

The derivatives of (B.1) with respect to x_j can be taken by product rule:

$$\Psi_{I,j}(\mathbf{x}) = C_{,j}(\mathbf{x};\mathbf{x}-\mathbf{x}_I)\Phi_a(\mathbf{x}-\mathbf{x}_I) + C(\mathbf{x};\mathbf{x}-\mathbf{x}_I)\Phi_{a,j}(\mathbf{x}-\mathbf{x}_I) \tag{B.5}$$

where

$$C_{,j}(\mathbf{x};\mathbf{x}-\mathbf{x}_I) = \mathbf{b}_{,j}(\mathbf{x})\mathbf{H}(\mathbf{x}-\mathbf{x}_I) + \mathbf{b}(\mathbf{x})\mathbf{H}_{,j}(\mathbf{x}-\mathbf{x}_I) \tag{B.6}$$

$$\mathbf{b}_{,j}(\mathbf{x}) = \mathbf{H}^T(\mathbf{0})\mathbf{M}^{-1}_{,j}(\mathbf{x}). \tag{B.7}$$

The term $\mathbf{M}^{-1}_{,j}(\mathbf{x})$ can be found by differentiating the identity $\mathbf{M}^{-1}(\mathbf{x})\mathbf{M}(\mathbf{x}) = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix:

$$\mathbf{M}^{-1}_{,j}(\mathbf{x}) = -\mathbf{M}^{-1}(\mathbf{x})\mathbf{M}_{,j}(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x}). \tag{B.8}$$

Table B.1. Operation counts for elementary matrix operations.

Operation	M/D	A/S
Inner product: $\mathbf{x}\cdot\mathbf{y}$	s	$s-1$
Outer product: $\mathbf{x}\otimes\mathbf{y}$	s^2	0
Matrix-scalar multiplication: $\gamma\mathbf{A}$	s^2	0
Matrix-vector multiplication: $\mathbf{Ax}$	s^2	s^2-s
Matrix-matrix multiplication: $\mathbf{AB}$	s^3	s^3-s^2

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

In the following, let I denote a shape function index associated with the k shape functions covering a nodal integration point $\mathbf{x}_L$. The number k is called the overlapping number.

The complexity of constructing the set of shape functions $\{\Psi_I(\mathbf{x}_L)\}_{I=1}^{NP}$ is

$$\begin{array}{l}\text{M/D: } s^3 + (2k+1)s^2 + ks + k \\ \text{A/S: } s^3 + (k-2)s^2 + k(s-1).\end{array} \tag{B.9}$$

The complexity of constructing the set of shape functions and their gradients $\{\Psi_I(\mathbf{x}_L), \nabla\Psi_I(\mathbf{x}_L)\}_{I=1}^{NP}$ is

$$\begin{array}{l}\text{M/D: } (2d+1)s^3 + (6dk+2d+2k+1)s^2 + (2dk+k)s + 2dk + k \\ \text{A/S: } (2d+1)s^3 + (3dk-4d+k-2)s^2 + (2dk-d+k)s + 2d - k.\end{array} \tag{B.10}$$

Note that the above takes into consideration repeated computations for shape functions and derivatives for multiple dimensions, as well as precise consideration of overlap numbers and provides a slightly sharper estimate than given in [44].

B.3. Complexity of modified stabilized non-conforming nodal integration and naturally stabilized nodal integration

Modified stabilized non-conforming nodal integration (MSNNI) with box smoothing cells with boundaries integrated with single point integration gives $2d$ smoothing evaluation points and associated stabilization sub-cells. For a node located at $\mathbf{x}_L$, let $\mathbf{x}_L^{\text{Sm}}$ denote an associated smoothing zone integration point and $\mathbf{x}_L^{\text{St}}$ an associated stabilization point. For MSNNI, the overlapping number k is considered as the total number of shape functions covering any of the various necessary evaluation points, because in practice, often only a single list of neighbors is constructed for all the shape function evaluations because of the node neighbor search cost. All terms and their operation counts necessary for MSNNI are shown in Table B.2.

The complexity of constructing the set of shape functions at a node $\{\Psi_I(\mathbf{x}_L)\}_{I=1}^{NP}$ and associated $2d$ smoothing points $\left\{\Psi_I\left(\mathbf{x}_L^{\text{Sm}}\right)\right\}_{I=1}^{NP}$ for MSNNI is thus

$$\begin{array}{l}\text{M/D: } (2d+1)\left(s^3 + (2k+1)s^2 + ks + k\right) \\ \text{A/S: } (2d+1)\left(s^3 + (k-2)s^2 + ks - k\right).\end{array} \tag{B.11}$$

The complexity of constructing the set of shape functions gradients $\left\{\nabla\Psi_I\left(\mathbf{x}_L^{\text{St}}\right)\right\}_{I=1}^{NP}$ at each of the stabilization points $\mathbf{x}_L^{\text{St}}$ in MSNNI is

Table B.2. Operation counts for matrix operations for modified stabilized non-conforming nodal integration.

Forming	M/D operations	A/S operations	Computed for all
$\mathbf{M}(\mathbf{x})$	$2ks^2$	$(k-1)s^2$	$\mathbf{x}_L, \mathbf{x}_L^{\text{Sm}}, \mathbf{x}_L^{\text{St}}$
$\mathbf{M}_{,j}(\mathbf{x})$	$6ks^2$	$3(k-1)s^2 + 2$	$\mathbf{x}_L^{\text{St}}, j$
$\mathbf{M}^{-1}(\mathbf{x})$	s^3	$s^3 - 2s^2 + s$	$\mathbf{x}_L, \mathbf{x}_L^{\text{Sm}}, \mathbf{x}_L^{\text{St}}$
$\mathbf{M}_{,j}^{-1}(\mathbf{x})$	$2s^3 + s^2$	$2(s^3 - s^2)$	$\mathbf{x}_L^{\text{St}}, j$
$\mathbf{b}(\mathbf{x})$	s^2	$s^2 - s$	$\mathbf{x}_L, \mathbf{x}_L^{\text{Sm}}, \mathbf{x}_L^{\text{St}}$
$\mathbf{b}_{,j}(\mathbf{x})$	s^2	$s^2 - s$	$\mathbf{x}_L^{\text{St}}, j$
$C(\mathbf{x}; \mathbf{x} - \mathbf{x}_I)$	s	$s-1$	$\mathbf{x}_L, \mathbf{x}_L^{\text{Sm}}, \mathbf{x}_L^{\text{St}}, I$
$C_{,j}(\mathbf{x}; \mathbf{x} - \mathbf{x}_I)$	$2s$	$2s-1$	$\mathbf{x}_L^{\text{St}}, j, I$
$\Psi_I(\mathbf{x})$	1	0	$\mathbf{x}_L, \mathbf{x}_L^{\text{Sm}}, I$
$\Psi_{I,j}(\mathbf{x})$	2	1	$\mathbf{x}_L^{\text{St}}, j, I$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

Table B.3. Operation counts for matrix operations in naturally stabilized nodal integration.

Forming	M/D operations	A/S operations	Computed for all
$\mathbf{M}(\mathbf{x})$	$2ks^2$	$(k-1)s^2$	$\mathbf{x}_L$
$\mathbf{M}_{,j}(\mathbf{x})$	$6ks^2$	$3(k-1)s^2+2$	$\mathbf{x}_L, j$
$\mathbf{M}^{-1}(\mathbf{x})$	s^3	s^3-2s^2+s	$\mathbf{x}_L$
$\mathbf{M}^{-1}_{,j}(\mathbf{x})$	$2s^3+s^2$	$2\left(s^3-s^2\right)$	$\mathbf{x}_L, j$
$\mathbf{b}(\mathbf{x})$	s^2	s^2-s	$\mathbf{x}_L$
$\mathbf{b}^i(\mathbf{x})$	s^2	s^2-s	$\mathbf{x}_L, i$
$\mathbf{b}_{,j}(\mathbf{x})$	s^2	s^2-s	$\mathbf{x}_L, j$
$\mathbf{b}^i_{,j}(\mathbf{x})$	s^2	s^2-s	$\mathbf{x}_L, i, j$
$C(\mathbf{x};\mathbf{x}-\mathbf{x}_I)$	s	$s-1$	$\mathbf{x}_L, I$
$C^i(\mathbf{x};\mathbf{x}-\mathbf{x}_I)$	s	$s-1$	$\mathbf{x}_L, i, I$
$C_{,j}(\mathbf{x};\mathbf{x}-\mathbf{x}_I)$	$2s$	$2s-1$	$\mathbf{x}_L, j, I$
$C^i_{,j}(\mathbf{x};\mathbf{x}-\mathbf{x}_I)$	$2s$	$2s-1$	$\mathbf{x}_L, i, j, I$
$\Psi_I(\mathbf{x})$	1	0	$\mathbf{x}_L, I$
$\Psi_{I,j}(\mathbf{x})$	2	1	$\mathbf{x}_L, j, I$
$\Psi^{\nabla}_{Ii}(\mathbf{x})$	1	0	$\mathbf{x}_L, i, I$
$\Psi^{\nabla}_{Ii,j}(\mathbf{x})$	2	1	$\mathbf{x}_L, i, j, I$

$$\begin{array}{ll}\text{M/D:} & 2^d\left((2d+1)s^3+(6dk+2d+2k+1)s^2+(2dk+k)s+2dk\right)\\ \text{A/S:} & 2^d\left((2d+1)s^3+(3dk-4d+k-2)s^2+(2dk-d+k)s+2d-k\right).\end{array} \tag{B.12}$$

Consequently, the total complexity of evaluating all shape functions for a nodal integration point $\mathbf{x}_L$ is given by (43).

For NSNI, the construction of the function $\Psi^{\nabla}_{Ii}(x)$ can also be expressed as a the product of the kernel $\Phi_a(\mathbf{x}-\mathbf{x}_I)$ and a correction function $C^i(\mathbf{x};\mathbf{x}-\mathbf{x}_I)$:

$$\Psi^{\nabla}_{Ii}(\mathbf{x}) = C^i(\mathbf{x};\mathbf{x}-\mathbf{x}_I)\Phi_a(\mathbf{x}-\mathbf{x}_I) \tag{B.13}$$

where

$$C^i(\mathbf{x};\mathbf{x}-\mathbf{x}_I) = \mathbf{b}^i(\mathbf{x})\mathbf{H}(\mathbf{x}-\mathbf{x}_I) \tag{B.14}$$

$$\mathbf{b}^i(\mathbf{x}) = \mathbf{H}_i^T\mathbf{M}^{-1}(\mathbf{x}). \tag{B.15}$$

Derivatives of (B.13) with respect to x_j are computed directly as

$$\Psi^{\nabla}_{Ii,j}(\mathbf{x}) = C^i_{,j}(\mathbf{x};\mathbf{x}-\mathbf{x}_I)\Phi_a(\mathbf{x}-\mathbf{x}_I) + C^i(\mathbf{x};\mathbf{x}-\mathbf{x}_I)\Phi_{a,j}(\mathbf{x}-\mathbf{x}_I) \tag{B.16}$$

where

$$C^i_{,j}(\mathbf{x};\mathbf{x}-\mathbf{x}_I) = \mathbf{b}^i_{,j}(\mathbf{x})\mathbf{H}(\mathbf{x}-\mathbf{x}_I) + \mathbf{b}^i(\mathbf{x})\mathbf{H}_{,j}(\mathbf{x}-\mathbf{x}_I) \tag{B.17}$$

$$\mathbf{b}^i_{,j}(\mathbf{x}) = \mathbf{H}_i^T\mathbf{M}^{-1}_{,j}(\mathbf{x}). \tag{B.18}$$

For NSNI, the necessary calculations at a node are shown in Table B.3. Consequently, the complexity is given as (44).

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

APPENDIX C: MATRIX FORMS OF VARIATIONALLY CONSISTENT NATURALLY STABILIZED NODAL INTEGRATION

C.1. Poisson problem

The matrix form for the Poisson problem in two dimensions using the present method variationally consistent naturally stabilized nodal integration is given below:

$$\begin{aligned}&\sum_{L=1}^{NP}\left\{\bar{\mathbf{B}}^T(\mathbf{x}_L)\mathbf{B}(\mathbf{x}_L)W_L+\mathbf{B}_x^T(\mathbf{x}_L)\mathbf{B}_x(\mathbf{x}_L)M_{Lx}+\mathbf{B}_y^T(\mathbf{x}_L)\mathbf{B}_y(\mathbf{x}_L)M_{Ly}\right\}\mathbf{d}\\&=\sum_{L=1}^{NP}\mathbf{N}^T(\mathbf{x}_L)s(\mathbf{x}_L)W_L+\sum_{L=1}^{NPh}\mathbf{N}^T(\mathbf{x}_L)h(\mathbf{x}_L)S_L\end{aligned}\tag{C.1}$$

where NPh is the number of integration points on the natural boundary, S_L is an associated weight of those points, and

$$\begin{aligned}\bar{B}_I(\mathbf{x}_L)&=\left[\bar{\Psi}_{I,1}(\mathbf{x}_L)\ \bar{\Psi}_{I,2}(\mathbf{x}_L)\right]^T, & B_I(\mathbf{x}_L)&=\left[\Psi_{I,1}(\mathbf{x}_L)\ \Psi_{I,2}(\mathbf{x}_L)\right]^T,\\B_{xI}(\mathbf{x}_L)&=\left[\Psi^{\nabla}_{I1,1}(\mathbf{x}_L)\ \Psi^{\nabla}_{I1,2}(\mathbf{x}_L)\right]^T, & B_{yI}(\mathbf{x}_L)&=\left[\Psi^{\nabla}_{I2,1}(\mathbf{x}_L)\ \Psi^{\nabla}_{I2,2}(\mathbf{x}_L)\right]^T,\\N_I(\mathbf{x}_L)&=\bar{\Psi}_I(\mathbf{x}_L), & d_I&=u_I,\end{aligned}\tag{C.2}$$

where Ψ_I is the RK shape function in (3), $\Psi_{I,j}$ is the direct derivative of Ψ_I with respect to x_j, $\bar{\Psi}_{I,j}$ is constructed by the VC correction of $\Psi_{I,j}$ given by (47)–(49), and $\Psi^{\nabla}_{Ii,j}(\mathbf{x}_L)$ is the direct differentiation of (27) with respect to x_j.

C.2. Elasticity

The matrix form of Elasticity in two dimensions, using variationally consistent naturally stabilized nodal integration is

$$\begin{aligned}&\sum_{L=1}^{NP}\left\{\bar{\mathbf{B}}^T(\mathbf{x}_L)\mathbf{D}\mathbf{B}(\mathbf{x}_L)W_L+\mathbf{B}_x^T(\mathbf{x}_L)\mathbf{D}\mathbf{B}_x(\mathbf{x}_L)M_{Lx}+\mathbf{B}_y^T(\mathbf{x}_L)\mathbf{D}\mathbf{B}_y(\mathbf{x}_L)M_{Ly}\right\}\mathbf{d}\\&=\sum_{L=1}^{NP}\mathbf{N}^T(\mathbf{x}_L)\mathbf{b}(\mathbf{x}_L)W_L+\sum_{L=1}^{NPh}\mathbf{N}^T(\mathbf{x}_L)\mathbf{h}(\mathbf{x}_L)S_L\end{aligned}\tag{C.3}$$

where NPh is the number of integration points on the natural boundary, S_L is an associated weight of those points, $\mathbf{D}$ is the matrix corresponding to the tensor $\mathbf{C}$, and

$$\begin{aligned}\bar{B}_I(\mathbf{x}_L)&=\begin{bmatrix}\bar{\Psi}_{I,1}(\mathbf{x}_L) & 0\\ 0 & \bar{\Psi}_{I,2}(\mathbf{x}_L)\\ \bar{\Psi}_{I,2}(\mathbf{x}_L) & \bar{\Psi}_{I,1}(\mathbf{x}_L)\end{bmatrix}, & B_I(\mathbf{x}_L)&=\begin{bmatrix}\Psi_{I,1}(\mathbf{x}_L) & 0\\ 0 & \Psi_{I,2}(\mathbf{x}_L)\\ \Psi_{I,2}(\mathbf{x}_L) & \Psi_{I,1}(\mathbf{x}_L)\end{bmatrix},\\B_{xI}(\mathbf{x}_L)&=\begin{bmatrix}\Psi^{\nabla}_{I1,1}(\mathbf{x}_L) & 0\\ 0 & \Psi^{\nabla}_{I1,2}(\mathbf{x}_L)\\ \Psi^{\nabla}_{I1,2}(\mathbf{x}_L) & \Psi^{\nabla}_{I1,1}(\mathbf{x}_L)\end{bmatrix}, & B_{yI}(\mathbf{x}_L)&=\begin{bmatrix}\Psi^{\nabla}_{I2,1}(\mathbf{x}_L) & 0\\ 0 & \Psi^{\nabla}_{I2,2}(\mathbf{x}_L)\\ \Psi^{\nabla}_{I2,2}(\mathbf{x}_L) & \Psi^{\nabla}_{I2,1}(\mathbf{x}_L)\end{bmatrix},\\N_I(\mathbf{x}_L)&=\begin{bmatrix}\Psi_I(\mathbf{x}_L) & 0\\ 0 & \Psi_I(\mathbf{x}_L)\end{bmatrix}, & \mathbf{d}_I&=\begin{bmatrix}u_{I1}\\ u_{I2}\end{bmatrix}.\end{aligned}\tag{C.4}$$

where Ψ_I is the RK shape function in (3), $\Psi_{I,j}$ is the direct derivative of Ψ_I with respect to x_j, $\bar{\Psi}_{I,j}$ is constructed by the VC correction of $\Psi_{I,j}$ given by (47)-(49), and $\Psi^{\nabla}_{Ii,j}(\mathbf{x}_L)$ is the direct differentiation of (27) with respect to x_j

ACKNOWLEDGEMENTS

The support of this work by the US Army Engineer Research and Development Center under contract W15QKN-12-9-1006 is greatly acknowledged.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

REFERENCES

1. Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
2. Belytschko T, Lu YY, Gu L. Crack propagation by element-free Galerkin methods. *Engineering Fracture Mechanics* 1995; **51**:295–459.
3. Liu WK, Jun S, Zhang YF. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids* 1995; **20**:1081–1106.
4. Chen JS, Pan C, Wu CT, Liu WK. Reproducing Kernel particle methods for large deformation analysis of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:195–227.
5. Duarte CA, Oden JT. An *h–p* adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:237–262.
6. Oden JT, Duarte CA, Zienkiewicz OC. A new cloud-based hp finite element method. *Computer Methods in Applied Mechanics and Engineering* 1998; **153**:117–126.
7. Atluri SN, Zhu T. A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* 1998; **22**:117–127.
8. De S, Bathe KJ. The method of finite spheres. *Computational Mechanics* 2000; **24**:329–345.
9. Braun J, Sambridge M. A numerical method for solving partial differential equations on highly irregular evolving grids. *Nature* 1995; **376**:655–660.
10. Sukumar N, Moran B, Belytschko T. The natural element method in solid mechanics. *International Journal for Numerical Methods in Engineering* 1998; **43**:839–887.
11. Sukumar N. Construction of polygonal interpolants: a maximum entropy approach. *International Journal for Numerical Methods in Engineering* 2004; **61**:2159–2181.
12. Arroyo M, Ortiz M. Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. *International Journal for Numerical Methods in Engineering* 2006; **65**:2167–2202.
13. Beissel S, Belytschko T. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:49–74.
14. Belytschko T, Guo Y, Liu WK, Xiao SP. A unified stability analysis of meshless particle methods. *International Journal for Numerical Methods in Engineering* 2000; **48**:1359–1400.
15. Chen JS, Wu CT, Yoon S, You Y. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering* 2001; **50**:435–466.
16. Babuška I, Banerjee U, Osborn JE, Li Q. Quadrature for meshless methods. *International Journal for Numerical Methods in Engineering* 2008; **76**:1434–1470.
17. Chen JS, Hillman M, Rüter M. An arbitrary order variationally consistent integration method for Galerkin meshfree methods. *International Journal for Numerical Methods in Engineering* 2013; **95**:387–418.
18. Kaljević I, Saigal S. An improved element free Galerkin formulation. *International Journal for Numerical Methods in Engineering* 1997; **40**:2953–2974.
19. Dolbow J, Belytschko T. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics* 1999; **23**:219–230.
20. Bonet J, Kulasegaram S. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering* 2000; **47**:1189–1214.
21. Randles PW, Libersky LD. Normalized SPH with stress points. *International Journal for Numerical Methods in Engineering* 2000; **48**:1445–1462.
22. Rabczuk T, Belytschko T, Xiao SP. Stable particle methods based on Lagrangian kernels. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:1035–1063.
23. Chen JS, Yoon S, Wu CT. Nonlinear version of stabilized conforming nodal integration for Galerkin meshfree methods. *International Journal for Numerical Methods in Engineering* 2002; **53**:2587–2615.
24. Griebel M, Schweitzer MA. A particle-partition of unity method–part II: efficient cover construction and reliable integration. *SIAM Journal on Scientific Computing* 2002; **23**:1655–1682.
25. Puso M, Chen JS, Zywicz E, Elmer W. Meshfree and finite element nodal integration methods. *International Journal for Numerical Methods in Engineering* 2008; **74**:416–446.
26. Babuška I, Banerjee U, Osborn JE, Li Q. Effect of numerical integration on meshless methods. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:2886–2897.
27. Liu Y, Belytschko T. A new support integration scheme for the weakform in mesh-free methods. *International Journal for Numerical Methods in Engineering* 2010; **82**:699–715.
28. Liu GR, Zhang GY, Wang YY, Zhong ZH, Li GY, Han X. A nodal integration technique for meshfree radial point interpolation method (NI-RPIM). *International Journal of Solids and Structures* 2007; **44**:3840–3890.
29. Hillman M, Chen JS, Chi SW. Stabilized and variationally consistent nodal integration for meshfree modeling of impact problems. *Computational Particle Mechanics* 2014; **1**:245–256.
30. Nagashima T. Node-by-node meshless approach and its applications to structural analyses. *International Journal for Numerical Methods in Engineering* 1999; **46**:341–385.
31. Fries TP, Belytschko T. Convergence and stabilization of stress-point integration in mesh-free and particle methods. *International Journal for Numerical Methods in Engineering* 2008; **74**:1067–1087.
32. Yoo JW, Moran B, Chen JS. Stabilized conforming nodal integration in the natural-element method. *International Journal for Numerical Methods in Engineering* 2004; **60**:861–890.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme

33. Strang G, Fix G. *An Analysis of the Finite Element Method* (2nd edn). Wellesley-Cambridge Press: Massachusetts, 2008.
34. Duan Q, Li X, Zhang H, Belytschko T. Second-order accurate derivatives and integration schemes for meshfree methods. *International Journal for Numerical Methods in Engineering* 2012; **92**:399–424.
35. Duan Q, Gao X, Wang B, Li X, Zhang H. A four-point integration scheme with quadratic exactness for three-dimensional element-free Galerkin method based on variationally consistent formulation. *Computer Methods in Applied Mechanics and Engineering* 2014; **280**:84–116.
36. Li S, Liu WK. Reproducing kernel hierarchical partition of unity part I: formulation and theory. *International Journal for Numerical Methods in Engineering* 1999; **45**:251–288.
37. Li S, Liu WK. Reproducing kernel hierarchical partition of unity part II: applications. *International Journal for Numerical Methods in Engineering* 1999; **45**:289–317.
38. Chen JS, Zhang X, Belytschko T. An implicit gradient model by a reproducing kernel strain regularization in strain localization problems. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:2827–2844.
39. Chi SW, Chen JS, Hu HY, Yang JP. A gradient reproducing kernel collocation method for boundary value problems. *International Journal for Numerical Methods in Engineering* 2013; **93**:1381–1402.
40. Chen JS, Wu Y, Guan PC, Teng H, Gaidos J, Hofstetter K, Alsaleh M. A semi-Lagrangian reproducing kernel formulation for modeling earth moving operations. *Mechanics of Materials* 2009; **41**:670–683.
41. Guan PC, Chi SW, Chen JS, Slawson TR, Roth MJ. Semi-Lagrangian reproducing kernel particle method for fragment-impact problems. *International Journal of Impact Engineering* 2011; **38**:1033–1047.
42. Chen JS, Hu W, Puso MA, Wu Y, Zhang X. Strain smoothing for stabilization and regularization of Galerkin mesh-free methods. In *Meshfree Methods for Partial Differential Equations III*, Griebel M, Schweitzer MA (eds). Springer: Berlin, 2007; 57–75.
43. Liu WK, Ong JSJ, Uras RA. Finite element stabilization matrices-a unification approach. *Computer Methods in Applied Mechanics and Engineering* 1985; **53**:13–46.
44. Hu HY, Lai CK, Chen JS. A study on convergence and complexity of reproducing kernel collocation method. *Interaction and Multiscale Mechanics* 2009; **2**:295–319.
45. Quarteroni A, Valli A. *Numerical Approximation of Partial Differential Equations*, Springer Series in Computational Mathematics, vol. 23. Springer: Berlin, 1994.
46. Norris DM, Moran B, Scudder JK, Quinones DF. A computer simulation of the tension test. *Journal of the Mechanics and Physics of Solids* 1978; **26**:1–19.
47. Taylor G. The use of flat-ended projectiles for determining dynamic yield stress, Part I. *Proceedings of the Royal Society of London A* 1948; **194**:289–299.
48. Wilkins ML, Guinan MW. Impact of cylinders on rigid boundary. *Journal of Applied Physics* 1973; **44**:1200–1206.
49. Sulsky D, Zhou SJ, Schreyer HL. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* 1995; **87**:236–252.
50. Hughes TJ, Winget J. Finite rotation effects in numerical integration of rate constitutive equations arising in large-deformation analysis. *International Journal for Numerical Methods in Engineering* 1980; **15**:1862–1867.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme