INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING

Int. J. Numer. Meth. Engng 2015; **101**:127–147

Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.4792

Axisymmetric form of the generalized interpolation material point method

John A. Nairn[1]*and James E. Guilkey[2]

[1]*Wood Science and Engineering, Oregon State University, Corvallis, OR, USA*
[2]*Schlumberger Technology Corporation, Rasharon, TX, USA*

SUMMARY

This paper reformulates the axisymmetric form of the material point method (MPM) using generalized interpolation material point (GIMP) methods. The reformulation led to a need for new shape functions and gradients specific for axisymmetry that were not available before. The new shape functions differ most from planar shape functions near the origin where $r = 0$. A second purpose for this paper was to evaluate the consequences of axisymmetry on a variety MPM extensions that have been developed since the original work on axisymmetric MPM. These extensions included convected particle domain integration (CPDI), traction boundary conditions, explicit cracks, multimaterial mode MPM for contact, thermal conduction, and solvent diffusion. Some examples show that the axisymmetric shape functions work well and are especially crucial near the origin. One real-world example is given for modeling a cylinder-penetration problem. Finally, a check list for software development describes all tasks needed to convert 2D planar or 3D codes to include an option for axisymmetric MPM. Copyright © 2015 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: Material Point Method, Axisymmetry, Penetration, MPM, GIMP, CPDI

Received 10 January 2013; Revised 20 December 2013; Accepted 18 August 2014

1. INTRODUCTION

The Material Point Method (MPM) [1, 2] is enjoying increasing popularity as a computational tool for solid mechanics investigations, particularly for simulations involving large deformations [3], contact [4, 5, 6], and fracture [7, 8, 9]. In addition, its utility for treating history dependent material properties has led to MPM being implemented in the historically Eulerian CTH software from Sandia National Laboratory [10]. Extensions to the original MPM formulations described by Sulsky *et al.* [1, 2] include the Generalized Interpolation Material Point (GIMP) Method [11] and more recently the Convected Particle Domain Interpolation (CPDI) Method [3], which have done a great deal to improve the robustness and accuracy of the original MPM.

*Correspondence to: Wood Science and Engineering, 119 Richardson Hall, Oregon State University, Corvallis, OR, USA. E-mail: John.Nairn@oregonstate.edu

Copyright © 2015 John Wiley & Sons, Ltd.

Many interesting scenarios from the preceding categories lend themselves to being treated as axisymmetric. These include, for example, impact and cratering events, rod penetration and shaped charge jet formation. Sulsky and Schreyer presented the first, and thus far, only, formulation for axisymmetric MPM [12]. Unfortunately, this prior axisymmetric MPM was based on early MPM methods that treated particles as point masses. It has long been recognized that this original MPM (*i.e.*, particles represented spatially as Dirac delta functions) has serious element crossing artifacts and should not be used. They have been replaced by GIMP methods [11], which have been developed for cartesian coordinates, but are not available for axisymmetric calculations. In addition to reformulation of axisymmetry using GIMP methods, numerous other extensions to MPM have developed since the original work on axisymmetric MPM [12]. Because none of these extensions considered axisymmetry, each one needs to be re-examined for use in axisymmetric code.

This paper derives axisymmetric MPM methods within the GIMP framework, first for the so-called uniform GIMP [11], where particle domains' spatial extents and orientations remain unchanged during a simulation, and then also for CPDI [3], where particle domains are allowed to evolve. Those methods are not reviewed here; readers are encouraged to consult the original references to review the context in which the current formulation is derived. Following derivation of relevant formulae for the basic axisymmetric methods, descriptions are provided for various additional features of MPM that require modification when used in axisymmetric simulations. These include — traction boundary conditions, multi-material contact, representation of cracks, diffusion calculations, and heat conduction. Following the algorithmic descriptions are examples intended to demonstrate the efficacy of these formulations. Finally, a check list for software development is presented to make addition of an axisymmetric capability as straightforward as possible for the reader.

2. GENERALIZED MATERIAL POINT METHOD

Although Sulsky and Schreyer [12] present one form of axisymmetric MPM, it is not a convenient starting point. Instead, we started from the cartesian formulation of GIMP [11] and modified it for axisymmetry. The key changes were to replace volume integration by cylindrical volume integration and to account for cylindrical coordinates in gradient evaluations. The main MPM result for momentum rate on node i becomes:

$$\frac{d\boldsymbol{p}_i}{dt} = \boldsymbol{f}_i^{(int)} + \boldsymbol{f}_i^{(ext)} + \boldsymbol{f}_i^{(b)} + \boldsymbol{f}_i^{(s)} \tag{1}$$

where nodal momenta are found by summation over all particles, p:

$$\boldsymbol{p}_i = \sum_p \boldsymbol{p}_p S_{ip} \tag{2}$$

where $\boldsymbol{p}_p$ is momentum of particle p and S_{ip} is an axisymmetric GIMP shape function defined below. The $\boldsymbol{f}$ terms are nodal forces. $\boldsymbol{f}_i^{(int)}$ is internal force due to specific Cauchy stresses on

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

the particles:

$$\boldsymbol{f}_i^{(int)} = -\sum_p m_p \begin{pmatrix} \sigma^{(s)}_{rr,p} & \sigma^{(s)}_{rz,p} \\ \sigma^{(s)}_{rz,p} & \sigma^{(s)}_{zz,p} \end{pmatrix} \boldsymbol{G}_{ip} - \sum_p m_p \left(\sigma^{(s)}_{\theta\theta,p}, 0\right) T_{ip} \tag{3}$$

Here m_p is particle mass and $\boldsymbol{G}_{ip}$ and T_{ip} are axisymmetric GIMP shape function gradients defined below. Nodal force $\boldsymbol{f}_i^{(ext)}$ and $\boldsymbol{f}_i^{(b)}$ are forces due to external loads, $\boldsymbol{F}_p$, and body forces per unit mass, $\boldsymbol{b}_p$, on the particles:

$$\boldsymbol{f}_i^{(ext)} = \sum_p \boldsymbol{F}_p S_{ip} \qquad \text{and} \qquad \boldsymbol{f}_i^{(b)} = \sum_p m_p \boldsymbol{b}_p S_{ip} \tag{4}$$

where S_{ip} is an axisymmetric GIMP shape function defined below. Finally, $\boldsymbol{f}_i^{(s)}$ is force due to surface tractions, $\boldsymbol{T}$:

$$\boldsymbol{f}_i^{(s)} = \frac{1}{2\pi} \int \boldsymbol{T} N_i(r,z)\, dS \tag{5}$$

which is integrated over the object boundary and $N_i(r,z)$ are standard MPM grid shape functions.

The axisymmetric GIMP shape functions and gradients are defined by

$$S_{ip} = \frac{1}{A_p \langle r_p \rangle} \int_{\Omega_p} r\, \chi_p(r,z) N_i(r,z)\, dr\, dz \tag{6}$$

$$G_{ip} = \frac{1}{A_p \langle r_p \rangle} \int_{\Omega_p} r\, \chi_p(r,z) \nabla N_i(r,z)\, dr\, dz \tag{7}$$

$$T_{ip} = \frac{1}{A_p \langle r_p \rangle} \int_{\Omega_p} \chi_p(r,z) N_i(r,z)\, dr\, dz \tag{8}$$

where Ω_p is the deformed particle domain in the r-z plane, $\chi_p(r,z)$ is a particle weighting function (its various choices are what make this method “generalized”), A_p is area of the domain, and $\langle r_p \rangle$ is mean radial position of the particle domain. Each MPM time step involves evaluation of constitutive laws on the particles, which must be implemented in material models. The deformation gradient is updated on each time step using $\mathbf{F}^{(n+1)} = d\mathbf{F}\,\mathbf{F}^{(n)}$ where $d\mathbf{F}$ is the incremental deformation gradient given by:

$$d\mathbf{F} = \exp(\Delta t \nabla v) = \mathbf{I} + \nabla u + \sum_{k=2}^{\infty} \frac{(\nabla u)^k}{k!} \tag{9}$$

where Δt is the time step and ∇v and ∇u are the velocity and displacement gradients. In many problems with sufficient resolution, $d\mathbf{F}$ can be found from the first two terms, but in problems involving rotation (less common in axisymmetry, but possible in penetration), the $k = 2$ or higher terms are needed (note that $k > 2$ terms in axisymmetry can be evaluated without needing any matrix multiplications by using the Cayley–Hamilton theorem; they therefore can

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

be included efficiently). The displacement gradient is given by

$$\nabla u = \begin{pmatrix} \frac{\partial u_r}{\partial r} & \frac{\partial u_r}{\partial z} & 0 \\ \frac{\partial u_z}{\partial r} & \frac{\partial u_z}{\partial z} & 0 \\ 0 & 0 & \frac{du_r}{r_p} \end{pmatrix} \tag{10}$$

where u_r and u_z are the r and z displacements. The deformation gradient terms are found by extrapolating nodal velocities, $\boldsymbol{v}_i$, to the particles:

$$\boldsymbol{v}_i = \boldsymbol{p}_i / \sum_p m_p S_{ip} \tag{11}$$

$$\left(\frac{\partial u_r}{\partial r}, \frac{\partial u_r}{\partial z}\right) = \Delta t \sum_i v_{i,r} \boldsymbol{G}_{ip} \tag{12}$$

$$\left(\frac{\partial u_z}{\partial r}, \frac{\partial u_z}{\partial z}\right) = \Delta t \sum_i v_{i,z} \boldsymbol{G}_{ip} \tag{13}$$

$$\frac{du_r}{r_p} = \Delta t \sum_i v_{i,r} T_{ip} \tag{14}$$

The original, or “classic,” axisymmetric MPM result [12] is recovered by selecting $\chi(\boldsymbol{r}) = A_p \delta(\boldsymbol{r}_p)$, resulting in $S_{ip} = N_i(r_p, z_p)$, $G_{ip} = \nabla N_i(r_p, z_p)$, and $T_{ip} = N_i(r_p, z_p)/r_p$.

The main differences between planar GIMP [11] and axisymmetric GIMP are the $\sigma^{(s)}_{\theta\theta,p}$ term in $\boldsymbol{f}^{(int)}_i$, particle masses are per radian ($m_p = \rho A_p \langle r_p \rangle$) and vary with position, and the incremental deformation gradient has du_r/r_p for hoop direction strain. Furthermore, the r in the integrands causes S_{ip} and $\boldsymbol{G}_{ip}$ to differ from the corresponding planar GIMP functions and axisymmetry requires a new GIMP shape function denoted here as T_{ip}. The main difference between a “classic” MPM version of axisymmetric MPM [12] and a GIMP version of axisymmetric MPM is that new shape functions are needed. In “classic” MPM, it turned out that axisymmetric MPM could re-use the planar shape functions, but when axisymmetric MPM is extended to GIMP, new shape functions are required and those shape function are not available in the literature. The rest of this section derives these new axisymmetric shape functions for two forms of GIMP — uniform GIMP (uGIMP) and convected particle domain integration GIMP (CPDI) [3].

2.1. Uniform GIMP

In uGIMP the particle domain is assumed to remain undeformed at the size of the initial domain but translates along with the particle and $\chi_p(r,z) = 1$ within the domain and zero elsewhere. This approach is commonly used because of numerical difficulty evaluating GIMP integrals for arbitrarily deformed particle domains. For simpler explicit calculations, uGIMP is restricted to a regular grid with all elements having size $\Delta r \times \Delta z$. For calculation at node i, we use dimensionless coordinates:

$$\xi = 2\frac{r - r_i}{\Delta r} \quad \text{and} \quad \eta = 2\frac{z - z_i}{\Delta z} \tag{15}$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

such that node i is at $(0,0)$ and element size is transformed to be 2×2. The uGIMP shape function can be partitioned into r and z components:

$$S_{ip} = S_r(\xi_p, r_i) S_z(\eta_p) \tag{16}$$

where (ξ_p, η_p) is the dimensionless particle location particle and $S_z(\eta_p)$ is identical to the partitioning in planar uGIMP (see Appendix A) but $S_r(\xi_p, r_i)$ is different and given by:

$$S_r(\xi_p, r_i) = \frac{1}{W_p\langle r_p\rangle}\int_{r_{min}}^{r_{max}} r\,N_i(r)\,dr = \frac{1}{2l_p(\xi_p+2n_i)}\int_{\xi_p-l_p}^{\xi_p+l_p}(\xi+2n_i)N_i(\xi)d\xi \tag{17}$$

where $W_p = r_{max} - r_{min}$ is the width of the particle domain, $\langle r_p\rangle = (r_{max}+r_{min})/2$ is its centroid, $2l_p$ is the domain width in dimensionless units, $n_i = r_i/\Delta r$, and

$$N_i(\xi) = \begin{cases} \dfrac{2+\xi}{2} & -2<\xi<0 \\ \dfrac{2-\xi}{2} & 0\le\xi<2 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

Accounting for all possible overlaps between the particle domain and the regions of $N_i(\xi)$, the above integral evaluates to:

$$S_r(\xi_p, n_i) = \begin{cases} \dfrac{(2+l_p+\xi_p)^2}{8l_p}\left(1-\dfrac{2(1-l_p)+\xi_p}{3(2n_i+\xi_p)}\right) & -2-l_p<\xi_p<-2+l_p \\ \dfrac{2+\xi_p}{2}+\dfrac{l_p^2}{6(2n_i+\xi_p)} & -2+l_p\le\xi_p<-l_p \\ \dfrac{(4-l_p)l_p-\xi_p^2}{4l_p}+\dfrac{\xi_p(\xi_p^2-3l_p^2)}{12l_p(2n_i+\xi_p)} & -l_p\le\xi_p<l_p \\ \dfrac{2-\xi_p}{2}-\dfrac{l_p^2}{6(2n_i+\xi_p)} & l_p\le\xi_p<2-l_p \\ \dfrac{(2+l_p-\xi_p)^2}{8l_p}\left(1+\dfrac{2(1-l_p)-\xi_p}{3(2n_i+\xi_p)}\right) & 2-l_p\le\xi_p<2+l_p \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

The domain regions for $S_r(\xi_p, r_i)$ around node i are shown in Fig. 1. $S_r(\xi_p, r_i)$ is non-zero between $\pm(2+l_p)$. This result has been expressed as planar GIMP result plus an additional term that depends on $1/(2n_i+\xi_p)$. Clearly, as $n_i\to\infty$, $S_r(\xi_p, n_i)$ approaches planar uGIMP shape functions (*i.e.,* $S_z(\xi_p)$). But, near the origin they are different.

The above integration satisfies partition of unity ($\sum_i S_r(\xi_p, r_i) = 1$), but when particles are within $l_p\Delta r/2$ of the origin, the particle domain crosses $r=0$. As r_p approaches zero, some shape functions will have singularities at $\xi_p = -2n_i$. We show below that this non-physical

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

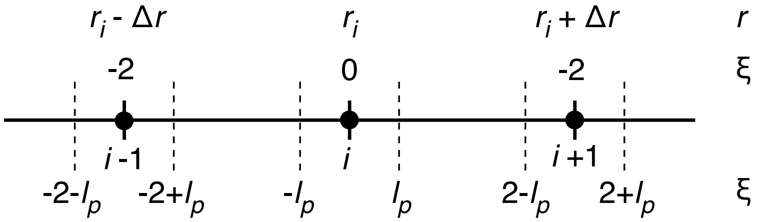

Figure 1. One dimensional view for the extent of $S_r(\xi_p, n)$ around node i. The dashed lines delimit the five regions listed in Eq. (19). The shape function for node i is nonzero only between $-(2+l_p)$ and $2+l_p$, which means it extends to the second neighbor grid cells.

extension of the domain and shape function singularities can lead to loss of accuracy for particles near the origin. The results are improved by truncating the particle domain, which is only needed when $n_i < (1+l_p/2)$ and $r_p < l_p\Delta r/2$ (or $\xi_p < -2n_i + l_p$). For simplicity, if the grid includes any nodes with $n_i < (1+l_p/2)$, the grid is required have nodes at $r=0$ and therefore all n_i in a regular grid will be integers. In such a grid, special case evaluations are needed only for $n_i = -1$, 0, and 1.

The simplest approach in uGIMP is to truncate the particle domain at $r=0$ and to disallow particle positions less than $r=0$. For $n_i = -1$, the shape function reduces to zero ($S_r^{(-1)}(\xi_p) = 0$). For $n_i = 0$, new evaluation is needed when $0 \le \xi_p < l_p$. In this case, the truncated particle domain width and centroid in dimensionless units are $\xi_p + l_p$ and $(\xi_p + l_p)/2$; the required integral reduces to

$$S_r^{(0)}(\xi_p) = \frac{2}{(\xi_p + l_p)^2}\int_0^{\xi_p + l_p} \xi N_i(\xi)d\xi = \begin{cases} 0 & \xi < 0 \\ \dfrac{3 - l_p - \xi_p}{3} & 0 \le \xi_p < l_p \\ S_r(\xi_p, 0) & \text{otherwise} \end{cases} \tag{20}$$

For $n_i = 1$, new evaluation is needed when $-2 < \xi_p < -2 + l_p$. In this case, the truncated particle domain width and centroid in dimensionless units are $\xi_p + l_p + 2$ and $(\xi_p + l_p - 2)/2$; the required integral changes to

$$S_r^{(1)}(\xi_p) = \frac{2}{(\xi_p + l_p + 2)^2}\int_{-2}^{\xi_p + l_p} (\xi + 2)N_i(\xi)d\xi = \begin{cases} 0 & \xi < -2 \\ \dfrac{2 + l_p + \xi_p}{3} & -2 \le \xi_p < -2 + l_p \\ S_r(\xi_p, 1) & \text{otherwise} \end{cases} \tag{21}$$

Figure 2 shows $S_r(\xi_p, n)$ for $n = -1$, 0, 1, and 2 and $l_p = 1/2$. The dashed lines show planar results. The differences are most noticeable for r near zero. The dotted lines near the origin are $S_r(\xi_p, -1)$, $S_r(\xi_p, 0)$, and $S_r(\xi_p, 1)$ that do not truncate the particle domain as discussed above.

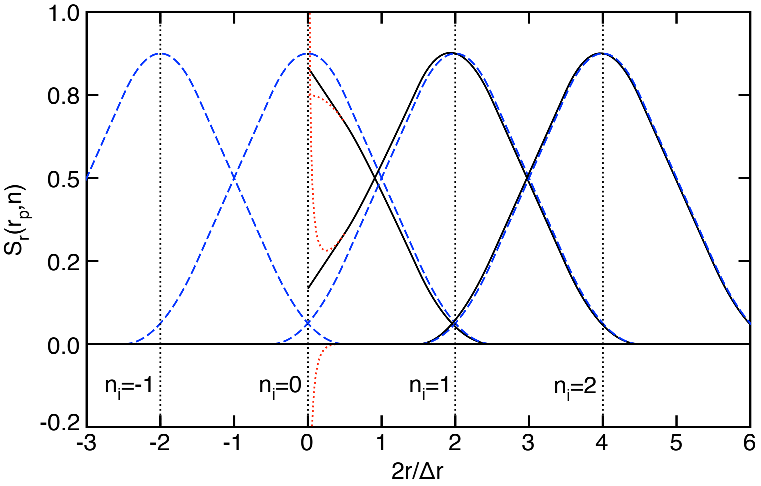

Figure 2. The solid lines show $S_r(\xi_p,n)$ for $n_i = 0$, 1, and 2 with nodes located at the vertical, dotted lines. The dashed lines shows planar results for n_i from -1 to 2. The dotted red lines are calculations that do not truncate the particle domain for $n_i = -1$, 0, and 1. The result for $n_i = -1$ is just the tail singularity that is negative.

Both $S_r(\xi_p,-1)$ and $S_r(\xi_p,1)$ are singular, although the singularities cancel to retain partition of unity. Some calculations below show these singularities can reduce accuracy.

The uGIMP gradient shape functions can be partitioned into r and z components:

$$G_{ip} = \left(\frac{2}{\Delta r} dS_r(\xi_p,r_i)S_z(\eta_p), \frac{2}{\Delta z} S_r(\xi_p,r_i)dS_z(\eta_p)\right) \tag{22}$$

where $dS_z(\eta_p)$ is identical to the partitioning in planar uGIMP (see appendix) but $dS_r(\xi_p,r_i)$ is different and given by:

$$dS_r(\xi_p,r_i) = \frac{\Delta r}{2W_p\langle r_p\rangle}\int_{r_{min}}^{r_{max}} r\,\frac{dN_i(r)}{dr}\,dr = \frac{1}{2l_p(\xi_p+2n_i)}\int_{\xi_p-l_p}^{\xi_p+l_p}(\xi+2n_i)\frac{dN_i(\xi)}{d\xi}d\xi \tag{23}$$

This term evaluates to

$$dS_r(\xi_p,n_i) = \begin{cases} \dfrac{(2+l_p+\xi_p)}{4l_p}\left(1-\dfrac{2-l_p+\xi_p}{2(2n_i+\xi_p)}\right) & -2-l_p<\xi_p<-2+l_p \\ 1/2 & -2+l_p<\xi_p<-l_p \\ -\dfrac{\xi_p}{2l_p}-\dfrac{l_p^2-\xi_p^2}{4l_p(2n_i+\xi_p)} & -l_p<\xi_p<l_p \\ -1/2 & l_p<\xi_p<2-l_p \\ -\dfrac{(2+l_p-\xi_p)}{4l_p}\left(1+\dfrac{2-l_p-\xi_p}{2(2n_i+\xi_p)}\right) & 2-l_p<\xi_p<2+l_p \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

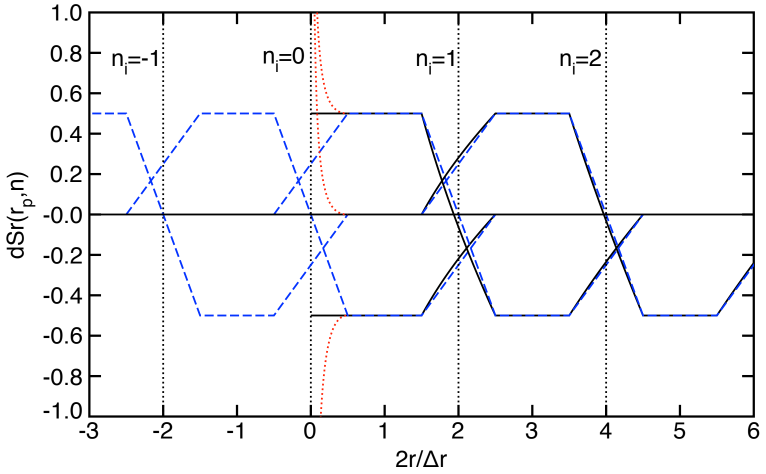

Figure 3. The solid lines show $dS_r(\xi_p, n)$ for $n_i = 0$, 1, and 2 with nodes located at the vertical, dotted lines. The dashed lines shows planar results for n_i from -1 to 2. The dotted red lines are calculations that do not truncate the particle domain for $n_i = -1$, 0, and 1. The result for $n_i = -1$ is just the tail singularity that rises from zero.

This result has been expressed as planar GIMP result plus an additional term (if needed) that depends on $1/(2n_i + \xi_p)$. Clearly, as $n_i \to \infty$, $dS_r(\xi_p, n_i)$ approaches planar uGIMP (*i.e.*, $dS_z(\xi_p)$). But, near the origin they are different. Again, the particle domain needs to be truncated for $n_i = -1$, 0, and 1. For $n_i = -1$, $dS_r^{(0)}(\xi_p) = 0$. For $n_i = 0$, the integral changes to:

$$dS_r^{(0)}(\xi_p) = \frac{2}{(\xi_p + l_p)^2} \int_{\xi_p - l_p}^{\xi_p + l_p} \xi \frac{dN_i(\xi)}{d\xi} d\xi = \begin{cases} 0 & \xi < 0 \\ -1/2 & 0 \le \xi_p < l_p \\ dS_r(\xi_p, 0) & \text{otherwise} \end{cases} \tag{25}$$

For $n_i = 1$, the integral changes to:

$$dS_r^{(1)}(\xi_p) = \frac{2}{(\xi_p + l_p + 2)^2} \int_{-2}^{\xi_p + l_p} (\xi + 2) \frac{dN_i(\xi)}{d\xi} d\xi = \begin{cases} 0 & \xi < -2 \\ 1/2 & -2 < \xi_p < -2 + l_p \\ dS_r(\xi_p, 1) & \text{otherwise} \end{cases} \tag{26}$$

Figure 3 shows $dS_r(\xi_p, n)$ for $n = -1$, 0, 1, and 2 and $l_p = 1/2$. The dashed lines show the planar results. The differences are most noticeable for r near zero. The dotted lines near the origin are $dS_r(\xi_p, -1)$, $dS_r(\xi_p, 0)$, and $dS_r(\xi_p, 1)$ that do not truncate the particle domain and all are singular. The truncation eliminates $dS_r(\xi_p, -1)$ and extends $dS_r(\xi_p, 0)$ and $dS_r(\xi_p, 1)$ to the origin at constant ±1/2.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

Finally T_{ip} is equal to planar uGIMP shape function except that it is divided by r_p:

$$T_{ip} = \frac{2}{\Delta r(2n+\xi_p)} S_z(\xi_p) S_z(\eta_p) \tag{27}$$

and that it needs truncation for $n_i = -1$, 0, and 1. Those results are $T_{ip}^{(-1)} = 0$,

$$T_{ip}^{(0)} = \begin{cases} 0 & \xi < 0 \\ \dfrac{1}{\Delta r}\left(\dfrac{4}{l_p + x_p} - 1\right) S_z(\eta_p) & 0 < \xi_p < l_p \\ \dfrac{2}{\Delta r \xi_p} S_z(\xi_p) S_z(\eta_p) & \text{otherwise} \end{cases} \tag{28}$$

and

$$T_{ip}^{(1)} = \begin{cases} 0 & \xi < -2 \\ \dfrac{1}{\Delta r} S_z(\eta_p) & -2 < \xi_p < -2 + l_p \\ \dfrac{2}{\Delta r(2+\xi_p)} S_z(\xi_p) S_z(\eta_p) & \text{otherwise} \end{cases} \tag{29}$$

2.2. Convected Particle Domain Integration

Although uGIMP explicitly evaluates S_{ip}, $\boldsymbol{G}_{ip}$, and T_{ip}, it will lose accuracy as the actual particle domain deforms. For example, large-deformation tension tests are limited to about 50% strain when using uGIMP. The solution is to evaluate the integrals over the deformed particle domain. Although this task is easy in 1D [11], it is very difficult in 2D or 3D. One approach, called convected particle domain integration method (CPDI) [3], integrates over the deformed domain, but makes that task efficient by replacing $N_i(r,z)$ in the integrand by an approximation:

$$S_{ip} = \frac{1}{A_p \langle r_p \rangle} \int_{\Omega_p} r \left(\sum_{\alpha=1}^{4} M_\alpha(r,z) N_i(\boldsymbol{r}_p^{(\alpha)}) \right) dr\, dz \tag{30}$$

where $M_\alpha(r,z)$ are shape functions in the particle domain and $\boldsymbol{r}_p^{(\alpha)}$ are the coordinates for the four corners of the domain (see Fig. 4).

This integral is evaluated as follows. First, the deformed particle domain is evaluated as a parallelogram based on the particle deformation gradient, $\mathbf{F}$, and defined by vectors $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ (see Fig. 4):

$$\boldsymbol{d}_1 = (F_{rr}\Delta r, F_{zr}\Delta z)/4 \qquad \text{and} \qquad \boldsymbol{d}_2 = (F_{rz}\Delta r, F_{zz}\Delta z)/4 \tag{31}$$

Next define natural coordinates within the parallelogram

$$(r,z) = \boldsymbol{r}_p + \rho \boldsymbol{d}_1 + \zeta \boldsymbol{d}_2 = \left(\sum_{\alpha=1}^{4} r_\alpha M_\alpha(\rho,\zeta), \sum_{\alpha=1}^{4} z_\alpha M_\alpha(\rho,\zeta) \right) \tag{32}$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

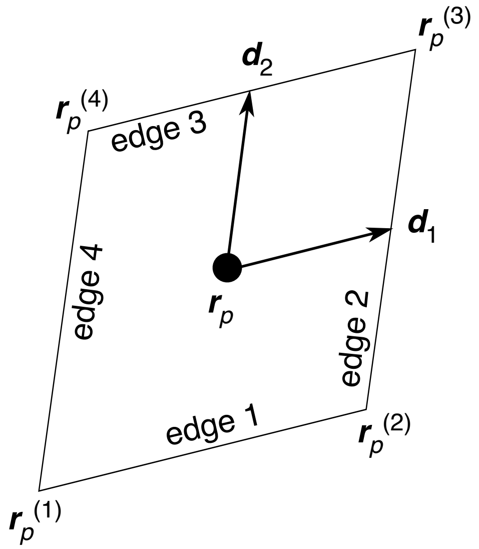

Figure 4. Particle domain deformed into a parallelogram showing vectors $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$, four corners, and four edges. The corners and edges are numbered in the counter clockwise direction from the lower-left corner.

where ρ and ζ range from -1 to 1 and $M_\alpha(\rho,\zeta)$ are standard 2D, four-node isoparametric shape functions (*e.g.*, $M_1(\rho,\zeta)=(1-\rho)(1-\zeta)/4$, *etc.* in the counter-clockwise direction). Transforming Eq. (30) to parallelogram coordinates gives:

$$S_{ip}=\frac{1}{A_p\langle r_p\rangle}\int_{-1}^{1}\int_{-1}^{1}\left(\sum_{\alpha=1}^{4} r_\alpha M_\alpha(\rho,\zeta)\right)\left(\sum_{\alpha=1}^{4} M_\alpha(\rho,\zeta)N_i(\boldsymbol{r}_p^{(\alpha)})\right)|\mathbf{J}|\,d\rho\,d\zeta \tag{33}$$

where $\mathbf{J}$ is the Jacobian of the transformation

$$\mathbf{J}=\begin{pmatrix} d_{1r} & d_{1z} \\ d_{2r} & d_{2z}\end{pmatrix} \tag{34}$$

For a parallelogram, $|\mathbf{J}|=d_{1r}d_{2z}-d_{2r}d_{1z}=A_p/4$, leading to

$$\begin{aligned} S_{ip} \quad = \quad & \frac{1}{4}\Bigg[\left(1-\frac{d_{1r}+d_{2r}}{3r_p}\right)N_i(\boldsymbol{r}_p^{(1)})+\left(1+\frac{d_{1r}-d_{2r}}{3r_p}\right)N_i(\boldsymbol{r}_p^{(2)}) \\ & +\left(1+\frac{d_{1r}+d_{2r}}{3r_p}\right)N_i(\boldsymbol{r}_p^{(3)})+\left(1-\frac{d_{1r}-d_{2r}}{3r_p}\right)N_i(\boldsymbol{r}_p^{(4)})\Bigg] \end{aligned} \tag{35}$$

As $r_p\to\infty$, the axisymmetric S_{ip} for CPDI approaches the planar CPDI result [3], but near the origin they are different.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

A similar analysis for the $\boldsymbol{G}_{ip}$ integral leads to:

$$\begin{aligned}\boldsymbol{G}_{ip} \quad &= \quad \frac{1}{A_p\langle r_p\rangle}\int_{-1}^{1}\int_{-1}^{1}\left(\sum_{\alpha=1}^{4} r_\alpha M_\alpha(\rho,\zeta)\right) \\ &\times\left(\sum_{\alpha=1}^{4}\mathbf{J}^{-1}\left(\frac{\partial M_\alpha(\rho,\zeta)}{\partial\rho},\frac{\partial M_\alpha(\rho,\zeta)}{\partial\zeta}\right)N_i(\boldsymbol{r}_p^{(\alpha)})\right)|\mathbf{J}|\,d\rho\,d\zeta \end{aligned}\tag{36}$$

which evaluates to:

$$\begin{aligned}\boldsymbol{G}_{ip} = \frac{1}{A_p}\Bigg[&\left(d_{1z}-d_{2z}-\frac{\beta}{3r_p}\right)N_i(\boldsymbol{r}_p^{(1)})+\left(d_{1z}+d_{2z}+\frac{\beta}{3r_p}\right)N_i(\boldsymbol{r}_p^{(2)}) \\ &-\left(d_{1z}-d_{2z}+\frac{\beta}{3r_p}\right)N_i(\boldsymbol{r}_p^{(3)})-\left(d_{1z}+d_{2z}-\frac{\beta}{3r_p}\right)N_i(\boldsymbol{r}_p^{(4)}), \\ -\,(d_{1r}-d_{2r})&\left(1-\frac{d_{1r}+d_{2r}}{3r_p}\right)N_i(\boldsymbol{r}_p^{(1)})-(d_{1r}+d_{2r})\left(1+\frac{d_{1r}-d_{2r}}{3r_p}\right)N_i(\boldsymbol{r}_p^{(2)}) \\ +\,(d_{1r}-d_{2r})&\left(1+\frac{d_{1r}+d_{2r}}{3r_p}\right)N_i(\boldsymbol{r}_p^{(3)})+(d_{1r}+d_{2r})\left(1-\frac{d_{1r}-d_{2r}}{3r_p}\right)N_i(\boldsymbol{r}_p^{(4)})\Bigg]\end{aligned}\tag{37}$$

where

$$\beta = d_{1r}d_{1z}-d_{2r}d_{2z}\tag{38}$$

Again, the axisymmetric $\boldsymbol{G}_{ip}$ for CPDI approaches the planar CPDI result [3] as $r_p\to\infty$, but near the origin they are different.

Finally, T_{ip} is equal to planar S_{ip} for CPDI [3] divided by r_p or:

$$T_{ip} = \frac{1}{4r_p}\left[N_i(\boldsymbol{r}_p^{(1)})+N_i(\boldsymbol{r}_p^{(2)})+N_i(\boldsymbol{r}_p^{(3)})+N_i(\boldsymbol{r}_p^{(4)})\right]\tag{39}$$

In accurate simulations, the deformed domain should naturally stay within valid territory, but cruder calculations or round-off error might cause some corners to pass $r=0$ (and they do in some sample calculations). Truncating as done in uGIMP cannot be used because it would leave a non-quadrilateral particle domain that is difficult to integrate. The option proposed here is simply to shrink the domain to fit. This can be done by looking at $r_p-|d_{1r}+d_{2r}|$. If this result is negative, at least one corner has $r<0$. When this occurs, define $s=r_p/|d_{1r}+d_{2r}|$ and then scale $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ by s to get a truncated parallelogram that just touches the origin, but is entirely within $r\geq 0$. Calculations below show this truncation improves the results.

3. GENERALIZED MATERIAL POINT METHOD EXTENSIONS

As MPM has matured, various extensions have been developed. Whenever deriving a new form of MPM, such as this new axisymmetric form of MPM, it is crucial to evaluate the consequences of the new form on all other MPM options. One motivation for this paper was that the original axisymmetric MPM [12] was described for classic MPM only and did not consider any subsequent extensions of MPM (because they all appeared after Ref. [12]). The previous

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

section derived a new axisymmetric MPM for GIMP methods [11]. This section considers the affect of axisymmetry on other important MPM extensions.

3.1. Traction Boundary Conditions

The most accurate method in MPM for applying tractions is to use the surface force term in Eq. (5), but it is rarely done. Its problem is the challenge of integrating over the deformed surface of the object, much like uGIMP is used to avoid integrating over the deformed particle domain. But, this challenge can be solved by exploiting CPDI methods [3]. The process is to replace the integrand by an approximation to the shape function and then integrate over one specified edge of the deformed particle domain. Those edges with tractions are pre-selected using simulation boundary conditions.

For example, consider application of constant traction $\boldsymbol{T}$ along edge 1 of a particle, which, after deformation, will be traction along the corresponding edge of its deformed parallelogram (see Fig. 4). By parameterizing the curve along edge 1 using $\boldsymbol{g}(\rho) = (r(\rho), z(\rho)) = \boldsymbol{r}_p + \rho\boldsymbol{d}_1 - \boldsymbol{d}_2$, where ρ varies from -1 to 1, and realizing that surface increment is $dS = 2\pi r(\rho)|g'(\rho)|d\rho = 2\pi r(\rho)|\boldsymbol{d}_1|d\rho$, the surface integral transforms to a line integral:

$$\boldsymbol{f}^{(s)}_{ip,1} = \frac{1}{2\pi}\int \boldsymbol{T} N_i(r,z)\,dS = \boldsymbol{T}|\boldsymbol{d}_1|\int_{-1}^{1} (\boldsymbol{g}(\rho)\cdot(1,0)) \left(\sum_{\alpha=1}^{4} M_\alpha(\rho,-1) N_i(\boldsymbol{r}_p^{(\alpha)})\right) d\rho \tag{40}$$

which evaluates to

$$\boldsymbol{f}^{(s)}_{ip,1} = |\boldsymbol{d}_1|\left[\left(r_{px} - d_{2x} - \frac{d_{1x}}{3}\right) N_i(\boldsymbol{r}_p^{(1)}) + \left(r_{px} - d_{2x} + \frac{d_{1x}}{3}\right) N_i(\boldsymbol{r}_p^{(2)})\right]\boldsymbol{T} \tag{41}$$

Similar analyses for the remaining edges give:

$$\begin{aligned}
\boldsymbol{f}^{(s)}_{ip,2} &= |\boldsymbol{d}_2|\left[\left(r_{px} + d_{1x} - \frac{d_{2x}}{3}\right) N_i(\boldsymbol{r}_p^{(2)}) + \left(r_{px} + d_{1x} + \frac{d_{2x}}{3}\right) N_i(\boldsymbol{r}_p^{(3)})\right]\boldsymbol{T} && (42)\\
\boldsymbol{f}^{(s)}_{ip,3} &= |\boldsymbol{d}_1|\left[\left(r_{px} + d_{2x} + \frac{d_{1x}}{3}\right) N_i(\boldsymbol{r}_p^{(3)}) + \left(r_{px} + d_{2x} - \frac{d_{1x}}{3}\right) N_i(\boldsymbol{r}_p^{(4)})\right]\boldsymbol{T} && (43)\\
\boldsymbol{f}^{(s)}_{ip,4} &= |\boldsymbol{d}_2|\left[\left(r_{px} - d_{1x} - \frac{d_{2x}}{3}\right) N_i(\boldsymbol{r}_p^{(1)}) + \left(r_{px} - d_{1x} + \frac{d_{2x}}{3}\right) N_i(\boldsymbol{r}_p^{(4)})\right]\boldsymbol{T} && (44)
\end{aligned}$$

This analysis applies to both axisymmetric uGIMP and CPDI, however, because uGIMP uses undeformed particle domains, the forces simplify by using $\boldsymbol{d}_1 = (l_p\Delta r/2, 0)$ and $\boldsymbol{d}_2 = (0, l_p\Delta z/2)$ to:

$$\begin{aligned}
\boldsymbol{f}^{(s)}_{ip,1} &= d_{1x}\left[\left(r_{px} - \frac{d_{1x}}{3}\right) N_i(\boldsymbol{r}_p^{(1)}) + \left(r_{px} + \frac{d_{1x}}{3}\right) N_i(\boldsymbol{r}_p^{(2)})\right]\boldsymbol{T} && (45)\\
\boldsymbol{f}^{(s)}_{ip,2} &= d_{2y}\left(r_{px} + d_{1x}\right)\left[N_i(\boldsymbol{r}_p^{(2)}) + N_i(\boldsymbol{r}_p^{(3)})\right]\boldsymbol{T} && (46)\\
\boldsymbol{f}^{(s)}_{ip,3} &= d_{1x}\left[\left(r_{px} + \frac{d_{1x}}{3}\right) N_i(\boldsymbol{r}_p^{(3)}) + \left(r_{px} - \frac{d_{1x}}{3}\right) N_i(\boldsymbol{r}_p^{(4)})\right]\boldsymbol{T} && (47)\\
\boldsymbol{f}^{(s)}_{ip,4} &= d_{2y}\left(r_{px} - d_{1x}\right)\left[N_i(\boldsymbol{r}_p^{(1)}) + N_i(\boldsymbol{r}_p^{(4)})\right]\boldsymbol{T} && (48)
\end{aligned}$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

Finally, if the particle crosses $r = 0$, the particle domain should be truncated before calculating surface forces. For CPDI, the same truncation method described above can be used. For uGIMP, if $r_{px} - d_{1x}$ is less than zero, change them to new values $r'_{px} = d'_{1x} = (r_{px} + d_{1x})/2$.

We are not aware of corresponding planar MPM results for traction forces in the literature, but they are easy to derive by the same methods using $dS = B|g'(\rho)|d\rho$ along the edge with B the thickness. For example, along edge 1, the planar result is:

$$f^{(s)}_{ip,1} = \int \boldsymbol{T} N_i(r,z)\, dS = \boldsymbol{T}|\boldsymbol{d}_1|B \int_{-1}^{1} \left(\sum_{\alpha=1}^{4} M_\alpha(\rho,-1) N_i(\boldsymbol{x}_p^{(\alpha)}) \right) d\rho \tag{49}$$

The results for all edges (and for both uGIMP and CPDI) are

$$\boldsymbol{f}^{(s)}_{ip,1} = |\boldsymbol{d}_1|B \left[N_i(\boldsymbol{r}_p^{(1)}) + N_i(\boldsymbol{r}_p^{(2)}) \right] \boldsymbol{T} \tag{50}$$

$$\boldsymbol{f}^{(s)}_{ip,2} = |\boldsymbol{d}_2|B \left[N_i(\boldsymbol{r}_p^{(2)}) + N_i(\boldsymbol{r}_p^{(3)}) \right] \boldsymbol{T} \tag{51}$$

$$\boldsymbol{f}^{(s)}_{ip,3} = |\boldsymbol{d}_1|B \left[N_i(\boldsymbol{r}_p^{(3)}) + N_i(\boldsymbol{r}_p^{(4)}) \right] \boldsymbol{T} \tag{52}$$

$$\boldsymbol{f}^{(s)}_{ip,4} = |\boldsymbol{d}_2|B \left[N_i(\boldsymbol{r}_p^{(1)}) + N_i(\boldsymbol{r}_p^{(4)}) \right] \boldsymbol{T} \tag{53}$$

The difference between the planar results and the axisymmetric results is that thickness, B, replaces all the coefficients in the axisymmetric results, which are radial positions along the traction edge. Further extension to 3D is straightforward by integrating over the six faces of the deformed particle domain.

The traction vector $\boldsymbol{T}$ is assumed constant over the edge during each time step. Furthermore, two options for finding the traction appear useful. First, the traction could be tied to global coordinate system. This form might correspond to macroscopically applied stress to an object by a rigid loading system. Alternatively, the traction could be tied to edge orientation. In this style, $\boldsymbol{T}$ would be calculated each time step from $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$, such as to be normal or tangential to the deformed edge. This form might correspond to an applied constant pressure to an object that rotates during a simulation.

The alternative to traction boundary conditions is to apply particle forces. Use of such forces, which are applied at the particle centroids, results in that force effectively being a distributed load over the particle domain. This approach leads to unphysical stress fluctuations in the region of the applied load. The traction approach described above remedies this problem. In planar analysis, constant particle forces for particles on the boundary approximates application of constant traction (albeit with edge artifacts). In axisymmetric calculations, the particle forces method needs forces per radian and thus those forces will depend on radial position when applied to a boundary that does not have r constant.

3.2. Explicit Cracks

Another advantage of MPM is it ability to model explicit cracks that use a particle-based definition to allow propagation in arbitrary directions. This extension of MPM, called CRAMP [9], requires dividing velocity fields into those above and below the crack by monitoring whether or not lines between nodes and particles cross the crack surface. That task is entirely

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

geometrical and therefore works in axisymmetric MPM without any modification from planar MPM.

Two crack mechanics options, however, do require modification for axisymmetry. First, MPM cracks can model cohesive zones by implementing traction laws based on crack opening displacements [13]. Cohesive laws are implemented in MPM by adding cohesive forces to the internal nodal forces, $\boldsymbol{f}_i^{int}$ [13]. The cohesive stress comes from the traction law, but it is that stress times local crack surface area that provides the contribution to internal force. In planar MPM, forces are applied to each segment of the crack and the required area is crack segment length times thickness. In axisymmetric MPM, the traction force needs to change to force per radian, which means the planar area is replaced by crack segment length times radial position of the node.

Second, MPM can implement fracture mechanics and crack propagation using J integral [14] methods [7, 8]. The J integral around an axisymmetric crack requires a different treatment than the cartesian J integral. Our literature search, however, did not find any fully developed results for axisymmetric J integral. Two different theoretical implementations are given in Bergkvist and Huong [15] and in Broberg [16]. Although these different methods theoretically give the same path independent result, they find it from different integrals. It remains to be demonstrated which is more effective in numerical calculations. Another limitation is that both these methods treat only radial cracks. The full implementation of J integral calculations for arbitrary crack orientation in axisymmetric problem for both MPM and finite element analysis is in need of additional development.

3.3. Multimaterial Contact Calculations and Imperfect Interfaces

MPM has proved well suited to problems involving massive amounts of contact because, in principal, it can handle contact physics without any need for special contact elements. The issue is discussed in [4, 5, 6, 17]. In brief, contact is handled by using multimaterial mode MPM where each material extrapolates to its own velocity field. Nodes interacting with a single material proceed by normal MPM methods, but nodes with more than one material use contact physics to adjust the nodal momenta and/or to add internal forces. Available methods can implement stick contact, frictional contact [4, 5, 6], or imperfect interface traction laws [17].

Most planar 2D contact tasks can be applied directly to axisymmetric MPM. The only exception is determination of the normal vector between two contacting surfaces, and experience has shown that accurate determination of the normal is crucial to accurate contact simulations [6]. The traditional approach to finding the normal is to use the mass gradient, or equivalently (and preferably when comparing gradients between materials with different densities) to use the volume gradient. This gradient can be extrapolated to nodes using shape functions gradients:

$$||\boldsymbol{n}_{i,a}||\boldsymbol{n}_{i,a} = -\nabla V_a(\boldsymbol{x}_i) = \sum_{p\in a} V_p \boldsymbol{G}_{ip} \tag{54}$$

where $\boldsymbol{n}_{i,a}$ is an unnormalized, outward directed normal for material a at node i and the sum is only for particles of material a. This method does not work in axisymmetric MPM because V_p depends on radial position, which causes $\nabla V_a(\boldsymbol{x}_i)$ to lose is connection to r-z plane surface normal. The solution is to realize the geometric nature of this calculation and base it in on area

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

gradient instead:

$$||\boldsymbol{n}_{i,a}||\boldsymbol{n}_{i,a} = -\nabla A_a(\boldsymbol{x}_i) = \sum_{p\in a} A_p \boldsymbol{G}_{ip} \tag{55}$$

where A_p is the deformed area of the particle domain in the r-z plane.

Two other refinements for axisymmetric contact are helpful. First, by symmetry, the area gradient in the r direction should be zero at $r = 0$, but that area gradient will always be non-zero by the above extrapolation method. The solution is to set the r component of $\boldsymbol{n}_{i,a}$ to zero for all nodes with $r_i = 0$. Similarly, some contact methods screen out non-contact nodes by ignoring contact unless the total volume at a node is close the the cell volume in the mesh. For axisymmetric MPM, this geometric result should switch from comparing volumes to comparing areas instead. Furthermore, the nodal area for all nodes at $r_i = 0$ should be doubled to account for the fact it is the the middle of a solid object and not on an edge. This same modification should be made in planar contact problems when they involve symmetry planes; the only difference is that axisymmetry always has a symmetry plane at $r = 0$.

The implementation of imperfect interfaces on cracks [18] or within multimaterial methods needs an additional adjustment for axisymmetric MPM when calculating the total interfacial contact area. The development of these axisymmetric methods are given in [17].

3.4. Conduction and Diffusion

MPM can simultaneously model heat conduction and liquid/gas diffusion and couple the results to mechanical strains through thermal or absorption expansion coefficients or temperature-dependent material properties. The algorithm extends easily to axisymmetric MPM. A weak-form analysis for conduction involves solving

$$\int_\Omega \left(\nabla\cdot \mathbf{k}\nabla T + q(\boldsymbol{x}) - \rho C_p \frac{\partial T}{\partial t}\right) w(\boldsymbol{x})\, dV \tag{56}$$

where T is temperature, $\mathbf{k}$ is conductivity tensor, $q(\boldsymbol{x})$ is heat source, ρ is density, C_p is constant-stress heat capacity, and $w(\boldsymbol{x})$ is a weighting function. Using the divergence theorem on the first term gives:

$$\int_{\partial\Omega} (w(\boldsymbol{x})\mathbf{k}\nabla T)\cdot \hat{n}\, dS = \int_\Omega \left(\nabla w(\boldsymbol{x})\cdot \mathbf{k}\nabla T + q(\boldsymbol{x})w(\boldsymbol{x}) - \rho C_p w(\boldsymbol{x})\frac{\partial T}{\partial t}\right) dV \tag{57}$$

Proceeding as in GIMP (*i.e.*, expanding $\boldsymbol{w}(\boldsymbol{x})$ in grid shape functions, expanding other terms in particle basis functions, making use of arbitrariness of $w(\boldsymbol{x})$, and using cylindrical volume integrals) [11] leads to:

$$M_i \frac{dT_i}{dt} = f_i^{(g)} + f_i^{(s)} + f_i^{(e)} \tag{58}$$

where

$$M_i = \sum_p m_p C_p^{(p)} S_{ip}, \quad T_i = \frac{1}{M_i}\sum_p m_p C_p^{(p)} T_p S_{ip} \tag{59}$$

$$f_i^{(g)} = -\sum_p V_p \mathbf{k}\nabla T_p \cdot \boldsymbol{G}_{ip}, \qquad f_i^{(s)} = \sum_p V_p q_p S_{ip}, \qquad f_i^{(e)} = \int_{\partial\Omega} N_i(\boldsymbol{x})\Phi\, dS \tag{60}$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

Here T_p is particle temperature, ∇T_p is particle temperature gradient that is calculated from T_i,

$$\nabla T_p = \sum_i T_i \boldsymbol{G}_{ip}, \tag{61}$$

and Φ is heat-flux boundary condition (*i.e.*, specified boundary value for $\mathbf{k}\nabla T \cdot \hat{n}$). The particle temperature is updated using

$$T_p^{(k+1)} = T_p^{(k)} + \Delta t \sum_i \frac{dT_i}{dt} S_{ip} \tag{62}$$

and importantly not by extrapolating T_i, because that causes numerical conduction. However, when using temperature in constitutive laws on particles, it is preferable to use an extrapolated T_i instead of T_p. T_i is a better representation of the temperature field, while T_p is prone to local fluctuations within grid cells.

The only changes needed for axisymmetry are to use axisymmetric S_{ip} and $\boldsymbol{G}_{ip}$, to use particle mass (m_p) and volume (V_p) on per-radian basis, and to evaluate flux boundary conditions as flux per radian on the cylindrical surface. This flux surface integral can be done using axisymmetric traction boundary condition methods described above by replacing vector traction, $\boldsymbol{T}$, with scalar flux, Φ. An axisymmetric GIMP analysis for diffusion is recovered by replacing $\mathbf{k}$ with the diffusion tensor, $\mathbf{D}$, heat source $q(\boldsymbol{x})$ with a concentration source, $m_p C_p^{(p)}$ with V_p, and temperature T with chemical potential, μ.

4. EXAMPLES

4.1. Uniform Radial Compression

A simple example is to pressurize a solid disk with initial radius R_0 in the radial direction while restraining it in the z direction. The resulting deformation gradient is constant throughout the disk:

$$\mathbf{F} = \begin{pmatrix} 1+\varepsilon_{rr} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1+\varepsilon_{rr} \end{pmatrix} \tag{63}$$

where $\varepsilon_{rr} = -\Delta R/R_0$ and ΔR is the amount of radial compression. Consider a Neohookean material with energy, W, defined by

$$W = \frac{K}{2}\left[\frac{1}{2}(J^2-1) - \ln J\right] + \frac{G}{2}\left(\bar{I}_1 - 3\right) \tag{64}$$

where $J = |\mathbf{F}| = (1+\varepsilon_{rr})^2$ is the relative volume, K is bulk modulus, G is shear modulus, and

$$\bar{I}_1 = \frac{B_{xx} + B_{yy} + B_{zz}}{J^{2/3}} \tag{65}$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

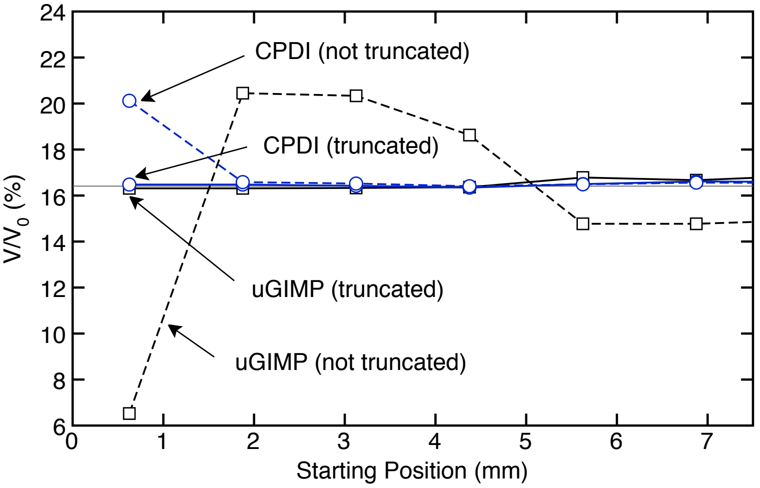

Figure 5. Relative volume (in %) in particles near the origin as a function of their initial radial particle position. The simulation was radial compression of a solid, flat disk with $\Delta R/R_0 = 0.6$. These calculations used axisymmetric shape functions that were truncated (as recommended) or not truncated at $r = 0$.

where $\mathbf{B} = \mathbf{F}\mathbf{F}^T$ is the left Cauchy-Green tensor. For this material, the exact solutions for stresses are:

$$\sigma_{rr} = \sigma_{\theta\theta} = \left(\frac{J-1}{J}\right)\left(\frac{K}{2}(J+1) + \frac{G_1}{3J^{2/3}}\right) \tag{66}$$

$$\sigma_{zz} = \left(\frac{J-1}{J}\right)\left(\frac{K}{2}(J+1) - \frac{2G_1}{3J^{2/3}}\right) \tag{67}$$

First, we considered MPM calculations with five possible shape functions — uGIMP and CPDI with truncated particle domains, uGIMP and CPDI without truncating particle domains, and classic MPM shape functions — and compressed slowly (compared to wave speed of the material) until $\varepsilon_{rr} = -0.6$ resulting in final relative volume of $J = 16\%$. This high compression provides a critical test of truncated *vs.* non-truncated shape functions near $r = 0$. The disk had $R_0 = 50$ mm, thickness of 10 mm, and the MPM grid cell size was 2.5×2.5 mm. The Neohookean material had $E = 2300$ MPa and $\nu = 0.33$ ($K = 2254.9$ MPa and $G = 864.662$ MPa). Compression was done at fixed velocity using rigid particles.

Figure 5 plots J, or relative volume, in particles near the origin as a function of their initial position. The only axisymmetric shape functions to correctly give J constant for all particles (the thin line is the mean J) are the truncated uGIMP and CPDI shape functions. If uGIMP shape functions are not truncated, the results are poor. Because CPDI approximately tracks deformed particle domain, the effects of truncation are smaller. Nevertheless, CPDI without truncation loses accuracy for the particle closest to the origin. No results for classic axisymmetric MPM shape functions are given, because they became unstable as many particles crossed element boundaries. GIMP was developed to solve this issue [11] and with these results now solves it in axisymmetric calculations as well.

Figures 2 and 3 suggest that axisymmetric S_{ip} and $\boldsymbol{G}_{ip}$ are not significantly different then planar shape functions except near $r = 0$. One is tempted to avoid the need for axisymmetric

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

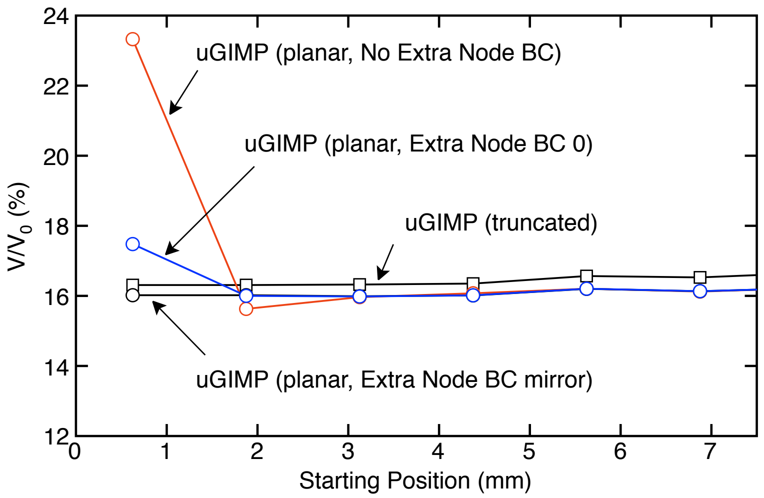

Figure 6. Relative volume (in %) in particles near the origin as a function of their initial particle position. The simulation was radial compression of a solid, flat disk with $\Delta R/R_0 = 0.6$.These calculations compared use of truncated, axisymmetric uGIMP shape functions (dsahed line) to planar shape functions (solid lines). The planar shape functions used three different methods for boundary conditions at $r = 0$.

GIMP shape functions by straightforward extension of classic axisymmetric MPM [12] to GIMP simply by replacing planar $N_i(r_p, z_p)$ and $\nabla N_i(r_p, z_p)$ shape functions from that work with planar S_{ip} and $\boldsymbol{G}_{ip}$ shape functions (we tried that ourselves before deriving new shape functions). Figure 6 compares the use of truncated, axisymmetric uGIMP formulation described here, with results for the same problem, using axisymmetric methods, but with planar shape functions. First, compare the curve labeled “uGIMP (planar, No Extra Node BC)” to the curve using axisymmetric shape functions (labeled “uGIMP (truncated)”). These curves had the same zero velocity boundary conditions at $r = 0$. The planar shape functions gave poor results near the origin while the new shape functions gave the correct constant value as a function of particle position.

Thus, simply using planar shape function instead of the axisymmetric shape function derived here in axisymmetric problems can cause large errors at the origin (see “uGIMP (planar, No Extra Node BC)” curve in Fig. 6). These errors, however, can be reduced by adding custom treatment of the $r = 0$ boundary conditions as illustrated by additional curves in Fig. 6. For all these simulations, the nodes at $r = 0$ had a boundary condition that set the r component of velocity to zero, while leaving the z component unchanged. The planar shape function results are differentiated by what is done with an extra layer of nodes located one cell away at $r = -\Delta r$. Although these nodes are naturally and correctly uninvolved when axisymmetric shape functions are used, they remain active for planar GIMP shape functions, because as particles approach domain boundaries, a portion of the particle domain will hang off of the defined computational domain. Without special logic, the numerical implementation of the shape functions will return nodes outside of the domain as being among those that interact with a particle. So called “extra nodes" (termed “ghost nodes" in some codes) are used to prevent accessing data that would otherwise not be stored in memory.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

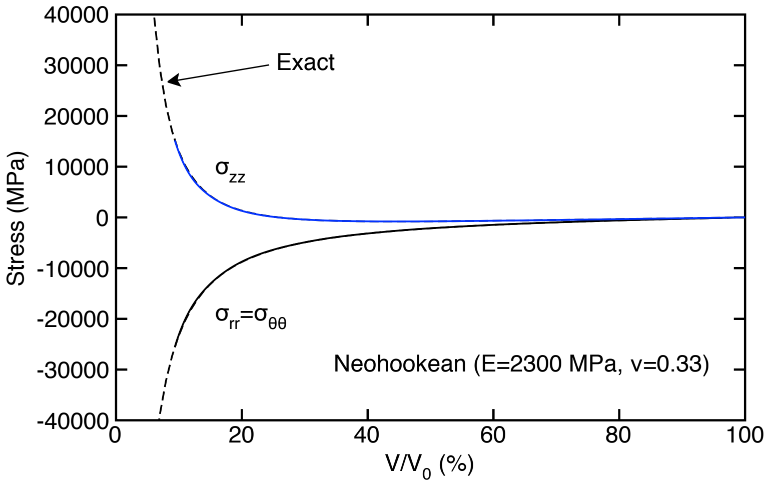

Figure 7. Average $\sigma_{rr} = \sigma_{\theta\theta}$ and σ_{zz} of all particles as a function of relative volume (in %) (solid lines) compared to exact solution (dashed lines). The simulation was radial compression of a solid, flat disk.

Three different treatments of the boundary conditions at those extra nodes were considered. The “No Extra Node BC" applies no special treatment to those nodes. The “Extra Node BC 0" case applies the same boundary condition to the extra nodes that is used for the $r = 0$ nodes. The “Extra Node BC mirror" case sets the r component of velocity on the extra node equal to the negative of the r component of velocity on the “mirrored” node at $r = \Delta r$ (leaving the z component unchanged). This last option is meant to mimic the value that node would have in a full three-dimensional simulation. What we observe is that, in the last option, which tends to keep the particle pushed farther away from r=0, the use of planar shape functions gave an acceptable result with constant level of compression. Note that because the axisymmetric formulation presented here truncates the GIMP domains at r=0, the extra nodes do not participate in the calculation, therefore the way boundary conditions are treated on those nodes is inconsequential; the results are always correct.

To verify axisymmetric material modeling, Fig. 7 plots $\sigma_{rr} = \sigma_{\theta\theta}$ and σ_{zz} as a function of J during radial compression. The dashed lines are the exact solutions. These calculations were for either uGIMP or CPDI with truncation and are plotted up to 84% compression. The uGIMP method broke down when compression reached 88%. The CPDI method began to break down at about 93% compression.

In summary, acceptable results in uniform compression are possible without using new shape functions, but only if great care is used in setting boundary conditions. Nevertheless, the truncated axisymmetric shape functions are recommended for at least four reasons. First, they are a natural result of re-deriving GIMP in an axisymmetric geometry and therefore are reasonably presumed to be more accurate and/or converge better. Although the effects are small in a simple example of uniform compression, larger effects might arise in unanticipated modeling geometries or in more complicated stress states. Second, the particle truncation method accounts well for $r = 0$ effects, which means ordinary zero-velocity boundary conditions at $r = 0$ are handled accurately for all particles without needing special treatment for the “ghost nodes” at $r = -\Delta r$. Third, when compared to prior work on

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

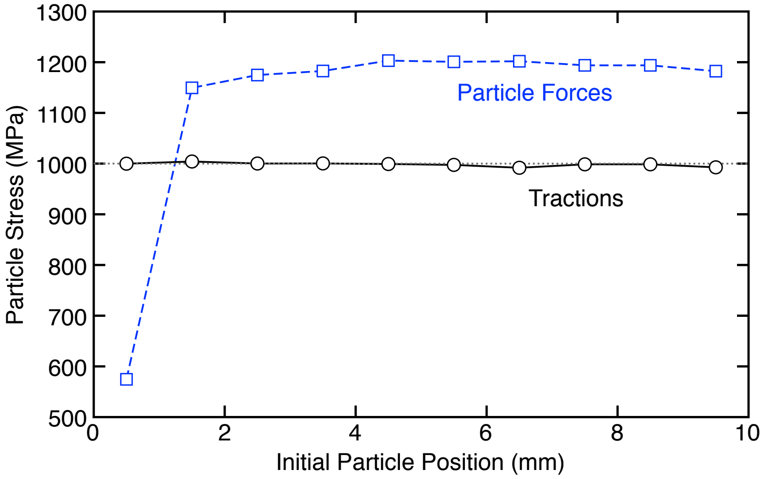

Figure 8. Particle σ_{zz} as function of their initial axial (z) distance to the loaded edge using either traction boundary conditions or particle force boundary conditions. The dotted, horiztonal line is the specified final stress of 1000 MPa.

axisymmetric MPM [12], this extension of the method to GIMP methods is essential when modeling large deformation because classic axisymmetric MPM methods are not robust. Fourth, any code that supports planar GIMP methods can easily add the extra terms needed for axisymmetric shape functions.

4.2. *Traction Boundary Conditions*

The simplest calculation to verify traction boundary conditions, and to focus on radial dependence in the forces, is to load a surface that varies in r, such as to end load a solid cylinder. This calculation used a cylinder of length 40 mm and radius 10 mm and MPM grid cell size was 2×2 mm. The $r = 0$ plane was fixed at zero r displacement and the cylinder bottom was fixed at zero z displacement. The particles on the top were loaded in tension by linearly increasing stress using either traction boundary conditions (on edge 3) or particle force conditions. The traction method calculated the forces as derived above. The particle force method applied force per radian that depended on particle location of $F_p(t) = \dot{\sigma} r_p l_p \Delta r t$. The Neohookean material had $E = 2300$ MPa and $\nu = 0.33$ ($K = 2254.9$ MPa and $G = 864.662$ MPa) and was loaded to final stress of 1000 MPa at slow displacement rate compared to wave speed of the material ($\sim 1\%$). The final strain was about 50%.

The axial stress, σ_{zz}, in particles as a function of initial distance from the loaded edge for traction or force boundary conditions is plotted in Fig. 8. These results confirm the new axisymmetric surface forces and show two clear advantages of surface tractions compared to particle force boundary conditions. First, the traction method maintains constant particle stress including the particle on the edge of the object. In contrast, particle force boundary conditions always have an artifact of low stress for the loaded particles. Second, the traction method applies the specified traction and maintains that traction when the particle deforms. In contrast, particle forces will apply a different stress if the surface area that is loaded changes during the calculation. In this simulation, the cylinder contracted in the radial direction due to Poisson

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

Table I. Material properties used for the penetration simulation as reported by Anderson *et al.* [21]. The material models used a Johnson-Cook flow rule [19] and a Mie-Grüneisen pressure response [20].

Property	Tungsten	Steel
A (GPa)	1.50696	1.50
B (GPa)	0.17664	0.569
C	0.016	0.003
n	0.12	0.22
m	1.0	1.17
T_m (K)	1723	1777
C_0 (m/s)	3980	3574
S	1.24	1.92
γ_0	1.54	1.69
Density (g/cc)	17.6	7.85
Shear Modulus (GPa)	124.0	77.3

effects and therefore the applied forces, which were based on initial dimensions, caused higher stresses. The ability of traction boundary conditions to maintain a specified stress requires CPDI because it evaluates surface forces on deformed edges. In contrast, uGIMP calculates surface forces using undeformed edges. The stresses in uGIMP would therefore increase like stresses due to particle forces, but would still have the advantage of constant stress on edge particles. The results for uGIMP are not given here because uGIMP calculations were not stable for the simulated 50% tensile strain. CPDI is required for large strain calculations and for constant stress boundary conditions.

4.3. Impact Penetration Example

To illustrate use of axisymmetric MPM in a complex stress state with most of the detail near the origin, we simulated a blunt nosed tungsten alloy cylinder (0.4 cm in diameter, 5.0 cm in length) impacting a high hard armor steel cylinder (4.0 cm in diameter, 4.95 cm long) at 1.70 km/s. Both materials were modeled as elastic-plastic, with a Johnson-Cook flow rule [19] and a Mie-Grüneisen pressure response [20]. Material parameters used are provided in Table I. These simulations are meant to represent the experimental configuration used by Anderson, *et al.* [21], for which data were collected for the nose and tail position at various times (using flash x-ray techniques). The numerical simulation depicted here used grid cells with $\Delta r = \Delta z = 0.25$ mm and 4 particles/cell were used to represent the material geometry. Simulations done at higher and lower resolutions (not shown) to confirm that the results are convergent with grid refinement.

Several views of the penetration process are provided in Fig. 9. The materials are colored according to equivalent stress. The nose and tail positions for the tungsten impactor were extracted from the numerical results and are plotted as smooth curves in Fig. 10. The simulation results have good agreement with experimental results from Anderson, *et al.* [21], which are plotted as symbols.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

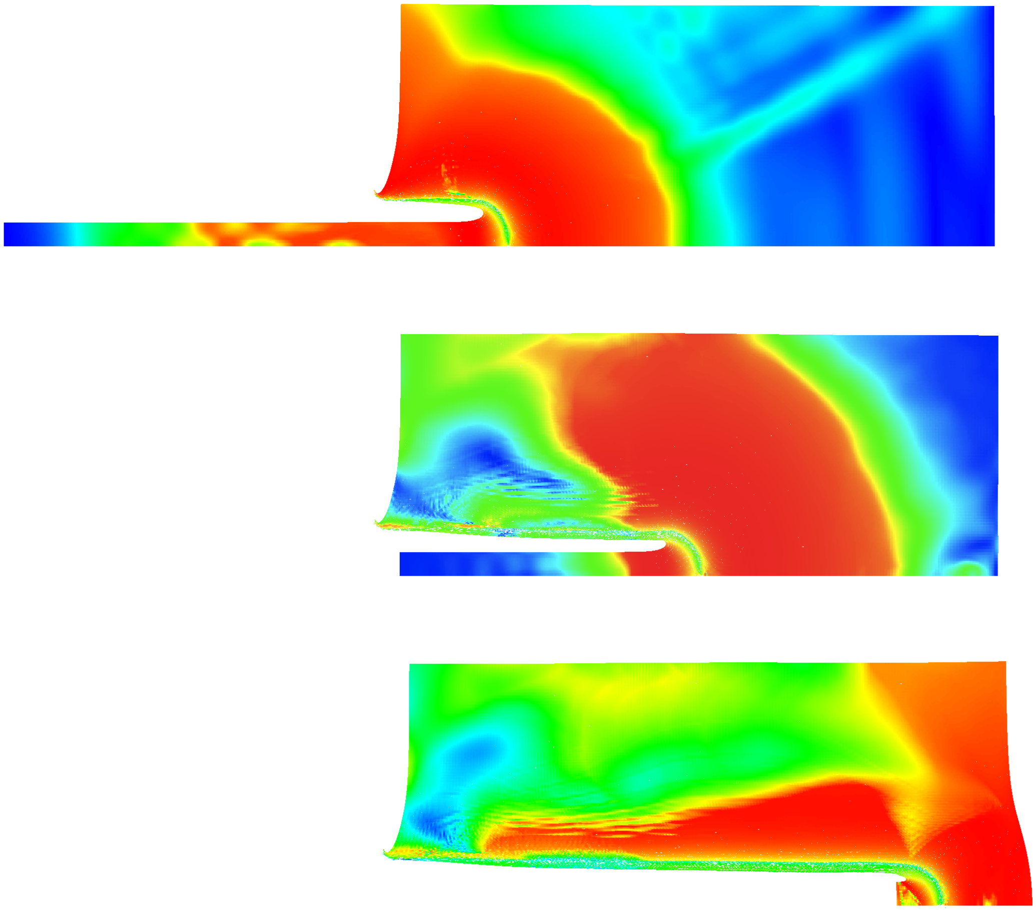

Figure 9. Several views of the MPM simulation for tungsten penetrating into high hard steel armor at 10 μs, 30 μs, and 60 μs after impact. The colors show show equivalent stress. The bottom edge is the $r = 0$ line of axisymmetry.

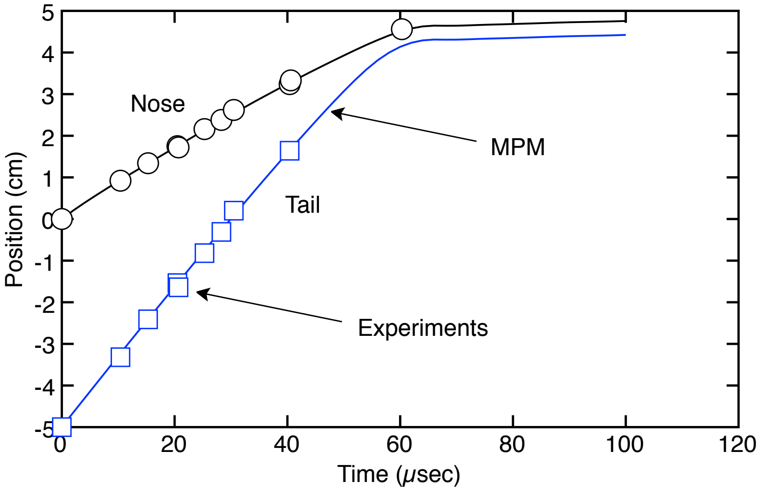

Figure 10. The position of the nose and tail of the tungsten impactor as a function of time. The symbols are experimental results from Anderson, *et al.* [21]. The lines are the axisymmetric MPM simulation.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792

A. CONCLUSION

Reformulation of GIMP for axisymmetry leads to new shape functions and gradients that are recommended for use in MPM axisymmetric simulations. The consequences of axisymmetry were considered for many MPM extensions, and all of them could be used in axisymmetric models with minor modifications, except for J integral calculations, which needs additional work. The following is a check list of changes needed to convert typical MPM codes to include an option for axisymmetric MPM:

1. Particle mass is mass per radian ($m_p = \rho A_p r_p$).
2. If the grid has any nodes at $r = 0$, apply zero velocity boundary conditions in the r direction.
3. Whenever shape functions or gradients are needed, find the axisymmetric shape functions (for uGIMP or CPDI) with domain truncation. Some results that depend on shape function gradients will also need T_{ip}. A convenient method in some codes is to store T_{ip} in the unused third component of a 3D gradient vector.
4. If including multimaterial contact, extrapolate the r-z plane area gradient to nodes and for nodes at $r = 0$ set the r component of that gradient to zero. This gradient is used to find contact normals. If contact methods use nodal areas for detecting contact, double the nodal area on $r = 0$ nodes.
5. On strain updates, find the hoop component of the deformation gradient using du_r/r in Eq. (14).
6. Each material used in simulations must support axisymmetric deformation gradients in their constitutive laws including updates for hoop stresses and strains.
7. When applying traction boundary conditions, calculate the axisymmetric form of the surface forces. When applying forces directly to particles, make sure the input is in force per radian and the forces correctly depend on position, if needed, to get desired stress along the edge.
8. When finding $\boldsymbol{f}_i^{(int)}$, include the hoop stress term in Eq. (3).
9. When updating particle positions, if any particle crosses $r = 0$, stop the calculations (or attempt a fix, such as to move to $r \geq 0$).

ACKNOWLEDGEMENTS

John Nairn was supported in part by an subcontract to Wasatch Molecular, Inc. within an SBIR grant from Office of Naval Research (contract N00014-13-M-0018) and by a grant from the National Science Foundation (award 1161305).

Copyright © 2015 John Wiley & Sons, Ltd.
Prepared using nmeauth.cls

APPENDIX A

The terms in shape functions and gradients along the z direction are identical to the corresponding terms in planar uGIMP along either direction. For reference, those terms are from [11] and given by:

$$S_z(\eta) = \begin{cases} \dfrac{(4-l_p)l_p - \eta^2}{4l_p} & |\eta| < l_p \\ \dfrac{2-|\eta|}{2} & l_p \le |\eta| < 2 - l_p \\ \dfrac{(2+l_p-|\eta|)^2}{8l_p} & 2 - l_p < |\eta| < 2 + l_p \\ 0 & \text{otherwise} \end{cases} \tag{68}$$

$$dS_z(\eta) = \begin{cases} -\dfrac{\eta}{2l_p} & |\eta| < l_p \\ -\text{sgn}(\eta)/2 & l_p \le \xi_p < 2 - l_p \\ -\text{sgn}(\eta)\dfrac{2+l_p-|\eta|}{4l_p} & 2 - l_p \le \xi_p < 2 + l_p \\ 0 & \text{otherwise} \end{cases} \tag{69}$$

REFERENCES

1. D. Sulsky, Z. Chen, H. L. Schreyer, A particle method for history-dependent materials, Comput. Methods Appl. Mech. Engrg. 118 (1994) 179–186.
2. D. Sulsky, S.-J. Zhou, H. L. Schreyer, Application of a particle-in-cell method to solid mechanics, Comput. Phys. Commun. 87 (1995) 236–252.
3. A. Sadeghirad, R. M. Brannon, J. Burghardt, A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations, Int. J. Num. Meth. Engng. 86 (12) (2011) 1435–1456.
4. S. G. Bardenhagen, J. U. Brackbill, Dynamic stress bridging in granular material, Journal of Applied Physics 83 (1998) 5732–5740.
5. S. G. Bardenhagen, J. E. Guilkey, K. M. Roessig, J. U. Brackbill, W. M. Witzel, J. C. Foster, An improved contact algorithm for the material point method and application to stress propagation in granular materials, Computer Modeling in Engineering & Sciences 2 (2001) 509–522.
6. V. Lemiale, A. Hurmane, J. A. Nairn, Material point method simulation of equal channel angular pressing involving large plastic strain and contact through sharp corners, Computer Modeling in Eng. & Sci. 70 (1) (2010) 41–66.
7. Y. Guo, J. A. Nairn, Calculation of J-integral and stress intensity factors using the material point method, Computer Modeling in Engineering & Sciences 6 (2004) 295–308.
8. Y. Guo, J. A. Nairn, Three-dimensional dynamic fracture analysis in the material point method, Computer Modeling in Engineering & Sciences 16 (2006) 141–156.
9. J. A. Nairn, Material point method calculations with explicit cracks, Computer Modeling in Engineering & Sciences 4 (2003) 649–664.
10. K. P. Ruggirello, S. C. Schumacher, A comparison of parallelization strategies for the material point method., in: 11th World Congress on Computational Mechanics, Barcelona, Spain, July 20-25, 2014.
11. S. G. Bardenhagen, E. M. Kober, The generalized interpolation material point method, Computer Modeling in Engineering & Sciences 5 (2004) 477–496.

Copyright © 2015 John Wiley & Sons, Ltd.

Prepared using nmeauth.cls
DOI: 10.1002/nme.4792

12. D. Sulsky, H. K. Schreyer, Axisymnmetric form of the material point method with applications to upsetting and Taylor impact problems, Comput. Methods. Appl. Mech. Engrg 139 (1996) 409–429.
13. J. A. Nairn, Analytical and numerical modeling of r curves for cracks with bridging zones, Int. J. Fract. 155 (2009) 167–181.
14. J. R. Rice, A path independent integral and the approximate analysis of strain concentration by notches and cracks, J. Applied Mech. June (1968) 379–386.
15. H. Bergkvist, G.-L. Huong, J-integral related quantities in axisymmetric cases, International Journal of Fracture 13 (1977) 556–558.
16. K. B. Broberg, Cracks and Fracture, Academic Press, London, 1999.
17. J. A. Nairn, Modeling of imperfect interfaces in the material point method using multimaterial methods, Computer Modeling in Eng. & Sci. 92 (3) (2013) 271–299.
18. J. A. Nairn, Numerical implementation of imperfect interfaces, Computational Materials Science 40 (2007) 525–536.
19. G. R. Johnson, W. H. Cook, A constitutive model and data for metals subjected to large strains, high strain rates and high, in: Proceedings of the 7th International Symposium on Ballistics, The Hague, The Netherlands, 1983, pp. 541–547.
20. M. L. Wilkens, Computer Simulation of Dynamic Phenomena, Springer-Verlag, New York, 1999.
21. C. E. Anderson, Jr., V. Hohler, J. D. Walker, A. J. Stilp, Time-resolved penetration of long rods into steel targets, Int. J. Impact Engng 16 (1995) 1–18.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/nme.4792