Analysis and reduction of quadrature errors in the material point method (MPM)

Michael Steffen, Robert M. Kirby*,† and Martin Berzins

*School of Computing and Scientific Computing and Imaging Institute, University of Utah,
Salt Lake City, UT 84112, U.S.A.*

SUMMARY

The material point method (MPM) has demonstrated itself as a computationally effective particle method for solving solid mechanics problems involving large deformations and/or fragmentation of structures, which are sometimes problematic for finite element methods (FEMs). However, similar to most methods that employ mixed Lagrangian (particle) and Eulerian strategies, analysis of the method is not straightforward. The lack of an analysis framework for MPM, as is found in FEMs, makes it challenging to explain anomalies found in its employment and makes it difficult to propose methodology improvements with predictable outcomes.

In this paper we present an analysis of the quadrature errors found in the computation of (material) internal force in MPM and use this analysis to direct proposed improvements. In particular, we demonstrate that lack of regularity in the grid functions used for representing the solution to the equations of motion can hamper spatial convergence of the method. We propose the use of a quadratic B-spline basis for representing solutions on the grid, and we demonstrate computationally and explain theoretically why such a small change can have a significant impact on the reduction in the internal force quadrature error (and corresponding 'grid crossing error') often experienced when using MPM. Copyright © 2008 John Wiley & Sons, Ltd.

1. INTRODUCTION

The material point method (MPM) [1, 2] is a particle method that represents a material as a collection of *material points* (hereafter referenced as particles) whose deformation is determined

*Correspondence to: Robert M. Kirby, School of Computing, University of Utah, 50 S. Central Campus Drive, Salt Lake City, UT 84112, U.S.A.
†E-mail: kirby@cs.utah.edu

by solving Newton's laws of motion for the internal force due to particle interaction. As with all methods of this form, the challenge (and novelty) of the method often comes from the means by which one defines approximations of differential and integral operators given the particle data. Although similar in nature to smoothed particle hydrodynamics (SPH) as used in fluid mechanics and to mesh-free (or meshless) methods as used in solid mechanics, MPM distinguishes itself as a mixed Lagrangian–Eulerian method, which utilizes a regular lattice 'background' grid for solving the equations of motion. MPM attempts to marry the best of both worlds—use of Lagrangian particles for representing materials (and its corresponding kinematic and dynamic properties) and use of an Eulerian grid upon which efficient numerical solvers can be built.

Although MPM has been shown to be extremely successful and robust in the simulation of a number of complicated engineering problems (see, for example, [3–5]), detailed analysis of the errors in MPM, even for simple problems, is lacking. This is not surprising, as MPM suffers the same challenges as experienced in almost all particle and mesh-free methods—finding a common framework or point of reference from which to define quantities such as truncation error and quadrature error. Such an analysis is not only needed as part of the classic numerical verification process advocated in the engineering sciences, but also for driving improvements of the methodology.

Consider particle grid crossing—one of the motivations given for the development of the generalized interpolation MPM (GIMP), a generalization of MPM [6]. Bardenhagen *et al.* demonstrate numerically that particles crossing grid cell boundaries cause unexpected computational artifacts, especially in the computation of internal forces. In a similar manner as is often done in the SPH literature, the problem was ascribed to the choice of particle representation. Not based on the direct analysis of MPM but on exploiting the analogies between MPM and other particle methodologies, GIMP introduces the idea of particle characteristic functions that have the effect of smoothing the impact of a particle's information on the underlying grid. The hope was that through this generalization one could eliminate these grid crossing artifacts. Although GIMP greatly reduced the impact of grid crossing, it did not eliminate it. The improved computational results of GIMP merit further investigation to explain why it provided tremendous benefit or to postulate why it did not completely solve the problem.

It is the thesis of this work that many of the numerical artifacts seen when employing MPM can be understood as being the result of the nature of the quadrature rules built into the methodology. This paper will examine the errors in internal force due to quadrature errors in the MPM framework. Detailed analysis will be performed using the standard piecewise-linear basis functions often used in MPM simulations. The hypothesis of smoother grid basis functions eliminating internal force errors will be tested by extending the analysis to both quadratic and cubic B-spline basis functions within the original MPM framework, showing that they do indeed reduce errors in internal forces. Finally, we will perform full dynamic simulations, showing that MPM with B-spline basis functions provide convergence properties not achievable with the standard piecewise-linear basis functions.

This paper is organized as follows. In Section 2 we provide background to help give context about where and how MPM fits into the family of particle and mesh-free methods. In Section 3 we provide an algorithmic overview of MPM with particular emphasis on the choice of the grid basis functions. In Section 4 we describe an interpretation of particle volume as the building block of our quadrature argument. Building upon our interpretation of particle volumes, we, in Section 5, provide an analysis of its impact on quadrature, in particular, how one might quantify quadrature errors. In Section 6 we present results that corroborate our perspective and our analysis and show

that MPM used with third-order B-splines provides a much improved method. In Section 7 we provide a summary of our findings, our conclusions and directions for future work.

2. BACKGROUND

In this section we provide some of the historic background supporting MPM. Recently, Brackbill provided an overview of a number of related particle methods [7] which is helpful in understanding where MPM fits in with the particle community. However, the purpose of this section is not to provide an exhaustive biography of MPM with comparison with all predecessors and methodological siblings. Rather, we will attempt to hit the salient points of comparison and contrast with respect to our thesis.

MPM is a mixed Lagrangian–Eulerian method with moving particles on a background grid. MPM [1, 2] descends from a long line of particle-in-cell (PIC) methods, specifically as a solid mechanics extension to the 'full particle' formulation of PIC called FLIP [8, 9]. These methods use similar approaches to SPH, namely to use an integral representation of field variables, or kernel approximation, when solving the governing equations. For example, in SPH, the evaluation of a field variable at a position $\mathbf{x}$ involves a weighted sum of particle data multiplied by a smoothing kernel function in the neighborhood of $\mathbf{x}$. The extent of the neighborhood, or the influence domain, is determined by the smoothing lengths of the particles. MPM similarly employs the idea of particles and particle smoothing kernel functions. However, unlike some particle methods where each particle represents a specific object, or a collection of objects, such as electrons or stars, MPM primarily uses particles and their associated volumes to partition a continuum. In MPM, particles are used to represent the Lagrangian state of a material. To solve the equations of motion, particle functions spread (or project) information to a background grid on which the equations of motion are solved. The background grid cells implicitly define influence domains, effectively eliminating the need for neighbor searches when a fixed Cartesian mesh is used. As mentioned earlier, GIMP [6] was developed as an extension to MPM, which modifies the type of particle functions used.

Although not derived directly from what is classically considered as mesh-free or meshless methods, MPM has been placed within the mesh-free community and has both many of the same advantages and many of the same challenges of other mesh-free methods [10]. Similar to many mesh-free methods, the primary partitioning of the material does not involve a polygonal tessellation (as in finite elements), but rather some alternative non-mesh-based unstructured representation. However, MPM utilizes a background mesh to perform differentiation, integration and solve equations of motion, similar to other mesh-free methods such as the element-free Galerkin method [11]. Although the background mesh is formally free to take any form, it is most often chosen for computational efficiency to be a Cartesian lattice (i.e. segments, quadrilaterals and hexahedra in 1D, 2D and 3D, respectively). These functions are used, in essence, as a means of discretizing the continuum equations, with the domain of these functions being an alternative representation of the deformed configuration of the material (*versus* the updated particles representing the deformed configuration). Nodal integration based on particle positions as is used in other particle methods such as PIC methods [12] is employed during the solution process. In this paper, we seek an understanding of the ramifications of this choice.

With all numerical methods, the accuracy of the method can depend highly on the accuracy of the numerical quadrature used. For some methods, such as the finite element method (FEM),

defining regions on which to perform integration and the act of numerical integration itself are well-established tasks. In mesh-free methods, however, information is often stored at seemingly 'random' positions, and hence the act of numerical integration becomes more difficult. The issue of numerical integration within mesh-free methods has received much attention and is well studied (e.g. see Dolbow and Belytschko [13] where the effect of mesh-free shape functions on integration error is reviewed).

The difficulty of various quadrature schemes, especially when using background grid cells as integration domains within Galerkin implementations, is one reason why collocation methods and nodal integration have been explored [14, 15]. It is these collocation schemes that particle methods such as PIC, SPH and MPM most resemble. Unfortunately, a detailed analysis of numerical integration errors within particle methods is limited in the literature. Vshivkov provides a detailed analysis of the interpolation errors in PIC [16], showing that the error depends on both the grid spacing and the mean number of particles in a cell. Recently, Qinlan *et al.* [17] looked at the truncation error when approximating spatial derivatives within SPH, showing that the error depends on smoothing length and the ratio of particle spacing to smoothing length. Both the particle-centric nature of the shape functions in both PIC and SPH, and the difference between shape functions in SPH and MPM, make the straightforward application of these results to MPM difficult. It is because of this that an analysis of integration errors specific to MPM is appropriate.

3. OVERVIEW OF MPM

In this section we provide an overview of the standard explicit MPM algorithm as presented by Sulsky *et al.* [2]. The method has five main steps: projection of particle data to the grid, evaluation of field variables at material points to be used in the constitutive model, constitutive model evaluation, evaluation of internal force on the grid, and advection of the Lagrangian particles. We will first give an algorithmic description of the process and then follow by a discussion of the choice of the grid basis functions one can employ in the method.

3.1. Standard MPM

The MPM procedure begins by discretizing the problem domain Ω with a set of material points or particles. The particles are assigned initial values including position, velocity, mass, volume and stress, denoted by $\mathbf{x}_p$, $\mathbf{v}_p$, m_p, V_p and $\boldsymbol{\sigma}_p$ (subscript index p is used to distinguish between particle values and an index of i for grid node values). Alternatively, instead of velocity and mass, momentum and mass density may be prescribed at the particle location, from which m_p and $\mathbf{v}_p$ can be calculated. Depending on the material being simulated, other quantities may be required at the material points, including items such as temperature, pressure, etc. The particles are then considered to exist within a computational grid, which for ease of computation is usually a regular Cartesian lattice. Figure 1 shows the setup for a typical 2D MPM problem.

At each time step t^k (all of the following quantities will be assumed to be at time t^k unless otherwise noted), the first step in the MPM computational cycle involves projecting (or spreading) data from the material points to the grid. Specifically we are interested in projecting particle mass and momentum to the grid to calculate mass and velocity at the grid nodes in the following
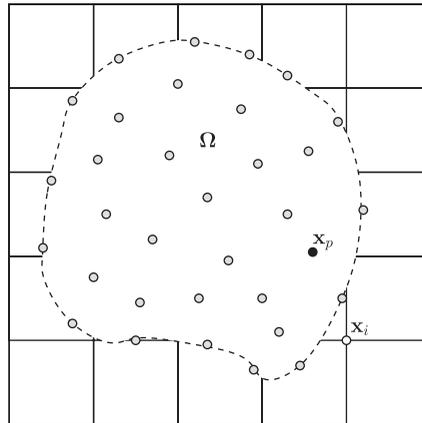
Figure 1. Typical 2D MPM problem setup. The dotted line represents the boundary of the simulated object Ω and each closed point represents a material point used to discretize Ω. The square mesh represents the background grid. Each square in the background grid is a grid cell and grid nodes are located at the corners of grid cells.

manner:

$$m_i = \sum_p \phi_{ip} m_p \tag{1}$$

$$\mathbf{v}_i = \left(\sum_p \phi_{ip} m_p \mathbf{v}_p \right) \bigg/ m_i \tag{2}$$

where $\phi_{ip} = \phi_i(\mathbf{x}_p)$ is a basis function centered at grid node i evaluated at the position $\mathbf{x}_p$. Note that Equation (1) represents the mass-lumped version of what Sulsky and Kaul [18] describe as the consistent mass matrix $M_{ij} = \sum_p \phi_{ip} \phi_{jp} m_p$. Next, velocity gradients are calculated at the particle positions as follows:

$$\nabla \mathbf{v}_p = \sum_i \nabla \phi_{ip} \mathbf{v}_i \tag{3}$$

where $\nabla \phi_{ip} = \nabla \phi_i(\mathbf{x}_p)$. The velocity gradient is then used in the evaluation of the constitutive model, resulting in an updated particle deformation gradient $\mathbf{F}_p$ and stress σ_p. Once σ is calculated, the internal forces at grid nodes can then be calculated as

$$\mathbf{f}_i^{\text{int}} = -\sum_p \sigma_p \cdot \nabla \phi_{ip} V_p \tag{4}$$

where $V_p = \det(\mathbf{F}_p) V_p^0$ denotes the volume of the particle voxel (in its deformed configuration). Combining the internal grid force with any external forces $\mathbf{f}_i^{\text{ext}}$, grid accelerations are then calculated as

$$\mathbf{a}_i = (\mathbf{f}_i^{\text{int}} + \mathbf{f}_i^{\text{ext}})/m_i \tag{5}$$

Next, grid velocities and positions are updated with an appropriate time-stepping scheme. Implicit time-stepping schemes exist for MPM [18–20]; however, we choose to use the explicit forward-Euler time discretization presented within the original MPM algorithm:

$$\mathbf{v}_i^{k+1} = \mathbf{v}_i^k + \mathbf{a}_i \Delta t \tag{6}$$

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \Delta t \tag{7}$$

Conceptually, when the grid nodes are advected, particles are then advected along with the grid. However, instead of explicitly moving the grid nodes as in Equation (7), the grid node location remains fixed and particles are advected using

$$\mathbf{v}_p^{k+1} = \mathbf{v}_p^k + \sum_i \phi_{ip} \mathbf{a}_i \Delta t \tag{8}$$

$$\mathbf{x}_p^{k+1} = \mathbf{x}_p^k + \sum_i \phi_{ip} \mathbf{v}_i^{k+1} \Delta t \tag{9}$$

Equations (1)–(9) outline one time step of MPM and are common to most material models and external force conditions. The calculation of $\boldsymbol{\sigma}_p$ involves a constitutive model evaluation and is specific for different material models. The 1D hyper-elastic constitutive model used in this paper involves updating the particle's history-dependent deformation gradient, F_p, using the velocity gradient from (3):

$$\tilde{F}_p = 1 + \nabla v_p^k \Delta t \tag{10}$$

$$F_p^{k+1} = \tilde{F}_p F_p^k \tag{11}$$

Particle stress is then calculated as follows:

$$\boldsymbol{\sigma}_p = E(F_p - 1) \tag{12}$$

where E is Young's modulus of the material. Multi-dimensional hyper-elastic constitutive models of the form found in the work of Simo and Hughes [21] have also been implemented for MPM, but will not be explicitly considered in this study.

Calculating $\mathbf{f}_i^{\text{ext}}$ is another problem-dependent procedure with several options. The first option is to calculate $\mathbf{f}_i^{\text{ext}}$ directly on the grid. This is easily done with body forces such as gravity where $\mathbf{f}_i^{\text{ext}} = m_i \mathbf{g}$. Another option is to calculate $\mathbf{f}_p^{\text{ext}}$ on the particles and interpolate to the grid through the grid shape functions:

$$\mathbf{f}_i^{\text{ext}} = \sum_p \phi_{ip} \mathbf{f}_p^{\text{ext}} \tag{13}$$

Moving forces, such as tractions on the end of an elastic bar, can be implemented by associating the forces with a finite set of 'end' particles and using Equation (13). Finally, for performance reasons, MPM is typically implemented using a fixed, equally spaced Cartesian lattice; however, nothing in the method requires the grid to be fixed. Moving grid nodes can be implemented to

track boundary forces, calculating $\mathbf{f}_i^{\mathrm{ext}}$ directly on the boundary nodes; however, implementing this for complex geometries in multiple dimensions is not trivial.

3.2. Choice of grid basis functions

In the above discussion, we purposely did not define precisely what grid basis function one should use for MPM. In the MPM algorithm outlined above, the choice of ϕ can be considered another option of the method.

Owing to their compact support, ease of computation, and partition of unity property, piecewise-linear basis functions are probably the most commonly used choice (with their natural extensions to bi-linear and tri-linear basis functions in 2D and 3D, respectively).

The notation used here for 1D piecewise-linear basis functions is

$$\phi_i(x) = \begin{cases} 1 + (x - x_i)/h, & x_i - h \leqslant x < x_i \\ 1 - (x - x_i)/h, & x_i \leqslant x \leqslant x_i + h \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

and the gradient of Equation (14) is given by

$$\nabla\phi_i(x) = \begin{cases} 1/h, & x_i - h \leqslant x < x_i \\ -1/h, & x_i \leqslant x \leqslant x_i + h \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where x_i denotes grid position i and h denotes a (uniform) grid spacing $h = x_{i+1} - x_i$.

It is worth noting that although these piecewise-linear basis functions are those used in many low-order FEMs, the discontinuous nature of $\nabla\phi$ is important in the analysis of MPM as it is a mixed Lagrangian–Eulerian method. Therefore, example zero-centered versions of Equations (14) and (15) are explicitly shown in Figure 2 (top). In FEMs, integration over the domain is decomposed into the sum of integrals over elements with quadrature points remaining fixed within elements. In MPM, however, particles act as integration points and are allowed to advect through the domain and across these discontinuities in $\nabla\phi$. The consequences of using particles to integrate discontinuous functions are explored in Section 5.

As Bardenhagen et al. describe in the development of GIMP [6], lack of regularity in $\nabla\phi_i$ is the root cause of grid cell crossing instabilities, and as can be seen in Figure 2, piecewise-linear basis functions are only C_0 continuous at nodal boundaries. Quadratic B-splines are then another logical choice of basis as they are C_1 continuous and only span three grid cells. Figure 2 (middle) shows an example set of quadratic B-spline basis functions for six equally spaced grid nodes (left) and their derivative functions (right).

To construct the basis function for grid node i, the knot vector $\{x_{i-3/2}, x_{i-1/2}, x_{i+1/2}, x_{i+3/2}\}$ is used, where x_i is the position of node i, and $x_{i+1/2} = \frac{1}{2}(x_i + x_{i+1})$. However, if node $i = 1$ is one node away from the left boundary x_0, the knot vector $\{x_0, x_{i-1/2}, x_{i+1/2}, x_{i+3/2}\}$ is used. Finally, the basis function for the border node $i = 0$ is defined by adding two B-splines defined by the knot vectors $\{x_i, x_i, x_{i+1/2}, x_{i+3/2}\}$ and $\{x_i, x_i, x_i, x_{i+1/2}\}$. A similar technique is used for the right
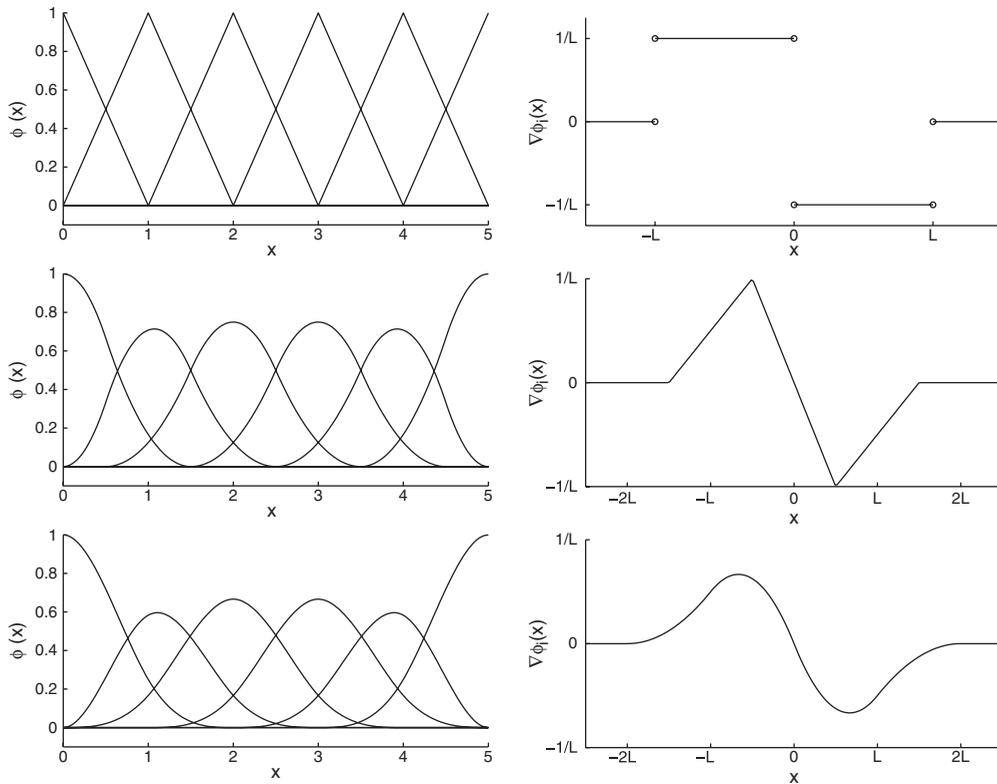
Figure 2. Example set of six equally spaced 1D basis functions (left column) and corresponding gradients for a selected grid node (right column) for piecewise-linear (top row), quadratic B-spline (middle row) and cubic B-spline (bottom row) basis functions.

boundary. As an example, an internal zero-centered quadratic B-spline has the form

$$
\phi(x) = \begin{cases}
\dfrac{1}{2h^2}x^2 + \dfrac{3}{2h}x + \dfrac{9}{8}, & -\dfrac{3}{2}h \leqslant x \leqslant -\dfrac{1}{2}h \\[2mm]
-\dfrac{1}{h^2}x^2 + \dfrac{3}{4}, & -\dfrac{1}{2}h \leqslant x \leqslant \dfrac{1}{2}h \\[2mm]
\dfrac{1}{2h^2}x^2 - \dfrac{3}{2h}x + \dfrac{9}{8}, & \dfrac{1}{2}h \leqslant x \leqslant \dfrac{3}{2}h \\[2mm]
0 & \text{otherwise}
\end{cases}
\tag{16}
$$

This method of constructing quadratic B-spline basis functions not only guarantees the peak of the basis functions to be centered over the grid nodes when nodes are equally spaced but also guarantees the partition of unity property, $\sum_i \phi_i(x) = 1$ for any $x \in \Omega$, required for the form of the implicit mass lumping often used in MPM. This method also generates interior basis functions that are zero on the boundary, i.e. $\phi_{i \neq 0}(x_0) = 0$. This boundary property combined with the partition of unity property combines such that a boundary basis function evaluates exactly to 1 on the

boundary, allowing for easier application of boundary conditions. As with the standard piecewise-linear basis functions, multi-dimensional basis functions can be created by tensor products of the 1D constructs.

Cubic B-splines may also be used and are, perhaps, somewhat more intuitive as the nodal positions are the values used in the knot vectors. Cubic B-splines are C_2 continuous and span four grid cells in each dimension. The knot vector $\{x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$ is used for internal nodes. The vector $\{x_{i-1}, x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$ is used when node i is one node away from the left boundary. When node i is the left boundary, the basis function is defined by adding the two B-splines defined by the vectors $\{x_i, x_i, x_i, x_{i+1}, x_{i+2}\}$ and $\{x_i, x_i, x_i, x_i, x_{i+1}\}$. Again, the construction of right boundary basis functions is similar. An internal zero-centered cubic B-spline has the following form:

$$
\phi(x) = \begin{cases}
\dfrac{1}{6h^3}x^3 + \dfrac{1}{h^2}x^2 + \dfrac{2}{h}x + \dfrac{4}{3}, & -2h \leqslant x \leqslant -h \\[2mm]
-\dfrac{1}{2h^3}x^3 - \dfrac{1}{h^2}x^2 + \dfrac{4}{3}, & -h \leqslant x \leqslant 0 \\[2mm]
\dfrac{1}{2h^3}x^3 - \dfrac{1}{h^2}x^2 + \dfrac{4}{3}, & 0 \leqslant x \leqslant h \\[2mm]
-\dfrac{1}{6h^3}x^3 + \dfrac{1}{h^2}x^2 - \dfrac{2}{h}x + \dfrac{4}{3}, & h \leqslant x \leqslant 2h \\[2mm]
0 & \text{otherwise}
\end{cases}
\tag{17}
$$

Figure 2 (bottom) shows an example set of cubic B-spline basis functions for six equally spaced grid nodes (left) and their derivative functions (right). Multi-dimensional B-spline basis functions are created by taking the tensor product of 1D basis functions. These multi-dimensional spline basis functions will have a rectilinear footprint on the grid and are not the same as the radial spline basis functions used in other mesh-free methods, such as SPH.

4. INTERPRETATION OF PARTICLE VOLUME

It is our contention that understanding how particle volumes are handled in MPM is necessary for developing a coherent analysis of the method. Every MPM simulation is initialized by discretizing the problem domain Ω with a set of particles. The exact method of discretization can vary depending on the situation, but normally consists of placing particles so that one obtains coverage of the material configuration. One example of a variation might be whether particles reside on the trace of the material configuration or not (which might be preferable for handling boundary conditions).

It is often convenient to imagine partitioning the material frame Ω into a set of initial voxels Ω_p^0 (we use the term voxel to denote the volumetric subset of the domain and reserve the use of the term volume to denote the scalar quantity describing the integral over a voxel) such that $\Omega = \bigcup_p \Omega_p^0$. The superscript '0' is used to emphasize that this occurs at a time level zero (i.e. in the material or reference configuration). The concept of a 'particle' in MPM is that of one of these voxels; however, the geometric information of the voxel itself is normally not maintained. A position $x_p \in \Omega_p^0$ (usually, but not always, consisting of the geometric centroid of the voxel) and volume $V_p^0 = \int_{\Omega_p^0} d\Omega$ are held by each particle. With each time step of the MPM algorithm
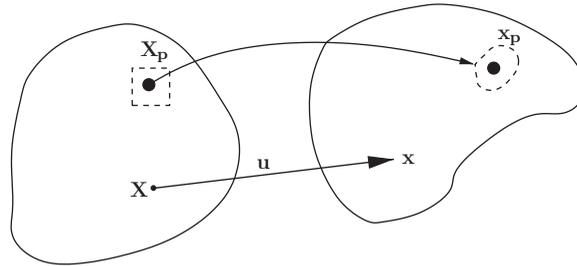
Figure 3. Reference (left) and deformed (right) configurations. The dotted lines represent the voxel associated with a particle in both the reference and deformed configurations. Note that the voxel of a particle does not maintain its shape in the deformed configuration, but space can still be tiled.

presented in Section 3, a particle's position and volume are updated. As the material deforms, the voxels tacitly deform as denoted in Figure 3. It is assumed that, at any time level, the deformed configuration of the material is represented by the union of the deformed voxels and that the sum of the volumes equals the volume of the deformed material.

Integration within MPM relies upon a particle's position and volume; it is used directly in calculating $\mathbf{f}_i^p$ in Equation (4) and is used tacitly in the mass projection as the mass at a grid node i is given by the following relation:

$$m_i = \int_\Omega \rho(x)\phi_i(x)\,\mathrm{d}\Omega \approx \sum_p \rho_p \phi_{ip} V_p \approx \sum_p \frac{1}{V_p} m_p \phi_{ip} V_p = \sum_p m_p \phi_{ip} \tag{18}$$

It is important to appreciate that the choice within MPM to only maintain a particle's volume and a single sample point (particle position) dictates the quadrature approximation properties of the method. MPM effectively employs, in the worst case, the first-order Riemann integration of field quantities. In the 1D case where the voxel consists of a line segment of length L and the sample point can be maintained at the center of the interval, the form of the integration reduces to the familiar midpoint rule. In the multi-dimensional case, it is more difficult to show that one can do better than first order if one only monitors volume and a single position, as the general shape of the voxel is only constrained by the laws of motion and the sampling point is not required to be maintained at the geometric centroid. *Much of the anomalous behavior exhibited by MPM can be attributed to the quadrature approximation properties of the method.* In fact, many of the proposed improvements to MPM either explicitly or tacitly attempt to control and improve MPM's quadrature behavior.

For example, Bardenhagen and Kober [6] try to address the problem of tracing deformed particle voxels with contiguous particle GIMP by using updated volumes in the weighting function $\overline{S}_{ip}$. This is used only in 1D and assumes space remains tiled with the updated set of Ω_p, x_p is in the center of the update voxel Ω_p, and the width of Ω_p is given by the updated particle volume V_p. This technique still uses information at the particle position to approximate the deformed shape of Ω_p.

Recently, Ma *et al.* [22] have modified GIMP to approximate the deformed state of each Ω_p, solving the problem of tiling space by placing massless tracking particles at the corner of the initial square voxels. These particles are advected with the grid velocity and are used to define the deformed voxel shape.

One interpretation of both of these previous efforts is that both have focused on improving the means of computing (or maintaining) the measure of integration, that is, attempting to more faithfully represent the voxel. Both, however, accomplish this at a computational cost which degrades the raw efficiency one can gain from the algorithm as presented in Section 3. In this study, we have taken an alternative view—to acknowledge the errors introduced by the quadrature employed in the current MPM algorithm, to attempt to understand the different contributing factors in that error, and to modify the integrand to help minimize the impact of the error. In particular, our focus is on the impact of changing the grid basis functions used within MPM in a way that reduces the quadrature error and provides consistent convergence results. In the following section, we present our analysis and our suggested improvements based on our findings.

5. ANALYSIS

Most of the grid values in MPM are calculated as approximations to the mass-lumped L^2 projections of data onto the grid basis functions or the gradients of the basis functions as in Equation (1). For example, if a field function $g(x)$ existed over the domain Ω, the values of g_i and ∇g_i at grid node i would be calculated as

$$g_i = \int_\Omega g(x)\phi_i(x)\,\mathrm{d}\Omega \approx \sum_p g_p \phi_{ip} V_p \qquad (19)$$

$$\nabla g_i = -\int_\Omega g(x)\nabla\phi_i(x)\,\mathrm{d}\Omega \approx -\sum_p g_p \nabla\phi_{ip} V_p \qquad (20)$$

where g_p is a sample of $g(x)$ at the particle position x_p, and V_p is the particle volume. The function g may be of the form $g(x)=f(x)\rho(x)$, with ρ the density, leading to a mass weighted projection of f as in (2) with $f_i=g_i/m_i$. If $g(x)$ was a vector-valued function, the divergence of $g(x)$ at a grid node would be calculated as follows:

$$(\nabla\cdot g)_i = -\int_\Omega g(x)\cdot\nabla\phi_i(x)\,\mathrm{d}\Omega \approx -\sum_p g_p\cdot\nabla\phi_{ip} V_p \qquad (21)$$

Note that in 1D equations (19)–(21) all have the form $\int_\Omega f(x)\,\mathrm{d}x \approx \sum_p f(x_p)V_p$, that is, they represent quadrature approximations of the integral. In this section, we discuss the case of 1D MPM in which we track the interval length L and maintain the sample point (*particle position*) at the centroid of the voxel. In this case, the quadrature in MPM reduces to the midpoint rule.

As a review, the midpoint rule for approximating the integral of $f(x)$ is typically expressed as $\int_\Omega f(x)\,\mathrm{d}x \approx h\sum_{i=1}^N f(x_i)$ where the domain Ω has been subdivided into N regions of size h and x_i is located in the center of region i. The midpoint rule, however, does not require each region to be of the same size. If the domain is divided into N regions with individual sizes h_i, the midpoint rule with uneven spacing is expressed as $\int_\Omega f(x)\,\mathrm{d}x \approx \sum_{i=1}^N f(x_i)h_i$, where again x_i is located in the center of region i.

It should be clear that if x_p is located in the center of the voxel defined by the volume V_p, and if the set of particle voxels tile the domain (or at least tile the non-zero regions of the function being
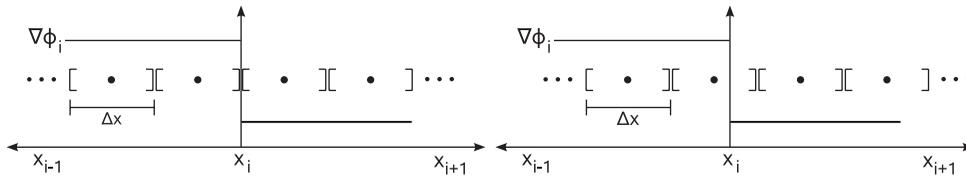
Figure 4. Particle configuration at time t (left) respects discontinuities in the gradients of piecewise-linear basis functions, allowing for exact integration using the midpoint rule. Advected particles at time $t+\Delta t$ (right) no longer respect the discontinuities, leading to quadrature errors.

integrated), then the MPM approximation to the integrals in Equations (19)–(21) is equivalent to a midpoint rule with uneven spacing. There is a problem, however, with applying the standard midpoint rule error analysis to this problem. The standard error analysis assumes continuity of f in each interval i. Depending on the choice of basis functions ϕ_i, the functions being integrated will not satisfy this continuity condition over the entire domain Ω. Integrating discontinuous functions with the midpoint rule is valid when the division of Ω into regions respects these discontinuities. In finite elements, for example, discontinuities occur at element boundaries; however, integration is always performed over individual elements, and thus these discontinuities are respected in FEM integration schemes. In MPM, however, the particle voxels will not, in general, respect these discontinuities for all time as particles advect through the domain. Figure 4 shows how a particle configuration may respect spatial discontinuities caused by the gradient of basis functions at time t, but will not respect at time $t+\Delta t$ once the particles have advected through the domain. Therefore, one way to understand the integration errors in MPM is to understand the errors in using the midpoint rule when integrating across discontinuities.

In this section, we first lay out the simple test problem presented in [6] for understanding errors in internal force computation of MPM. We will then examine the interplay between the midpoint rule nature of MPM quadrature and the choice of the grid basis functions, in particular, examining the commonly used piecewise-linear functions as well as the B-splines we introduced in Section 3.2.

5.1. Uniformly stressed body in MPM

In MPM, the internal force calculated by Equation (4) is an approximation to

$$f_i^{\text{int}} = - \int_\Omega \sigma(x) \cdot \nabla \phi_i(x)\, \mathrm{d}\Omega \approx - \sum_p \sigma_p \cdot \nabla \phi_{ip} V_p \tag{22}$$

which is similar in form to Equation (21). The stress field $\sigma(x)$ can in general take any form, making error analysis difficult. One form of $\sigma(x)$ that allows for easy analysis is the case of a uniformly stressed body with constant particle spacing Δx, where $\sigma_p = \sigma$ for all particles. Bardenhagen and Kober [6] note that force imbalances can develop with uniformly stressed bodies when different numbers of particles are in adjacent cells. Since MPM can be thought of as using a midpoint-rule approximation to expressions such as Equation (22), the errors in the internal force calculation can be analyzed by looking at errors in midpoint integration when integrating across discontinuities.

For this uniformly stressed body problem, the error in internal force can be evaluated as

$$E_{\text{f}} = \int_{\Omega} \sigma(x) \cdot \nabla \phi_i \, \mathrm{d}\Omega - \sum_p \sigma_p \cdot \nabla \phi_{ip} V_p = \sigma \cdot \left[\int_{\Omega} \nabla \phi_i \, \mathrm{d}\Omega - \Delta x \sum_p \nabla \phi_{ip} \right] \tag{23}$$

The bracketed term in Equation (23) not only corresponds to the error in internal force but also represents the error in integrating $\nabla \phi_i$ using the midpoint rule. Since we are using piecewise-polynomial basis functions, $\nabla \phi_i$ is also a piecewise polynomial and the internal force errors can be analyzed by looking at the midpoint integration errors present in Equation (23). We will now examine those errors for the three basis function choices presented in Section 3.2.

5.2. Piecewise-linear basis functions

Consider the scenario depicted in Figure 4 where the particle configuration does not respect discontinuities in the underlying integrand. Figure 5 shows this scenario in more detail, focusing on the particle overlapping the discontinuity. When a background grid with a cell width of h is discretized using piecewise-linear basis functions, $\nabla \phi$ is piecewise constant and discontinuities in $\nabla \phi$ occur at $-h$, 0, and h, with jumps in the polynomial's leading coefficient (see Equation (15)) of $1/h$, $-2/h$ and $1/h$, respectively. The midpoint integration error from integrating across a discontinuity in a piecewise-constant function is shown in detail in Appendix A for the case of the above-described uniformly stressed body. As shown in Equation (A6), the maximum error due to integrating over a discontinuity is given by $E_{\text{jump}} = C_1 [[\phi'(0)]] \Delta x$, where $[[\cdot]]$ denotes the jump condition, Δx is the particle width, and C_1 is a constant depending on the polynomial. With piecewise-linear basis functions used in MPM, the coefficient C_1 for integrating $\nabla \phi$ is $\frac{1}{2}$.

For particles not crossing over a discontinuity, there will be no error contribution since the midpoint rule can integrate constant functions exactly. Therefore, the only integration intervals contributing to the total error are those that cross the discontinuities. Substituting these individual E_{jump} errors into the bracketed term in Equation (23) leads to an upper bound on the total force error E_{f} of

$$E_{\text{total}} = \sigma \left[\frac{1}{2} \left(\frac{1}{h} + \frac{2}{h} + \frac{1}{h} \right) \Delta x \right] = 2\sigma \frac{\Delta x}{h} \tag{24}$$

The previous analysis considered only the magnitude of the jump when calculating an upper bound on the error. If the sign of the jump is taken into account, the maximum positive error occurs when integration over the discontinuity at 0 is respected by the particle distribution (i.e. no particles cross the discontinuity at 0), but the discontinuities at $-h$ and h are not. Similarly,
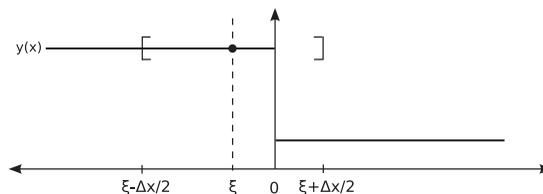

Figure 5. Piecewise-constant function with a midpoint region spanning a discontinuity at $x = 0$. This is a representative of a particle of width Δx centered at ξ whose voxel, Ω_p, crosses a grid cell boundary.
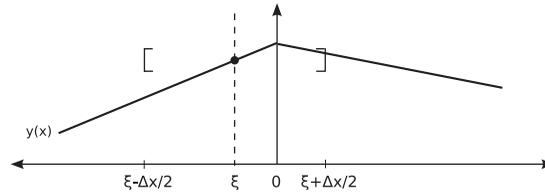
Figure 6. Piecewise-linear function with a midpoint region spanning a discontinuity in y' at $x=0$. This is a representative of a particle of width Δx centered at ξ whose voxel, Ω_p, crosses a grid cell boundary.

the maximum negative error occurs when the discontinuities at $-h$ and h are respected, but the discontinuity at 0 is not. These cases lead to errors of

$$E_{\text{total}} = \pm \sigma \frac{\Delta x}{h} \tag{25}$$

5.3. Quadratic B-spline basis functions

Consider the scenario depicted in Figure 6. For quadratic B-spline basis functions, $\nabla \phi$ is piecewise-linear, and discontinuities in $\nabla^2 \phi$ at $-3h/2$, $-h/2$, $h/2$ and $3h/2$, with jumps in the second derivative (from Equation (16)) of $1/h^2$, $-3/h^2$, $3/h^2$ and $-1/h^2$, respectively. The midpoint integration error from integrating across a discontinuity in a piecewise-linear function is shown in detail in Appendix A. As shown in Equation (A14), the maximum error due to integrating over a discontinuity is given by $E_{\text{jump}} = C_2 [[\phi''(0)]] \Delta x^2$, where $[[\cdot]]$ denotes the jump, Δx is the particle width, and C_2 is a constant depending on the polynomial. With the quadratic B-spline basis functions used in MPM, the coefficient C_2 for integrating $\nabla \phi$ is $\frac{1}{8}$.

Again, for particles not crossing over a discontinuity, there will be no error contribution since the midpoint rule can integrate linear functions exactly. Therefore, the only integration intervals contributing to the total error are those crossing discontinuities. Substituting these individual E_{jump} errors into the bracketed term in Equation (23) leads to an upper bound on the total force error E_{f} of

$$E_{\text{total}} = \sigma \left[\frac{1}{8} \left(\frac{1}{h^2} + \frac{3}{h^2} + \frac{3}{h^2} + \frac{1}{h^2} \right) \Delta x^2 \right] = \sigma \frac{\Delta x^2}{h^2} \tag{26}$$

Taking the signs of the jump in second derivatives into consideration, the maximum positive error occurs when the particle distribution is respectful of discontinuities when integrating over the discontinuities at $-h/2$ and $3h/2$ (those corresponding to a negative a^*), but not respectful at $-3h/2$ and $h/2$. The maximum negative error occurs when the opposite is true. Once again, these cases lead to errors, which are half of the maximum error calculated using the magnitude of the jump, and are given by

$$E_{\text{total}} = \pm \sigma \frac{\Delta x^2}{2h^2} \tag{27}$$

5.4. Cubic B-spline basis functions

Consider the scenario depicted in Figure 7. For cubic B-splines, $\nabla \phi$ is piecewise-quadratic and discontinuities in $\nabla^3 \phi$ occur at $-2h, -h, 0, h$ and $2h$, with jumps in the third derivative (see
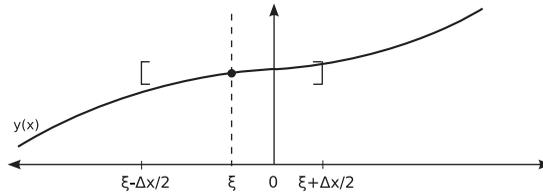
Figure 7. Piecewise-quadratic function with a midpoint region spanning a discontinuity in y'' at $x = 0$.

Equation (17)) of $1/2h^3$, $-2/h^3$, $3/h^3$, $-2/h^3$ and $1/2h^3$, respectively. The midpoint integration error from integrating across a discontinuity in a piecewise-quadratic function is shown in detail in Appendix A. As shown in A.3, the maximum error due to integrating over a discontinuity is given by $E_{\text{jump}} = C_3[[\phi'''(0)]]\Delta x^3$, where $[[\cdot]]$ denotes the jump, Δx is the particle width and C_3 is a constant depending on the polynomial. With the cubic B-spline basis functions used in MPM, the coefficient C_3 for integrating $\nabla \phi$ is $\frac{1}{24}$.

The total maximum error from integrating across discontinuities is then given by

$$E_{\text{total}} = \sigma \left[\frac{1}{24} \left(\frac{1}{2h^3} + \frac{2}{h^3} + \frac{3}{h^3} + \frac{2}{h^3} + \frac{1}{2h^3} \right) \Delta x^3 \right] = \sigma \frac{\Delta x^3}{3h^3} \tag{28}$$

However, $\nabla \phi_i$ is quadratic between the discontinuities and the midpoint rule cannot exactly integrate quadratic functions. As a review, if $f \in C^2[a, b]$, then for some μ in (a, b), the composite midpoint error with subintervals of size Δx is given by [23]

$$E = \frac{(b-a)}{24} \Delta x^2 f''(\mu) \tag{29}$$

For our piecewise-cubic ϕ_i, the value of the midpoint error for integrating $\nabla \phi_i$ in the four separate regions (see Equation (17)) is given by

$$E = \frac{1}{24} \Delta x^2 [h_1 \phi'''(\mu_1) + h_2 \phi'''(\mu_2) + h_3 \phi'''(\mu_3) + h_4 \phi'''(\mu_4)]$$

$$= \frac{1}{24} \Delta x^2 \left[h_1 \frac{1}{h^3} - h_2 \frac{3}{h^3} + h_3 \frac{3}{h^3} - h_4 \frac{1}{h^3} \right] \tag{30}$$

where h_1, h_2, etc. are the sizes of the integration intervals in the different regions after subintervals crossing the discontinuities have been removed. There exist many arrangements of particles such that $h_1 = h_2 = h_3 = h_4$ in which case the bracketed term in Equation (30) goes to zero, leaving Equation (28) as the only source of error. In general, however, the total error from integrating $\nabla \phi_i$ is the sum of Equations (28) and (30), and since h_1, h_2, h_3 and h_4 are $\mathcal{O}(h)$ the total error is $\mathcal{O}(\sigma \Delta x^2/h^2)$.

We see the internal force error improves when using quadratic B-spline basis functions instead of standard piecewise-linear basis functions. The quadratic B-splines have an error of $\mathcal{O}(\Delta x^2/h^2)$, whereas the piecewise-linear basis functions have an error of $\mathcal{O}(\Delta x/h)$. Using even higher-order splines, such as cubic B-splines, may give further improvement for some special particle configurations, but in general the composite midpoint integration error limits the error to the same order as with quadratic B-splines.

Brackbill [7] makes a similar observation concerning PIC methods when, building upon theoretical results presented by Vshivkov [16], he states that the error of the PIC method is bounded by

$$\varepsilon \leqslant C_1 \left(\frac{\delta}{h} \right)^2 + C_2 h^2 \tag{31}$$

where δ is the particle spacing, h is the mesh spacing and C_1 and C_2 are constants depending only on the smoothness of the data. Here, the quantity δ/h is a measure of the inverse of the number of particles per cell (PPC).

This result follows from Vshivkov's earlier analysis [16] where he calculates the error, δ_k, in the charge density at node k as calculated with PIC. One would expect this error to be analogous to measuring the error in the projection of particle information to the grid in MPM (such as the projection of mass) since the piecewise-linear mesh kernel functions that Vshivkov assumes in his analysis are the same as the grid basis functions used in standard MPM. His result states that the error is bounded by

$$\delta_k \leqslant \left(\frac{3\rho_{av}^2}{2\rho_{min}} + h \frac{\rho_{av}^2 \rho_{max}}{6\rho_{min}^3} \left| \frac{\partial \rho}{\partial x} \right|_{max} \right) \frac{1}{N^2} + \frac{h^2}{12} \left| \frac{\partial^2 \rho}{\partial x^2} \right|_{max} \tag{32}$$

where N is the average number of particles in a cell. The first term on the right-hand side relates to the 'quadrature' error as a consequence of number of particles and grid spacing, and the second term relates to the 'approximation' error as a consequence of grid spacing (and tacitly the choice of basis functions).

These results for PIC demonstrate the interplay between approximation error (based on the choice of the basis functions) and the quadrature error—results that are consistent with and indeed motivated this study.

6. RESULTS

In this section we now attempt to use the perspective provided in Section 4 and analysis provided in Section 5 to explain the common test cases presented in the MPM literature.

6.1. Uniformly stressed body

We first begin by revisiting the uniformly stressed example mentioned in the previous section. First, we must present a algorithmic way of setting up the problem. Consider the diagram given in Figure 8. To describe any arrangement of uniformly spaced particles surrounding a grid cell, we start by selecting a particle spacing Δx less than the cell width h, with $b = \Delta x / h$. Here, b is a fractional measure of the inverse of the number of PPC. Next, place a particle at a location of $\alpha \in [0, \Delta x]$. Define $a = \alpha / \Delta x$ (a percentage shift). Next, fill the region $[-2h, 2h]$ with particles, maintaining the particle spacing Δx. Let the grid span the region $[-2h, 2h]$ with five grid nodes; thus, the grid locations will be $x_i = -2h, -h, 0, h, 2h$. Now, give all the particles' constant stress, $\sigma_p = \sigma$, and volume $V_p = \Delta x$. We will consider the calculation of f_i^{int} on grid node $i = 2$ (the center node). Since stress and volume are constant, we would expect f_2^{int} to be equal to 0.
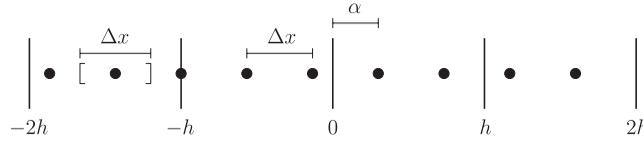
Figure 8. Example grid and particle arrangement used for the uniformly stressed body test.

The internal force in MPM is calculated as follows:

$$f_i^{\text{int}} = \sum_p \sigma_p \cdot \nabla \phi_i(x_p) V_p \tag{33}$$

For piecewise-linear basis functions, this becomes

$$f_2^{\text{int}} = \sum_{p \in [-h,0]} \sigma \frac{-1}{h} \Delta x + \sum_{p \in [0,h]} \sigma \frac{1}{h} \Delta x \tag{34}$$

$$= N_2 \sigma \frac{1}{h} \Delta x - N_1 \sigma \frac{1}{h} \Delta x \tag{35}$$

where N_1 is the number of particles in the region $[-h, 0]$, and N_2 is the number of particles in the region $[0, h]$. With this problem setup, there will be either equal number of particles on both sides of the grid nodes or $N_2 - N_1 = \pm 1$. Thus, f_2^{int} takes one of the three values: $\sigma(1/h)\Delta x$, 0 or $-\sigma(1/h)\Delta x$, depending on the particle arrangement. This can be seen in the contour plot shown in Figure 9 (top). Note that the error of $\sigma(1/h)\Delta x$ is half of the maximum analytical error shown in (24), which considered only the magnitudes of errors when particles overlapped discontinuities in $\nabla \phi_i$. If signs of errors were taken into account, the maximum error would be $\sigma(1/h)\Delta x$.

Several things can be observed in Figure 9. First, note that there are combinations of fractional offset a and inverse PPC b, which yield zero error. These combinations consist of two things: (1) when voxel boundaries line up with element boundaries such that there is no jump error term (like in the case of fractional offset $a = 0.5$ where there is a line of zero error for all choices of inverse PPC b) and (2) when symmetries in the particle positions cause cancellations in the error due to signs of the jumps. The second observation that can be seen in Figure 9 is that when quadratic B-spline basis functions are used, the magnitude of the maximum error (on the order of 0.25) is much less than with piecewise-constant basis functions (on the order of 1.0), and the error approaches zero much faster as the measure of particle spacing b decreases and the number of PPC increases. When cubic B-spline basis functions are used, the magnitude of the maximum error is again much lower (on the order of 0.04) and the error approaches zero faster than with quadratic B-splines or piecewise-linear basis functions. This suggests that the maximum error decreases and the convergence rate of the error improves as the continuity of the basis functions is increased.

To better understand the convergence of f_2^{int}, for each b value in Figure 9 the maximum error over the fractional offset a was tabulated and plotted in Figure 10 on log–log graphs. The left column shows the error for evenly spaced particles, whereas the right column shows particles in the same configuration, but randomly perturbed up to 40% of the measure of particle spacing b to the left or right from their nominal positions. Errors that are $\mathcal{O}(\Delta x)$ for piecewise-linear and $\mathcal{O}(\Delta x^2)$ for quadratic B-splines agree with the analysis from the previous section. The cubic B-splines demonstrate errors of $\mathcal{O}(\Delta x^3)$.
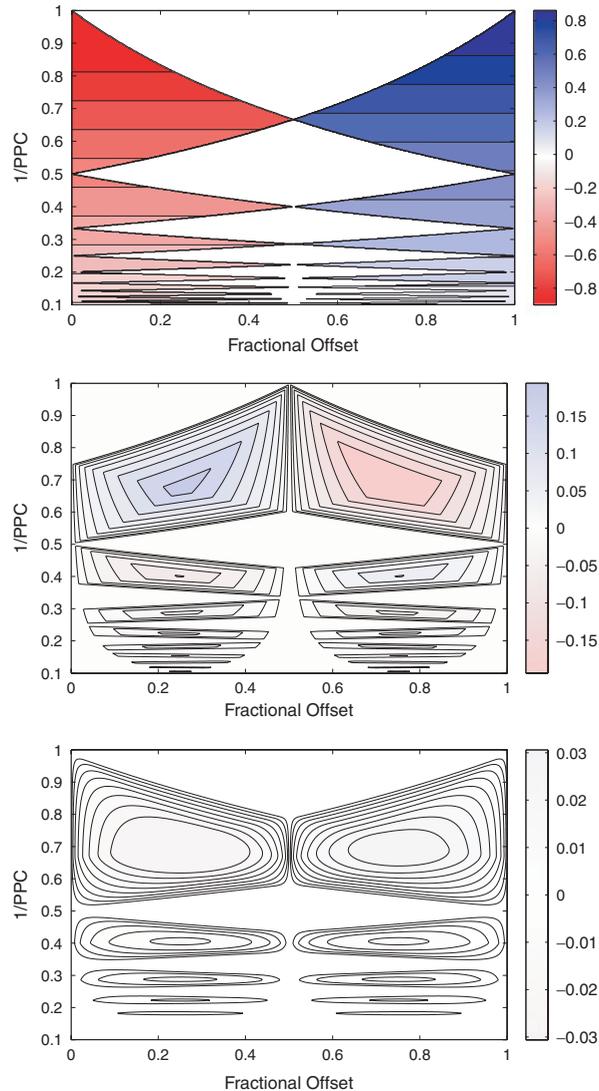
Figure 9. Plots of internal force error $|f_2^{\text{int}}|$ for a uniformly stressed body discretized with evenly spaced particles. Various particle spacings, b, and offsets, a, are shown using standard piecewise-linear (top), quadratic B-spline (middle) and cubic B-spline (bottom) basis functions. The figures show that the maximum error decreases and the convergence rate of the error improves as the continuity of the basis functions is increased. All plots use a consistent color scale.

The $\mathcal{O}(\Delta x^3)$ behavior of the cubic B-splines is due to the globally uniform spacing of particles in the test, even when the particles are perturbed. This uniform spacing of particles exploits the symmetry in ϕ and the $\mathcal{O}(\Delta x^2)$ error from Equation (30) cancels out. Another random test was run where particles were randomized in a global sense (the left half of the domain might have more particles than the right half of the domain) and the results are shown in Figure 11. Here, for larger
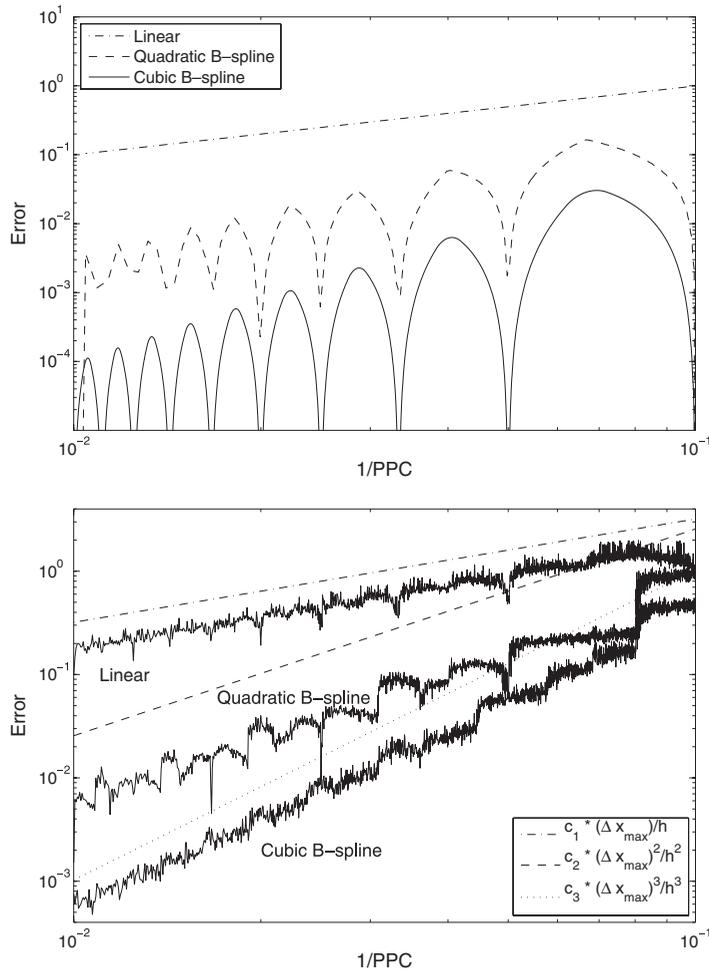
Figure 10. Errors in internal force *versus* particle spacing b (or the inverse of the number of PPC) for constant $\sigma = 1$ and grid spacing $h = 0.1$. The top plot uses evenly spaced particles where each sample point is the maximum error over various offsets a. The bottom plot uses randomly spaced particles. The constants on the error bounds are $c_1 = 2$, $c_2 = 1$ and $c_3 = \frac{1}{4}$. A tighter error bound may be possible for the evenly spaced particles due to symmetries. The $\Delta x_{\max}$ in the bottom plot is 1.6 times the Δx in the top plot due to the random spacing of the particles. This leads to higher error bounds for the randomly spaced particles than the evenly spaced particles.

values of b (that is, fewer PPC), the internal force has $\mathcal{O}(x^2)$ behavior. As b decreases, however, there exist more particles contributing to the integration and the $\mathcal{O}(x^3)$ behavior returns.

6.2. Test problem with dynamic traction boundary conditions

Previous analysis of the spatial convergence properties of MPM has been performed using quasi-static computations and comparisons with analytical solutions [6]. Since MPM is often used for dynamics problems, a 1D test case with an analytic transient solution was developed.
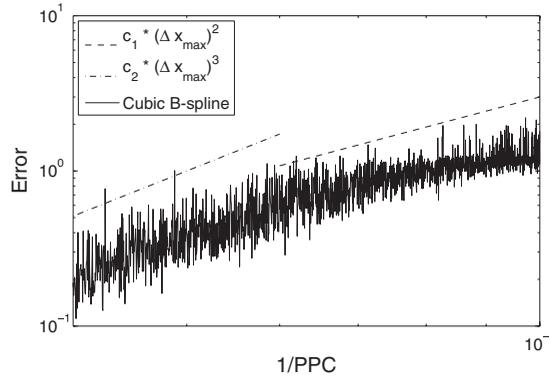
Figure 11. Errors in internal force *versus* particle spacing for constant $\sigma = 1$ and grid spacing $h = 0.1$. Here particles were globally randomly spaced (the left half of the domain can have more particles than the right half of the domain). This is opposed to the previous figure where particles were only locally randomly spaced and overall particle density remained constant throughout the domain. Instead of perturbing particles from a nominal even spacing, the globally random spacing is accomplished by filling the domain from one side to the other with randomly sized particles, while still using the same overall number of particles as the locally random case.

Given a bar of length l, fixed at $x = 0$, free at $x = l$, with Young's modulus of E, the density of $\rho = E$ (wave speed of 1.0), driven by a forcing function of

$$q(x,t) = \delta(x-l)H(t)\tau\sin(xt/l) \tag{36}$$

where $H(t)$ is the Heaviside step function, and initial conditions of $u(x,0) = 0$, $v(x,0) = 0$, has an analytical displacement function derived by wave propagations of the form:

$$u(x,t) = \begin{cases} 0, & t \in [0, l-x) \\ \alpha[1+\cos(\omega(t+x))], & t \in [l-x, l+x) \\ \alpha[\cos(\omega(t+x)) - \cos(\omega(t-x))], & t \in [l+x, 3l-x) \\ \alpha[-1-\cos(\omega(t-x))], & t \in [3l-x, 3l+x) \\ 0, & t \in [3l+x, 4l] \end{cases} \tag{37}$$

on $x \in [0, l]$ and $t \geqslant 0$, where $\alpha = l\tau/(\rho\pi)$ and $\omega = \pi/l$. The stress is given by

$$\sigma(x,t) = \begin{cases} 0, & t \in [0, l-x) \\ \tau\sin(\omega(t+x)), & t \in [l-x, l+x) \\ \tau[\sin(\omega(t+x)) + \sin(\omega(t-x))], & t \in [l+x, 3l-x) \\ \tau\sin(\omega(t-x)), & t \in [3l-x, 3l+x) \\ 0, & t \in [3l+x, 4l] \end{cases} \tag{38}$$

As was stated before, the traction for this problem occurs on the free end of the bar; however, the bar end position is time-dependent. The analytic end-bar position at any time t can be found
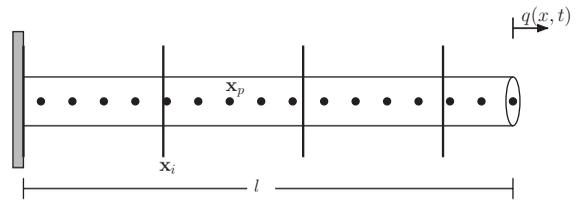
Figure 12. 1D bar with traction and corresponding MPM discretization.

by calculating $u(l, t)$ from (37). Interpolating the traction force, the external grid forces can then be calculated as

$$f_i^{\text{ext}} = \phi_i(l + u(l, t))q(l, t) \tag{39}$$

Since there is only one traction force, a maximum of two grid nodes will have non-zero external forces.

The parameters used were $\rho = E = 100$, $\tau = 1$ and $l = 1$. The uniform MPM grid spanned the region $[0, 1.15]$. The number of grid cells was varied to understand the spatial convergence of the methods. The bar was discretized using $n_p = 3n_g$ number of particles, where n_g is the number of MPM grid nodes. This ends up being slightly more than three PPC since the bar is only of length 1. The problem setup is illustrated in Figure 12.

The maximum extension of the bar occurs first at time $T = 1$, which results in an end-bar displacement of $u(l, 1) = \pi/50 \approx 0.06283$. The simulations were thus run to a time of 1 and the RMS errors in displacement were calculated as

$$e^{\text{RMS}} = \sqrt{\frac{1}{n_p} \sum_p (u(x_p, 1) - u_p)^2} \tag{40}$$

with $u_p = x_p - x_p^0$, the difference between the current and original position of particle p.

A number of important questions can be asked regarding the convergence properties of MPM. Recent studies by Wallstedt and Guilkey [24] looked at convergence with respect to the number of PPC. Kim *et al.* [25] provided an analysis framework for a modified version of MPM used for gas dynamics and showed first-order convergence of the method. Bardenhagen and Kober [6] performed grid resolution studies with MPM and GIMP in the context of a quasi-static compression problem. For the classic MPM, these tests showed convergence for a few data points corresponding to very low grid resolutions (between roughly 5 and 20 grid cells). However, as resolutions increased, the errors started to increase, showing a lack of convergence. In our studies, we fix the number of PPC at approximately 3.5 and focus our attention on the convergence properties with respect to grid resolution for a full dynamic test of the expansion of our fixed-free elastic bar. The results for the simulation with various basis functions are shown in Figure 13.

Similar to the Bardenhagen tests [6], standard MPM using piecewise-linear basis functions shows a lack of convergence for 20 grid cells and higher. Simply substituting smoother basis functions for the standard piecewise-linear basis functions traditionally used in MPM drastically improves spatial convergence, with convergence rates nearing 2 for both quadratic B-splines and cubic B-splines. Significant integration errors will always exist when using nodal integration such as in MPM, especially when particles are free to move through the domain. Error plateaus can
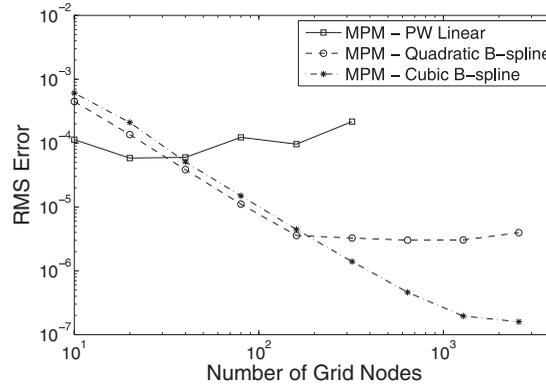
Figure 13. Convergence test for a 1D fixed free elastic bar with sinusoidal traction. The RMS errors in displacements are calculated at time $T = 1$, when the bar is at maximum extension.

be seen in the simulation when B-spline basis functions are used; however, positive convergence results can be obtained to many thousands of grid cells.

The level of improvement due to increasing the basis function regularity is greater than that which might be anticipated based solely on the quadrature results seen in the previous section. The benefit of increased regularity goes beyond just improving quadrature as seen in Figure 10. In future work, we will investigate the other parts of the MPM algorithm that might benefit from the increased level of smoothness of the grid basis functions.

7. SUMMARY AND CONCLUSIONS

In this paper we have considered the impact of the MPM's choice of maintaining only particle position and volume information when approximating integrals of particle voxels. The nodal integration of equations within MPM was shown to be similar to using midpoint approximations to the integrals. However, the underlying equations being integrated have discontinuities and the partitioning of the domain specified by the particle configuration does not respect these discontinuities, leading to quadrature errors containing information about the jumps in the integrand. Errors in using the midpoint rule across discontinuities were analyzed and applied to the MPM force calculation, the results showing that simply using smoother basis functions such as quadratic and cubic B-splines drastically reduce integration errors.

Grid resolution tests with quadratic B-spline basis functions showed positive spatial convergence up to about 120 grid cells, allowing much lower errors than MPM with piecewise-linear basis functions while only increasing the basis function span by one extra grid cell width. After 120 grid cells, the error plateaus and remains effectively the same up 2560 grid cells. The same test was run with cubic B-splines where positive spatial convergence was demonstrated out to 2560 grid cells, although the convergence rate starts to drop dramatically after 1280 grid cells. These results are in stark contrast to those obtained using standard (piecewise-linear) MPM in which a lack of convergence is observed.

The analysis and corresponding convergence studies suggest that basis functions smoother than piecewise-linear should be used for moderate grid resolutions. B-spline basis functions are simple

to construct and are easily extendible to multiple dimensions. We have implemented these multi-dimensional B-spline basis functions in Uintah, a massively parallel problem-solving environment from the University of Utah [26], which provides a framework for large-scale MPM simulations. For future work, we plan to accomplish comparisons in 2D and 3D where other subtitles not exhibited in 1D arise, namely, additional complexity of handling boundary conditions and material contact conditions.

APPENDIX A

A.1. Integrating piecewise-constant functions with the midpoint rule

If $y(x)$ is piecewise constant with a discontinuity at $x = 0$, as shown in Figure 5, then y can be thought of as

$$y(x) = \begin{cases} y_1(x) = a_1, & x \leqslant 0 \\ y_2(x) = a_2, & x > 0 \end{cases} \tag{A1}$$

Assuming $0 \in [\xi - \Delta x/2, \xi + \Delta x/2]$, the exact integral of $y(x)$ over the region needs to be evaluated in two parts,

$$\int_{\xi - \Delta x/2}^{0} y_1 \, \mathrm{d}x = a_1 x \big|_{\xi - \Delta x/2}^{0} = -a_1 \left(\xi - \frac{\Delta x}{2} \right) \tag{A2}$$

and

$$\int_{0}^{\xi + \Delta x/2} y_2 \, \mathrm{d}x = a_2 \left(\xi + \frac{\Delta x}{2} \right) \tag{A3}$$

giving the total integral over the region

$$\int_{\xi - \Delta x/2}^{\xi + \Delta x/2} y(x) \, \mathrm{d}x = (a_2 - a_1)\xi + \frac{1}{2}(a_2 + a_1)\Delta x \tag{A4}$$

Using the midpoint rule over the same region gives $\mathrm{MP}(\xi) = y(\xi)\Delta x$. When $\xi \leqslant 0$, $\mathrm{MP}(\xi) = a_1 \Delta x$. The integration or the quadrature error, in this case is

$$E(\xi) = \int_{\xi - \Delta x/2}^{\xi + \Delta x/2} y(x) \, \mathrm{d}x - \mathrm{MP}(\xi) = (a_2 - a_1) \left[\xi + \frac{1}{2} \Delta x \right] \tag{A5}$$

The only value for ξ which gives zero error in (A5) is $\xi = -\Delta x/2$, which corresponds to a zero error when integrating the region $[-\Delta x, 0]$. Since $y(x) = y_1$, with no jumps on the open interval $(-\Delta x, 0)$, a zero error when $\xi = -\Delta x/2$ makes sense because the midpoint rule can integrate a constant function exactly. When $\xi > 0$, a similar analysis shows that $E(\Delta x/2) = 0$. The maximum error magnitude occurs when $\xi = 0$, giving

$$E_{\max} = E(0) = \tfrac{1}{2}|a_2 - a_1|\Delta x \tag{A6}$$

A.2. Integrating piecewise-linear functions with the midpoint rule

If $y(x)$ is piecewise linear, as shown in Figure 6, composed of $y_1(x)$ for $x \leqslant 0$ and $y_2(x)$ for $x > 0$, with $y_1(0) = y_2(0) = 0$ and $y_1'(0) \neq y_2'(0)$, $y(x)$ can be expressed as

$$y(x) = \begin{cases} y_1(x) = a_1 x, & x \leqslant 0 \\ y_2(x) = a_2 x, & x > 0 \end{cases} \tag{A7}$$

The exact integral of $y(x)$ over the region $[\xi - \Delta x/2, \xi + \Delta x/2]$ needs to be evaluated in two parts,

$$\int_{\xi - \Delta x/2}^{0} y_1 \, dx = a_1 x^2 |_{\xi - \Delta x/2}^{0} = -\frac{1}{2} a_1 \left(\xi^2 - \xi \Delta x + \frac{1}{4} \Delta x^2 \right) \tag{A8}$$

and

$$\int_{0}^{\xi + \Delta x/2} y_2 \, dx = a_2 x^2 |_{0}^{\xi + \Delta x/2} = \frac{1}{2} a_2 \left(\xi^2 + \xi \Delta x + \frac{1}{4} \Delta x^2 \right) \tag{A9}$$

giving the total integral over the region

$$\int_{\xi - \Delta x/2}^{\xi + \Delta x/2} y(x) \, dx = \frac{1}{2}(a_2 - a_1)\xi^2 + \frac{1}{2}(a_2 + a_1)\xi \Delta x + \frac{1}{8}(a_2 - a_1)\Delta x^2 \tag{A10}$$

Using the midpoint rule over the same region gives $\mathrm{MP}(\xi) = y(\xi)\Delta x$. When $\xi \leqslant 0$, $\mathrm{MP}(\xi) = a_1 \xi \Delta x$. The integration, or the quadrature error, in this case is

$$E_1(\xi) = \int_{\xi - \Delta x/2}^{\xi + \Delta x/2} y(x) \, dx - \mathrm{MP}(\xi) = \frac{1}{2}(a_2 - a_1)\xi^2 + \frac{1}{2}(a_2 - a_1)\xi \Delta x + \frac{1}{8}(a_2 - a_1)\Delta x^2 \tag{A11}$$

To find values of ξ corresponding to the maximum or minimum error, we solve $d/d\xi E_1 = 0$ for ξ:

$$\frac{d}{d\xi} E_1 = (a_2 - a_1)\xi + \frac{1}{2}(a_2 - a_1)\Delta x \tag{A12}$$

$$= (a_2 - a_1)\left(\xi + \frac{1}{2}\Delta x \right) \tag{A13}$$

Since $a_2 \neq a_1$, the above equation is zero only when $\xi = -\Delta x/2$. A similar analysis for $\xi > 0$ shows that $d/d\xi E_2 = 0$ only when $\xi = \Delta x/2$. To find the maximum magnitude, we first note that $E(\pm \Delta x/2) = 0$, which also implies that the midpoint rule integrates a linear function exactly. Finally, since $E_1(0) = E_2(0)$, the maximum magnitude error must be when $\xi = 0$, giving

$$E_{\max} = E(0) = \frac{1}{8}|a_2 - a_1|\Delta x^2 \tag{A14}$$

A.3. Integrating piecewise-quadratic functions with the midpoint rule

If $y(x)$ is piecewise-quadratic, as shown in Figure 7, composed of $y_1(x)$ for $x \leqslant 0$ and $y_2(x)$ for $x > 0$, with $y_1(0) = y_2(0) = 0$ and $y_1'(0) = y_2'(0)$, and $y_1''(0) \neq y_2''(0)$, $y(x)$ can be expressed as

$$y(x) = \begin{cases} y_1 = a_1 x^2 + bx, & x \leqslant 0 \\ y_2 = a_2 x^2 + bx, & x > 0 \end{cases} \tag{A15}$$

The exact integral of $y(x)$ over the region $[\xi - \Delta x/2, \xi + \Delta x/2]$ needs to be evaluated in two parts,

$$
\begin{aligned}
\int_{\xi - \Delta x/2}^{0} y_1 &= \left[\frac{1}{3} a_1 x^3 + \frac{1}{2} b x^2\right]_{\xi - \Delta x/2}^{0} \\
&= -\left[\frac{1}{3} a_1 \left(\xi - \frac{\Delta x}{2}\right)^3 + \frac{1}{2} b \left(\xi - \frac{\Delta x}{2}\right)^2\right] \\
&= -\frac{1}{3} a_1 \left(\xi^3 - \frac{3}{2} \xi^2 \Delta x + \frac{3}{4} \xi \Delta x^2 - \frac{1}{8} \Delta x^3\right) - \frac{1}{2} b \left(\xi^2 - \xi \Delta x + \frac{1}{4} \Delta x^2\right)
\end{aligned}
\tag{A16}
$$

and

$$
\begin{aligned}
\int_{0}^{\xi + \Delta x/2} y_2 &= \left[\frac{1}{3} a_2 x^3 + \frac{1}{2} b x^2\right]_{0}^{\xi + \Delta x/2} \\
&= \frac{1}{3} a_2 \left(\xi^3 + \frac{3}{2} \xi^2 \Delta x + \frac{3}{4} \xi \Delta x^2 + \frac{1}{8} \Delta x^3\right) + \frac{1}{2} b \left(\xi^2 + \xi \Delta x + \frac{1}{4} \Delta x^2\right)
\end{aligned}
\tag{A17}
$$

giving the total integral over the region

$$
\begin{aligned}
\int_{\xi - \Delta x/2}^{\xi + \Delta x/2} y &= \int_{\xi - \Delta x/2}^{0} y_1 + \int_{0}^{\xi + \Delta x/2} y_2 \\
&= \frac{1}{3} (a_2 - a_1) \xi^3 + \frac{1}{2} (a_2 + a_1) \Delta x \xi^2 \\
&\quad + \frac{1}{4} (a_2 - a_1) \Delta x^2 \xi + \frac{1}{24} (a_2 + a_1) \Delta x^3 + \xi b \Delta x
\end{aligned}
\tag{A18}
$$

Using the midpoint rule over the same region gives $\mathrm{MP}(\xi) = y(\xi) \Delta x$. When $\xi \leqslant 0$, the midpoint rule gives

$$
\mathrm{MP}(\xi) = (a_1 \xi^2 + b \xi) \Delta x
\tag{A19}
$$

The integration error across the discontinuity is then

$$
\begin{aligned}
E_1(\xi) &= \int_{\xi - \Delta x/2}^{\xi + \Delta x/2} y - \mathrm{MP}(\xi) \\
&= \frac{1}{3} (a_2 - a_1) \xi^3 + \frac{1}{2} (a_2 - a_1) \Delta x \xi^2 + \frac{1}{4} (a_2 - a_1) \Delta x^2 \xi + \frac{1}{24} (a_2 + a_1) \Delta x^3
\end{aligned}
\tag{A20}
$$

To find ξ corresponding to the maximum error, we solve for $\mathrm{d}/\mathrm{d}\xi E_1 = 0$.

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\xi} E_1 &= (a_2 - a_1) \xi^2 + (a_2 - a_1) \Delta x \xi + \frac{1}{4} (a_2 - a_1) \Delta x^2 \\
&= (a_2 - a_1) \left(\xi + \frac{1}{2} \Delta x\right)^2
\end{aligned}
\tag{A21}
$$

Since $a_2 \neq a_1$, this is zero only when $\xi = -\frac{1}{2}\Delta x$. A similar analysis for $\xi > 0$ shows that $d/d\xi E_2 = 0$ only when $\xi = \Delta x/2$. Since $E(\pm\frac{1}{2}\Delta x) = 0$, $|E|$ must be a maximum at $\xi = 0$. Therefore, the maximum error is given by

$$E_{\max} = E(0) = \frac{1}{24}|a_1 + a_2|\Delta x^3$$

ACKNOWLEDGEMENTS

REFERENCES

1. Sulsky D, Chen Z, Schreyer HL. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering* 1994; **118**:179–196.
2. Sulsky D, Zhou S, Schreyer HL. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* 1995; **87**:236–252.
3. Bardenhagen SG, Brydon AD, Guilkey JE. Insight into the physics of foam densification via numerical simulation. *Journal of the Mechanics and Physics of Solids* 2005; **53**(3):597–617.
4. Nairn JA. Numerical simulations of transverse compression and densification in wood. *Wood and Fiber Science* 2006; **38**(4):576–591.
5. Sulsky D, Schreyer H, Peterson K, Kwok R, Coon M. Using the material-point method to model sea ice dynamics. *Journal of Geophysical Research* (*Oceans*) 2007; **112**:C02S90. Available at: http://www.agu.org/pubs/crossref/2007/2005JC003329.shtml.
6. Bardenhagen SG, Kober EM. The generalized interpolation material point method. *Computer Modeling in Engineering and Science* 2004; **5**(6):477–495.
7. Brackbill JU. Particle methods. *International Journal for Numerical Methods in Fluids* 2005; **47**:693–705.
8. Brackbill JU, Ruppel HM. FLIP: a method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics* 1986; **65**:314–343.
9. Brackbill JU, Kothe DB, Ruppel HM. FLIP: a low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications* 1988; **48**:25–38.
10. Li S, Liu WK. *Meshfree Particle Methods*. Springer: Berlin, 2004.
11. Belytschko T, Lu YY, Gu L. Element free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**(2):229–256.
12. Grigoryev YN, Vshivkov VA, Fedoruk MP. *Numerical 'Particle-in-Cell' Methods*. VSP: Utrech, Boston, 2002.
13. Dolbow J, Belytschko T. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics* 1999; **23**(3):219–230.
14. Beissel S, Belytschko T. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**(1):49–74.
15. Chen JS, Yoon S, Wu CT. Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering* 2002; **53**(12):2587–2615.
16. Vshivkov VA. The approximation properties of the particles-in-cells method. *Computational Mathematics and Mathematical Physics* 1996; **36**(4):509–515.
17. Quinlan NJ, Basa M, Lastiwka M. Truncation error in mesh-free particle methods. *International Journal for Numerical Methods in Engineering* 2006; **66**(13):2064–2085.
18. Sulsky D, Kaul A. Implicit dynamics in the material-point method. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**(12–14):1137–1170.
19. Guilkey JE, Weiss JA. Implicit time integration for the material point method: quantitative and algorithmic comparisons with the finite element method. *International Journal for Numerical Methods in Engineering* 2003; **57**(9):1323–1338.

20. Love E, Sulsky DL. An unconditionally stable, energy–momentum consistent implementation of the material-point method. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**(33–36):3903–3925.
21. Simo JC, Hughes TJR. *Computational Inelasticity*. Springer: Berlin, 1998.
22. Ma J, Lu H, Komanduri R. Structured mesh refinement in generalized interpolation material point (GIMP) method for simulation of dynamic problems. *Computer Methods in Applied Mechanics and Engineering* 2006; **12**:213–227.
23. Ralston A, Rabinowitz P. *A First Course in Numerical Analysis* (2nd edn). Dover: New York, 2001.
24. Wallstedt P, Guilkey J. Improved velocity projection for the material point method. *Computer Modeling in Engineering and Sciences* 2007; **19**(3):223–232.
25. Kim J, Berzins M, Tran LT. Solving time-dependent PDEs using the material point method, a case study from gas dynamics. *Technical Report UUSCI-2007-010*, SCI Institute, University of Utah, 2007.
26. Davison de St Germain J, McCorquodale J, Parker SG, Johnson CR. Uintah: a massively parallel problem solving environment. *Ninth IEEE International Symposium on High Performance Distributed Computing (HPDC-9'00)*, 2000; 33. Available at: http://doi.ieeecomputersociety.org/10.1109/HPDC.2000.868632.