Noname manuscript No.
(will be inserted by the editor)

IMPROVING ACCURACY IN THE MPM METHOD BY USING A NULL SPACE FILTER

Chris Gritton[1] · **Martin Berzins**[2]

Received: date / Accepted: date

Abstract The Material Point Method (MPM) has been very successful in providing solutions to many challenging problems involving large deformations. Nevertheless there are some important issues that remain to be resolved with regard to its analysis. One key challenge applies to both MPM and Particle in Cell (PIC) methods and arises from the difference between the number of particles and the number of the nodal grid points to which the particles are mapped. This difference between the number of particles and the number of grid points gives rise to a non-trivial null space of the linear operator that maps particles values onto nodal grid-point values. In other words, there are non-zero particle values that when mapped to the grid point nodes result in a zero value there. Moreover when the nodal values at the grid points are mapped back to particles part of those particle values may be in that same null space. Given positive mapping weights from particles to nodes such null space values are oscillatory in nature. While this problem has been observed almost since the begining of PIC methods there are still elements of it that are problematical today as well as methods that transcend it. The null space may be viewed as being connected to the ringing instability identified by Brackbill for PIC methods. It will be shown that it is possible to remove these null space values from the solution by using a null space filter. This filter improves the accuracy of the MPM methods by using an approach that is based upon a local Singlular Value Decomposition (SVD) calculation. This local SVD approach is compared against the global SVD approach previously considered by the authors and to a recent MPM method by Zhang and colleagues.

Keywords MPM · Particles · Null Space · Instability

C. Gritton
SCI Institute, University of Utah, Salt Lake City, USA
E-mail: cgritton@sci.utah.edu

M. Berzins
SCI Institute, University of Utah, Salt Lake City, USA

1 Introduction

Particle-in-cell, (PIC), methods and the Material Point Method (MPM) that has been developed from it are based on moving particles whose dynamics are defined by calculating quantities such as acceleration on a fixed background grid, have been in use and development for over sixty years. The original PIC method was developed by F.H. Harlow as a hydrodynamics code [16] to handle fluid dynamics problems that involved large slips and distortions. MPM [34] may be viewed as being a solid mechanics method that is derived from the fluid implicit particle, FLIP, [9] method which is itself a modification to the original PIC method that is applied to fluid dynamics problems. The main idea in such methods is to make use of basis functions at particles (originally delta functions) and basis functions at nodal grid points (originally linear basis "hat" functions). The MPM method has been much studied since it was introduced and there are many examples of papers that improve the performance of the method. Examples of such papers are the improved basis functions for MPM derived by [3,30,37] and the higher order basis functions derived by [32,33]. Many of these papers produce improved results by developing methods that help reduce the grid crossing error that occurs when particles cross a grid cell boundary. In general The combination of moving Lagrangian particles and a fixed Eulerian grid used by PIC and MPM is successful on many challenging problems, however many theoretical issues to do with such methods remain at least partially unresolved in the areas of stability, accuracy and convergence.

One such issue is that the PIC method is well-known, e.g. [21,8], to have an aliasing error due to the difference between the degrees of freedom at the grid points of the spatial mesh cell compared to the degrees of freedom at the particles. This error may result in oscillatory solution values. For example Brackbill [8] states that, *Because the number of particles is finite, the number of Fourier modes is also finite. Thus, when there are n particles in each cell, there are n times as many Fourier modes as there are grid points.* When values are mapped from nodes to particles the lack of resolution at the nodes compared to resolution at the particles can cause an aliasing error. Again to quote Brackbill, *Aliases occur because all Fourier modes with wavelengths shorter than the grid spacing are indistinguishable at the grid points.* [8]. This aliasing is exactly the null space error addressed in this work. Brackbill showed that the user of better interpolants associated with the nodes to particles mapping helped reduce these errors and conducted a Fourier analysis of the problem. Early attempts to address this instability started from the PIC jiggling work of Langdon [21] and Chen, Langdon and Birdsall [11] to Brackbill and Lapenta [7]. The jiggling approach artificially moves the points and results in reduced oscillations.

An alternative approach is to use the conservation of energy to improve the performance of PIC and MPM. While conservation of energy does not necessarily itself imply stability [28], it is a desirable property for a numerical method. Examples of work to improve the accuracy and stability of PIC methods by introducing energy conservation are Lapenta et al. [20] and Brackbill [10] address energy and momentum conservation properties of PIC applied to plasma calculations. At the same time it is not entirely clear how the approach used with PIC methods applied to plasma calculations may be used with MPM. Energy conservation in MPM is addressed

by Bardenhagen [2] and Love and Sulsky provide energy consistent and conserving approaches, [24,25]. Finally the work of Mast et al. [26] addresses the locking phenomenon observed originally in Finite Element Methods in the context of MPM. Locking is a phenomemon that is different to the grid crossing error mentioned above and is addressed by multi-dimensional filtering of stresses and strains. However while these papers greatly improve the accuracy and conservation properties of PIC and MPM and indirectly result in more stable methods they do not directly address the null space issue in MPM. Part of the challenge with undertaking analysis of MPM as Wallstedt and Guilkey point out is its nonlinearity [36] and the fact that moving particles may dynamically change the effective computational stencils frequently.

Similar issues of stability due to moving particles arise in Smooth Particle Hydrodynamics (SPH) and in the many Finite Element formulations of particle methods. For examples of the related work on SPH [18,29,27] and finite element methods, see [4,5,6]. Belytschko and Xaio [6] address two sources of instability of their particle methods. The first one is a rank defficiency of the discrete equations that is similar to the ringing instability or the null space problem considered here and the second is a distortion of the material instability. The work on these methods deals with continually moving points without having a background grid as is used in MPM. For this reason it is not at all clear how the body of work introduced above immediately relates to MPM with its mixture of Lagrangian particles and an Eulerian background grid.The general approaches adopted clearly have some potential application to MPM however.

In previous work we began to address the general area of spurious oscillations in PIC [15] based on a matrix approach that directly reflects the fact that there are non-zero values at the particles that cannot be "seen" at the nodes and the space of such values is oscillatory. The positive results that were obtained have provided encouragement to expand these ideas to MPM. As the intention here is to shed some light upon this issue for MPM, attention is focused on one dimensional problems to add clarity. Section 2 and 3 describes the MPM method. Section 4 addresses the null space that results from the difference between the number of particles and the nodal grid points points that they are mapped to. While a global approach based on a singular value decomposition for addressing this was given in [15], Section 5 defines a method based upon a local calculation using values in the elemnents on either side of a grid node. Finally Section 6 describes a numerical experiment that illustrates the benefits of this approach and compares it against the method of [37] as well as some of the challenges that remain.

2 Material Point Method

The model problem used here is a pair of equations connecting velocity v, displacement u and density ρ and is typical of straightforward MPM applications:

$$\frac{Du}{Dt} = v, \tag{1}$$

$$\rho \frac{Dv}{Dt} = \frac{\partial \sigma}{\partial x} + b, \tag{2}$$

with a linear stress model $\sigma = E\frac{\partial u}{\partial x}$ for which Young's modulus, E, is constant, a body force b and with appropriate boundary and initial conditions. In describing MPM the starting point is to assume that there is a mesh of $N+1$ fixed nodes X_i on some fixed interval $[a,b]$such that

$$a = X_0 < X_1 < ... < X_N = b. \tag{3}$$

The mesh has N elements with the interval I_i defined by

$$I_i = [X_i, X_{i+1}], \tag{4}$$

and the width of each interval is denoted by

$$h_i = X_i - X_{i-1}. \tag{5}$$

A fixed evenly spaced mesh is used here with $h = h_i \forall i$.

It is assumed that there are m particles between each pair of nodes, situated at x_p^n points where at each time step, $t^n = \delta t * n$, where n is the nth time step, and the computed solution at the pth particles will be written as $u_p^n = u(x_p^n, t^n)$. The superscript n and the dependence on t will often be dropped unless they are necessary to clarify timestep issues. The restriction of attention to equal numbers of particles in each interval is used solely to simplify the algebra in the analysis that follows and imposes no constraint on the approaches discussed. In this case the particles in interval i lie between X_i and X_{i+1} and have positions $x_{im+j}, j = 1,..,m$.

2.1 Mapping Matrix For Solution Values

In both the MPM and PIC grid-based particle methods it is necessary to map values from the particles to the nodes and from the nodes to the particles [34,3]. For example, in mapping particles to the node X_i and back to particles, the linear basis functions centered at that point and given by $\phi_i(x)$,

$$\phi_i(x) = \frac{x - X_i}{X_i - X_{i-1}}, X_{i-1} \leq x \leq X_i, \tag{6}$$

$$\phi_i(x) = \frac{X_i - x}{X_{i+1} - X_i}, X_{i+1} \geq x \geq X_i. \tag{7}$$

$$\phi_i(x) = 0, x \notin [X_{i-1}, X_{i+1}] \tag{8}$$

are often used. There are of course many alternative choices [3,30,37,32,33].

In the case of the basis functions associated with particles there are a number of possibilities ranging from delta functions to approaches such as [3]. In general given a function value $u_p(t)$ at a particle x_p, the representation used in MPM is given by $u_p(t)\chi_p(x,t)$ where the basis functions $\chi_p(x,t)$ in the case of delta functions $\delta(x-x_p)$ are given by:

$$\chi_p(x,t) = \delta(x - x_p)V_p \tag{9}$$

where V_p is the width in one dimension of the particle as defined below. In two dimensions V_p is the area of the particle and in three dimensions its volume. Bardenhagen points out [3] that this is not to be a partition of unity in that

$$\sum_p \chi_p(x,t) \neq 1. \tag{10}$$

As a result other representations such as piecewise constant functions

$$\chi_p(x,t) = 1, x \in V_p, \tag{11}$$
$$\chi_p(x,t) = 0, x \notin V_p, \tag{12}$$

or even more complex basis functions such as [30,32,33]are used. In general the volume of the pth particle V_p is defined as

$$V_p = \int_a^b \chi_p(x,t)dx. \tag{13}$$

The particle volumes are initially defined to span the interval $[a,b]$ and are then modified as the particles move, as is shown in the next section.

The mapping from particles to nodes in MPM takes into account all the particles in adjacent intervals to the node in question (X_i for example) and may be written, regardless of the choice of particle basis, as

$$U_i(t) = \sum_p S_{ip} u_p(t). \tag{14}$$

where $U_i(t)$ is the value at the node X_i. The argument (t) will often be dropped unless needed for clarity. From [3,31] the mapping constants are given by

$$S_{ip} = \frac{1}{\int_a^b \chi_p(x,t)dx} \int_a^b \chi_p(x,t)\phi_i(x)dx. \tag{15}$$

In the case of linear basis functions and particles represented by equation (9) it follows that

$$S_{ip} = \phi_i(x_p). \tag{16}$$

This mapping may be modified in two ways. The first modification is that a mass weighting is often used when velocity is mapped e.g. [36], see equation (40) below. The second modification is that the S_{ip} multipliers correspond to a partition of unity by dividing by the sum of all the components that contribute to node X_i using the modified mapping coefficients S^*_{ip} defined by

$$S^*_{ip} = \frac{\phi_i(x_p)}{W_i} \tag{17}$$

where $W_i = \sum_{p \in I_{i-1} \cup I_i} \phi_i(x_p)$ and so that

$$\sum_p S^*_{ip} = 1. \tag{18}$$

While this mapping from particles to a node can also be expressed in terms of a system-wide matrix and this is the approach adopted in [15], here a local version of this mapping from the set of particles in the two elements on either side of a grid point to that grid point will be used.

$$U_i = \mathbf{S}^*_{ip}\mathbf{u}_{ip}, \tag{19}$$

where U_i is the mapped value at the node X_i and $\mathbf{u}_{ip}$ is a vector of the values at the particles in the two adjacent elements to X_i. Thus as there are $2m$ particles in these two elements then the matrix $\mathbf{S}^*_{ip}$ has only one row and $2m$ columns with the value in the jth column being given by $S^*_{i((i-1)m+j)}$. As the entries of the row vector $\mathbf{S}^*_{ip}$ are positive, it is straightforward to construct vectors consisting of non-zero entries, say $\mathbf{v}_{ip}$ that give rise to zero values at the nodes. In other words that

$$\mathbf{S}^*_{ip}\mathbf{v}_{ip} = 0, \tag{20}$$

This is the null space problem in our previous work [15] and is also, when considered for gradient mappings, termed a high frequency instability that arises from the rank defficiency of the discrete divergence operator [4]. This topic will be examined in detail in below.

2.2 Internal Force Calculation

The calculation of the internal forces in MPM at the nodes requires the calculation of the volume integral of the divergence of the stress, [36]. The divergence of the stress, σ, at a node X_i, as denoted by $\frac{\partial \sigma_i}{\partial x}$ given the stress at particles σ_p. After integration by parts the force calculation is written as

$$f_i^{int} = -\sum_p DS_{ip}\sigma_p V_p \tag{21}$$

where in the case of linear functions at the nodes from [3] in the case when Delta functions are used to represent particles it follows that:

$$DS_{ip} = \frac{d\phi(x_p)}{dx}. \tag{22}$$

In the same way that the mapping coefficients S^*_{ip} are modified to be a partition of unity the coefficients DS_{ip} are amended to reproduce derivatives of constant and linear functions exactly. In order to reproduce linear functions exactly a derivative correction is needed [19,5]. Multiple methods have been proposed to correct for this deficiency [22,18,29,12]. In one space dimension the modified coefficients for the derivative of a constant to be zero are given by

$$DS^{**}_{ip} = \frac{1}{\sum_{pp}\phi_i(x_{pp})}\frac{d\phi_i(x_p)}{dx} - \frac{\phi_i(x_p)}{(\sum_{pp}\phi_i(x))^2}\sum_{pp}\frac{d\phi_i(x_{pp})}{dx} \tag{23}$$

From this it follows immediately that

$$\sum_p DS^{**}_{ip} = 0. \tag{24}$$

A constant C is then used to ensure that differentiating the function x at the the particles give the correct value of one at the node X_i;

$$C\sum_p DS^{**}_{ip} x_p = 1. \tag{25}$$

The correction thus is defined [6] by,

$$C = \frac{1}{\sum_p x_p DS^{**}_{ip}}. \tag{26}$$

So that if the mapping coefficients are defined by

$$DS^{*}_{ip} = \frac{1}{\sum_{pp} x_{pp} DS^{**}_{ipp}} \cdot \left[\frac{1}{\sum_{pp} \phi_i(x_{pp})} \frac{d\phi_i(x_p)}{dx} - \frac{\phi_i(x_p)}{(\sum_{pp} \phi_i(x))^2} \sum_{pp} \frac{d\phi_i(x_{pp})}{dx} \right]. \tag{27}$$

The Force calculation is given by

$$f^{int}_i = -\sum_p DS^{*}_{ip} \sigma_p V_p \tag{28}$$

It should also be noted that this is essentially a derivative approximation in that the derivative at a node $dy/dx(X_i)$ of a function whose values at particles are given by y_p is approximated by

$$\frac{dy}{dx}(X_i) = -\sum_p DS^{*}_{ip} y_p V_p \tag{29}$$

2.3 Interpolating Back to Particles

Once the nodal values of either the solution or the derivative are calculated then by using the piecewise linear basis functions, $\phi_i(x)$, associated with the mesh nodes, a linear approximation to the function $u(x)$ at a particle may be defined as,

$$u_p = \sum_i U_i \phi_{ip}. \tag{30}$$

where $\phi_{ip} = \phi_i(x_p)$. It is convenient to write this mapping as a matrix operation. In the case of linear basis functions the individual mappings have the form given by:

$$u_{m(i-1)+j} = U_{i-1}\alpha_{m(i-1)+j} + U_i(1-\alpha_{m(i-1)+j}), \tag{31}$$

$$u_{mi+j} = U_i\alpha_{mi+j} + U_{i+1}(1-\alpha_{mi+j}), j = 1,...,m. \tag{32}$$

where

$$\alpha_{m(i-1)+j} = \frac{X_i - x_{m(i-1)+j}}{h_i}, \tag{33}$$

$$\alpha_{mi+j} = \frac{X_{i+1} - x_{mi+j}}{h_{i+1}}, j = 1,...,m. \tag{34}$$

This may be written as a matrix equation to define the values at all the points in $[X_{i-1}, X_{i+1}]$.

$$\mathbf{u}_{ip} = \hat{\mathbf{M}} \begin{bmatrix} U_{i-1} \\ U_i \\ U_{i+1} \end{bmatrix}, \tag{35}$$

where the mapping matrix $\hat{\mathbf{M}}$ is defined by:

$$\hat{\mathbf{M}} = \begin{bmatrix} \alpha_{m(i-2)+1} & (1-\alpha_{m(i-2)+1}) & 0 \\ \vdots & \vdots & \vdots \\ \alpha_{m(i-2)+m} & (1-\alpha_{m(i-2)+m}) & 0 \\ 0 & \alpha_{m(i-1)+1} & (1-\alpha_{m(i-1)+1}) \\ \vdots & \vdots & \vdots \\ 0 & \alpha_{mi} & (1-\alpha_{mi}) \end{bmatrix}$$

and where the vector $\mathbf{u}_{ip}$ is defined as in equation (19). The mapping to construct derivative values at the particle points is similar and is defined by

$$\frac{\partial u_{m(i-1)+j}}{\partial x} = (U_i - U_{i-1})/(X_i - X_{i-1}), \tag{36}$$

$$\frac{\partial u_{mi+j}}{\partial x} = (U_{i+1} - U_i)/(X_{i+1} - X_i), j = 1, ..., m, \tag{37}$$

and, in the case of evenly-spaced nodes, gives rise to a similar form of the mapping matrix.

$$\hat{\mathbf{M}}_{\mathbf{D}} = \frac{1}{h} \begin{bmatrix} -1 & 1 & 0 \\ \vdots & \vdots & \vdots \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ \vdots & \vdots & \vdots \\ 0 & -1 & 1 \end{bmatrix}$$

and a similar equation to calculate the derivatives at the particles as in equation 35.

$$\frac{\partial}{\partial x}\mathbf{u}_{ip} = \hat{\mathbf{M}}_{\mathbf{D}} \begin{bmatrix} U_{i-1} \\ U_i \\ U_{i+1} \end{bmatrix}. \tag{38}$$

3 MPM Method

The description of the MPM method given here follows that of a number of authors such as [36]. The first two steps in MPM are to compute the mass and velocity values at the nodes. Masses at the particles m_p are mapped to the nodes as follows,

$$m_i = \sum_p S^*_{ip} m_p. \tag{39}$$

Velocity is similarly mapped to the nodes as follows,

$$v_i^n = \sum_p \hat{S}_{ip} v_p^n, \tag{40}$$

where the coefficients of the mass-weighted mapping are $\hat{S}_{ip} = S_{ip}^* \frac{m_p}{m_i}$. Forces at the nodes are computed by using equation (21)

$$f_i^n = -\sum_p D S_{ip}^* \sigma_p^n V_p^n + b_i \tag{41}$$

where σ_p^n is the stress at particle p, b_i is the body force at the node and V_p^n is the volume of that particle. The nodal acceleration is then computed using the nodal mass and force,

$$a_i^n = \frac{f_i^n}{m_i}. \tag{42}$$

With the acceleration computed at the node, nodal velocities can now be updated using a Euler-forward time stepping scheme,

$$v_i^{n+1} = v_i^n + a_i^n \delta t. \tag{43}$$

Using the updated nodal velocity, the velocity gradients are then computed at the particles using the gradient of the interpolating function,

$$\frac{\partial v_p^{n+1}}{\partial x} = \sum_p \frac{\partial \phi_{ip}}{\partial x} v_i^{n+1}. \tag{44}$$

The deformation gradient at particle p is updated as follows,

$$F_p^{n+1} = F_p^n + \frac{\partial v_p^{n+1}}{\partial x} F_p^n \delta t. \tag{45}$$

Stress is updated using the appropriate constitutive model. In this case using the velocity gradient, $\frac{\partial v_p^{n+1}}{\partial x}$, and Young's Modulus, E,

$$\sigma_p^{n+1} = \sigma_p^n + dt E \frac{\partial v_p^{n+1}}{\partial x}. \tag{46}$$

The particle velocity and position is updated by interpolating nodal values to the particles,

$$v_p^{n+1} = v_p^n + \sum_p \phi_{ip} a_i^n \delta t, \tag{47}$$

$$x_p^{n+1} = x_p^n + \sum_p \phi_{ip} v_i^{n+1} \delta t. \tag{48}$$

Finally the particle volumes are updated using the determinant of the deformation gradient, J, and the initial particle volume,

$$V_p^{n+1} = J V_p^0. \tag{49}$$

In one dimension the determinant of the deformation gradient is, $J = F_p^{n+1}$.

4 Null Space of the Mapping Vector

The general mapping matrix $\mathbf{S}_{ip}$ maps from all particles to nodes and so has $N+1$ rows and $m \times N$ columns, [15]. While the global SVD analysis used in [15] applies to the MPM method considered here, it is prohibitively expensive to use as an algorithmic approach for large meshes and more than one space dimension. For this reason while a global SVD method is a useful comparison tool in one space dimension it is important to consider local mappings from particles to each node and back again independently. In what follows the standard mapping matrix to the nodes defined by the coefficients S^*_{ip} is used. The approach could be applied equally well to any mapping matrix. In this local mapping case the mapping matrix reduces to a row-vector, but still has an SVD decomposition and a non-trivial null space. For example, let $\mathbf{c}$ be a vector in $\mathbb{R}^{2m}$. If $\mathbf{S}^*_{ip}\mathbf{c} = \mathbf{0}$, then we say that $\mathbf{c}$ is in the null space of $\mathbf{S}^*_{ip}$. We can determine the null space of $\mathbf{S}^*_{ip}$ by making use of its singular value decomposition, SVD [13]. Taking the SVD of $\mathbf{S}^*_{ip}$ gives the following decomposition,

$$\mathbf{S}^*_{ip} = \hat{U}_{svd}\hat{\mathbf{S}}\mathbf{V}^T, \tag{50}$$

where $\hat{U}_{svd}$ is a scalar that is assumed to have value one, $\hat{\mathbf{S}}$ is 1 by $2m$, and $\mathbf{V}$ is $2m$ by $2m$.

In the global case in contrast $\hat{\mathbf{U}}_{svd}$ is a matrix of size $(N+1)x(N+1)$, $\hat{\mathbf{S}}$ is $N+1$ by $m \times N$, and $\mathbf{V}$ is $m \times N$ by $m \times N$. While all the remaining analysis will be for the local nodal case, the full global form of the SVD as used by [15] will be used below to motivate the local approach used here.

In both local and global cases the matrix $\mathbf{V}$ is unitary, meaning that the columns are orthonormal [35]. In other words, if $\mathbf{v}_i$ and $\mathbf{v}_j$ are columns of the matrix V, then,

$$\mathbf{v}_i^T\mathbf{v}_j = \delta_{ij}, \tag{51}$$

where δ_{ij} is the Kroenecker delta. The columns of $\mathbf{V}$ are orthogonal, linearly independent and span the space $\mathbb{R}^{2m}$, [35]. The matrix $\hat{\mathbf{S}}$ is an 1 by $2m$ matrix of the form

$$\hat{\mathbf{S}} = \begin{bmatrix}\hat{\sigma}_i\ 0\ \ldots\ 0\ 0\ \ldots\ 0\end{bmatrix}. \tag{52}$$

Taking the matrix product of $\hat{\mathbf{S}}$ and $\mathbf{V}^T$, gives,

$$\hat{\mathbf{S}}\mathbf{V}^T = \left(\mathbf{V}\hat{\mathbf{S}}^T\right)^T = \left(\begin{bmatrix} \hat{\sigma}_i\mathbf{v}_1 \,\|\, 0*\mathbf{v}_2 \,|\, \ldots \,|\, 0*\mathbf{v}_{2m} \end{bmatrix}\right). \tag{53}$$

Consequently the column vectors $\mathbf{v}_2$ to $\mathbf{v}_{2m}$ span the null space of $\mathbf{S}^*_{\mathbf{ip}}$. Since the columns of $\mathbf{V}$ are orthogonal, they form a basis for $\mathbb{R}^{2m}$, which means that any vector $c \in \mathbb{R}^{2m}$ can be expressed as a linear combination of the columns of $\mathbf{V}$,

$$\mathbf{c} = c_1\begin{bmatrix}\mathbf{v}_1\end{bmatrix} + \underbrace{c_2\begin{bmatrix}\mathbf{v}_2\end{bmatrix} + c_3\begin{bmatrix}\mathbf{v}_3\end{bmatrix} + \cdots + c_{2m}\begin{bmatrix}\mathbf{v}_{2m}\end{bmatrix}}_{null(S_{ip^*})}. \tag{54}$$

A portion of $\mathbf{c}$ is thus in the null space of $\mathbf{S}^*_{ip}$ if $c_j \neq 0$ for some $j = 2,..,2m$. Using the inner product [35] allows the components of a vector that are in the null space to be found. As the vectors $\mathbf{v_i}$ are a basis it follows that

$$\mathbf{u}_{ip} = \sum_{i=1}^{2m} (\mathbf{v}_i \cdot \mathbf{u}_{ip}) \mathbf{v}_i. \tag{55}$$

where $\mathbf{u}_{ip}$ is defined by equation (19). A similar equation exists in the case of the full global form of the SVD matrix [15]. In the local case considered here the vectors $\mathbf{v_2}$ to $\mathbf{v_{2m}}$ span the null space of the mapping from particles to a node. From this it follows that a filtered form of the vector $\mathbf{u}_{ip}$ with the null space elements removed is denoted by $\mathbf{u_{ip}}^F$ and is defined by

$$\mathbf{u}^F_{ip} = (\mathbf{v}_1 \cdot \mathbf{u}_{ip}) \mathbf{v}_1, \tag{56}$$

or by using the SVD decomposition of $\mathbf{S}^*_{ip}$ to get

$$U_i = \hat{\sigma} \mathbf{v}_1^T \mathbf{u}_{ip}. \tag{57}$$

equation (19) in terms of the nodal value U_I as

$$\mathbf{u}^F_{ip} = \frac{U_i}{\hat{\sigma}} \mathbf{v}_1. \tag{58}$$

The portion of any given vector that lies in the null space of $\mathbf{S}^*_{ip}$ can be found as follows. As the vector $\mathbf{S}^*_{ip}$ can be written in terms of the singular value components,

$$\begin{aligned} \mathbf{S}^*_{ip} &= \pm 1 \left[\hat{\sigma}, 0, 0, \cdots, 0\right] \left[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \cdots, \mathbf{v}_n\right]^T \\ &= \pm 1 (\hat{\sigma}_i \mathbf{v}_1 + 0\mathbf{v}_2 + 0\mathbf{v}_3 + \cdots + 0\mathbf{v}_n)^T \\ &= \pm 1 \hat{\sigma}_i \mathbf{v}_1^T. \end{aligned} \tag{59}$$

Taking the dot product of the vector $\mathbf{S}^*_{ip}$ with itself gives,

$$\begin{aligned} \mathbf{S}^*_{ip} \cdot \mathbf{S}^*_{ip} &= \hat{U}_{svd} \hat{\mathbf{S}} \mathbf{V^T} (\hat{U}_{svd} \hat{\mathbf{S}} \mathbf{V^T})^\mathbf{T}, \\ &= \hat{U}_{svd} \hat{\mathbf{S}} \mathbf{V^T} \mathbf{V^T} \hat{\mathbf{S}}^\mathbf{T} \hat{U}^T_{svd}, \\ &= \hat{U}_{svd} \hat{\mathbf{S}} \hat{\mathbf{S}}^\mathbf{T} \hat{U}^T_{svd}, \\ &= (\hat{\sigma}_i)^2. \end{aligned} \tag{60}$$

From this it can be seen that the singular value for the mapping defined by $\mathbf{S}^*_{ip}$ is $\hat{\sigma} = \sqrt{(\mathbf{S}^*_{ip} \cdot \mathbf{S}^*_{ip})}$. Using the above two observations the column vector $\mathbf{v}_1$ can be found as follows,

$$\mathbf{v}_1^T = \frac{1}{\sqrt{(\mathbf{S}^*_{ip} \cdot \mathbf{S}^*_{ip})}} \mathbf{S}^*_{ip}. \tag{61}$$

Given the vector $\mathbf{v}_1$, the part of $\mathbf{u}_p$ that lies in the the null space of the mapping vector $\mathbf{S}^*_{ip}$ is then given by $\mathbf{u}_{ipnull}$ is then analogously as in the full SVD version and is given by

$$\mathbf{u}_{ipnull} = \mathbf{u}_{ip} - (\mathbf{v}_1 \cdot \mathbf{u}_{ip}) \mathbf{v}_1. \tag{62}$$

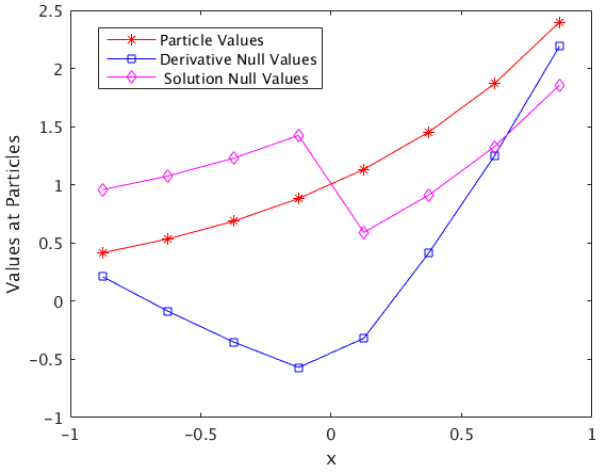

Fig. 1 Nullspaces of Particle to Node Solution and Derivative Mappings

or as

$$\mathbf{u}_{ipnull} = \mathbf{u}_{ip} - \frac{U_i}{\hat{\sigma}_i^2}(\mathbf{S}^*_{ip})^T \tag{63}$$

4.1 Numerical Example of Nullspace

The following example shows the null spaces for both the derivative mappings and the solution mappings, in the case of the original MPM method with a linear basis at the nodes and delta functions at the particles. The null space vectors for both the derivative mappings and the solution mappings are shown in the case of two cells adjacent to a node have 4 particles per cell. The cell width is 1 and the spatial domain of the problem goes from -1 to 1. The values at the particles is defined by the function,

$$u(x) = e^x. \tag{64}$$

Using the function a vector of particle values is computed where $u_p = u(x_p)$. The vector of mapping weights are computed about the node at position 0.0, $S_{ip} = \phi_i(x_p)$. Figure 1 shows the values at the particles and the portions of the particles values that lie in the null space of the nodal mapping equation (19) and of the derivative mapping equation (28) using linear basis functions at the nodes and delta functions at the particles. Figure 1 shows the single oscillation about the node. Other more oscillatory examples of null space components occur with symmetric positive S^*_{ip} values. In this case for each pair of S^*_{ip} values one solution component may be positive in one interval and the other solution value in the other interval may be negative. In the case of the mapping DS^*_{ip}, associated with derivatives, as the mapping values will have different signs the null space could still be oscillatory but may not necessarily change sign.

4.2 Mapping to particles re-inforces the null space

Equation (19) maps particle values to the nodes. At this point, the null space component of $\mathbf{u}_p$ has been removed by the nature of the mapping. It is at the next step in the computation that a null space component can be re-introduced. From equations (55 to 62) it follows that if for any $\mathbf{v_i}$ for $i = 2,...,2m$ the following is true

$$\mathbf{v_i^T}\hat{\mathbf{M}}\begin{bmatrix} U_{i-1} \\ U_i \\ U_{i+1} \end{bmatrix}, \quad \mathbf{v_i} \neq 0, \quad i = 2,...,2m \tag{65}$$

then the vector $\mathbf{u}_{ip}$, as defined by the mapping in equation (19), has a null space component.

Example As an example consider the case of one particle per cell with the particles numbered $0,1,2,3$, etc being midway between nodes, $i-1,i,i+1$ etc. Then with linear basis functions the mapping to nodal values is given by

$$\begin{aligned} U_{i-1} &= (u_0+u_1)/2, \\ U_i &= (u_1+u_2)/2, \\ U_{i+1} &= (u_2+u_3)/2. \end{aligned} \tag{66}$$

The new values at particles 1 and 2 are now given by the mapping

$$u_1^* = (u_0+2u_1+u_2)/4, \tag{67}$$
$$u_2^* = (u_1+2u_2+u_3)/4. \tag{68}$$

As the null space vector associated with the mapping from particles to nodes is

$$\mathbf{v} = [-1,1]^T. \tag{69}$$

The mapped values u_1^* and u_2^* only have a zero null space component if they are equal:

$$[u_1^*,u_2^*].[-1,1]^T = 0. \tag{70}$$

which requires that

$$(u_0+2u_1+u_2)-(u_1+2u_2+u_3) = 0 \tag{71}$$

or that

$$u_0+u_1-u_2-u_3 = 0 \tag{72}$$

holds. A similar result is obtained if gradient values are calculated at the points 1 and 2 using the nodal values.

5 Local Removal of the Nullspace Noise

As was shown in [15,14] using a full singular value decomposition across the whole grid for the removal of null space noise works well for small one-dimensional problems, but is computationally expensive, as the computational complexity of generating the matrix $\mathbf{V}$ with a singular value decomposition is $O((mN)^3)$ [13]. This method calculates a filtered full particle vector across the whole grid as denoted by $\mathbf{U}_p^{filter}$ in terms of the full vector of all the particle values $\mathbf{U}_p^{full}$.

$$\mathbf{U}_p^{filter} = \sum_i (\mathbf{V_i} \cdot \mathbf{U}_p^{full}) \mathbf{V}_i, \tag{73}$$

where the vectors $\mathbf{V}_i$ are the vectors that are not in the null space as calculated from the full SVD decomposition [15].

It is thus desirable to have a method for removing the null space noise that works locally at each node, X_i with less complexity than the full SVD method. If we consider the interval $[X_{i-1}, X_{i+1}]$, then using equation (56) it is possible to define a particle vector in this interval using the null space associated with node X_i by

$$\mathbf{u}_{ip}^{AF} = (\mathbf{v}_{1,i} \cdot \mathbf{u}_{ip}) \mathbf{v}_{1,i}, \tag{74}$$

where $\mathbf{v}_{1,i}$ is the vector $\mathbf{v}_1$ associated with the mapping to node X_i. and a particle vector in the interval $[X_i, X_{i+2}]$ using the null space associated with node $i+1$ by

$$\mathbf{u}_{(i+1)p}^{AF} = (\mathbf{v}_{1,i+1} \cdot \mathbf{u}_{(i+1)p}) \mathbf{v}_{1,i+1}, \tag{75}$$

or by using equation (63) to get

$$\mathbf{u}_{ip}^{AF} = \frac{U_i}{(\mathbf{S}_{ip}^* \cdot \mathbf{S}_{ip}^*)} (\mathbf{S}_{ip}^*)^T. \tag{76}$$

$$\mathbf{u}_{(i+1)p}^{AF} = \frac{U_{i+1}}{(\mathbf{S}_{(i+1)p}^* \cdot \mathbf{S}_{(i+1)p}^*)} (\mathbf{S}_{(i+1)p}^*)^T. \tag{77}$$

From this and equation (17) we see that the x_pth component obtained by adding equations (76) and (77) is

$$\left[\mathbf{u}_{ip}^{AF} + \mathbf{u}_{(i+1)p}^{AF}\right]_p = \frac{U_i}{SN_i} \frac{\phi_i(x_p)}{W_i} + \frac{U_{i+1}}{SN_{i+1}} \frac{\phi_{i+1}(x_p)}{W_{i+1}} \tag{78}$$

where $x_p \in I_i$, $SN_i = (\mathbf{S}_{ip}^* \cdot \mathbf{S}_{ip}^*)$, $SN_{i+1} = (\mathbf{S}_{(i+1)p}^* \cdot \mathbf{S}_{(i+1)p}^*)$ and W_i, W_{i+1} are defined by equation (17). Hence the filtered values correspond to a modified form of linear interpolation. However as the example in Section 4.1 illlustrates the linear interpolation mapping can still re-introduce null space errors. The reason for this is that parts of the vector $\mathbf{S}_{ip}$ may lie in the null space of $\mathbf{S}_{(i+1)p}$ and vice-versa. For this reason a second mapping is done back to the grid nodes and then linear interpolation used again.

This method thus takes a different approach than the SVD method to removing the null space components. The key idea in the local method is to

– first map particle values to the nodes, using equation (19) to compute U_i and U_{i+1}.
– and then interpolate values from nodes to particles as in equations (31,32). When this happens, a null space component is introduced by this calculation.
– The newly computed solution values are again mapped back to the nodes, and
– These new nodal values are interpolated back to the particles.

In the case of mapping velocities to the nodes and then forming velocity gradients at any point x_p in an interval I_i the value of the derivative instead of being

$$\frac{\partial v}{\partial x}(x_p) = \frac{V_{i+1} - V_i}{h} + (1 - 2\alpha_p)\frac{h}{2}V_{xx}(x_p) + \text{h.o.t} \tag{79}$$

is now defined for the new method by

$$\begin{aligned}\frac{\partial v}{\partial x}(x_p) = (1-\alpha_p)S_i^+\left(\frac{(V_{i+1}-V_i)}{h} + S_i^-\frac{(V_i - V_{i-1})}{h}\right) \\ + \alpha_p\left(S_{i+1}^+(\frac{(V_{i+2}-V_{i+1})}{h} + S_{i+1}^-\frac{(V_{i+1}-V_i)}{h}\right)\end{aligned} \tag{80}$$

where α_p is defined by equation (33) or (34) and

$$S_i^+ = \sum_{p\in I_i} S_{ip}^* \tag{81}$$

$$S_i^- = \sum_{p\in I_{i-1}} S_{ip}^*. \tag{82}$$

The error of this new method is given by a Taylor's series analysis as

$$\begin{aligned}\frac{\partial v}{\partial x}(x_p) - (1-\alpha_p)\left(S_i^+\frac{(V_{i+1}-V_i)}{h} + S_i^-\frac{(V_i-V_{i-1})}{h}\right) \\ + \alpha_p\left(S_{i+1}^+\frac{(V_{i+2}-V_{i+1})}{h} + S_{i+1}^-\frac{(V_{i+1}-V_i)}{h}\right) \\ = i((1-\alpha_p)(S_i^+ - S_i^-) + \alpha_p(S_{i+1}^+ - S_{i+1}^-))\frac{h_i}{2}V_{xx}(x_p).\end{aligned} \tag{83}$$

The effect of this approach is thus not to increase the underlying accuracy but to broaden out the stencil used to create the derivatives at the particles. This approach can be used with both solution and gradient values, but is used here with the velocity gradient calculation.

5.1 Multidimensional extensions

As this approach only makes use of the standard mappings from particles to nodes and back again, there is no conceptual problem in extending it to the use of MPM in two and three space dimensions. While there is further work to be done, there is clearly room for combining the approaches suggested here with the improved grid crossing approaches such as [37] and the locking approach of Mast [26]. In particular the latter approach with its careful tensor-based decomposition ideas may have important implications for multi-dimensional extensions of the approach suggested here. What may well be the case however is that it is a combination of all these approaches that will be important.

6 Computational Experiments

6.1 Model Problem

$$\sigma = P = E\frac{\partial u}{\partial X} = E(F-1), \tag{84}$$

where E is the Young's modulus. The rate of change of stress is then computed as,

$$\dot{\sigma} = E(\dot{F}), \tag{85}$$
$$= E(lF), \tag{86}$$

where l is the velocity gradient in the spatial description.

The problem considered is a 1D bar problem, following similar examples in [31]. The analytic solutions for displacement and velocity defined in the material description are:

$$u(X,t) = Asin(2\pi X)sin(c\pi t), \tag{87}$$
$$\frac{\partial u}{\partial t} = Ac\pi sin(2\pi X)cos(c\pi t), \tag{88}$$

where $c = \sqrt{E/\rho_o}$ and A is the maximum displacement. The constitutive model is defined in Equation 84 and the body force is,

$$b(X,t) = 3A(c\pi)^2 u(X,t). \tag{89}$$

The initial spatial discretization uses two evenly spaced particles per cell with the spatial domain being $[0,1]$. The periodic nature of the analytic solution means that both periodic boundary conditions and zero Dirichlet boundary conditions are both appropriate. The initial conditions for the updated Lagrangian description of the particles are:

$$F = 1, \tag{90}$$
$$x_p = X_p^0, \tag{91}$$
$$V_p = V_p^0. \tag{92}$$

For the 1d bar problem the cell width is $h = 10^{-2}$, the material density is $\rho_0 = 1$, Young's modulus is $E = 10^3$, maximum displacement is $A = 3 \times 10^{-3}$, and the timestep is $dt = 10^{-5}$. It should be noted that with the use of the above parameters no particles will cross from one cell to another.

6.2 MPM Methods used in Experiments

Initial experiments were undertaken with the standard MPM method using linear basis functions at the nodes and delta functions at the particles to provide a baseline calculation. Two modifications were then made to this standard MPM method in order to apply the new corrected derivatives. First the mapping of mass and velocity

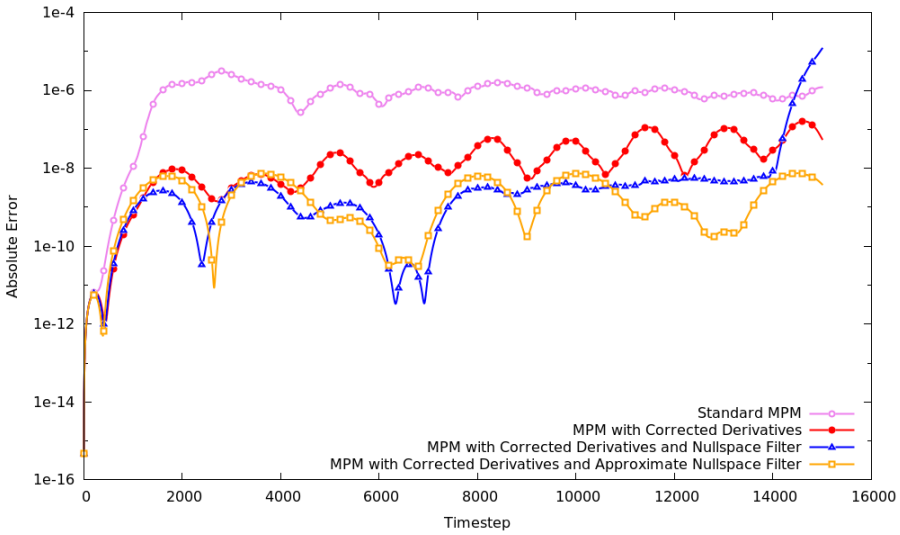

Fig. 2 1d Bar, Error vs Timesteps for MPM, Corrected Derivatives MPM and Corrected Derivatives MPM with Full and Approximate Nullspace Filter

to the nodes was modified to use the mapping function as defined by equation (19). Similarly the modified derivative weight function defined by equation (27) replaces the standard derivative weight function when computing the internal forces as found in Equation (41) of the MPM algorithm. From Figure 2 it can be seen that the application of the corrected derivatives weight functions to the internal force calculation greatly increased the accuracy of the method. Not many timesteps were needed before a difference in accuracy between the two different versions of MPM could easily be noted.

To further increase the accuracy of the method the null space filter is used as described in Section 5. The filter is applied after velocity gradients are computed as done in Equation (80). The problem was then solved with the improved corrected derivatives MPM defined by equation (27) and corrected derivatives MPM with the applied full SVD null space filter as described in Section 5. Finally the simulations was run comparing corrected derivatives MPM with corrected derivatives MPM with the approximated null space filter that was described in Section 5. Figure 2 shows a comparison between these four methods. From the plot it can be seen that the applied null space filter in both the SVD form and the approximate form appear to initially improve the accuracy of the corrected derivatives MPM. It is noted that beyond 14000 timesteps there is a rapid increase in the error of the corrected derivatives MPM with the full global matrix SVD null space filter. Further work still needs to be done to explore the cause of this error increase.

In these simulations it is observed that the null space filter removes the null space noise that is introduced when velocity gradients are computed at particles. The null space noise produces oscillations that are visible in the computed velocities of the particles. Figure 3 shows the computed velocity of each particle. On the right hand side

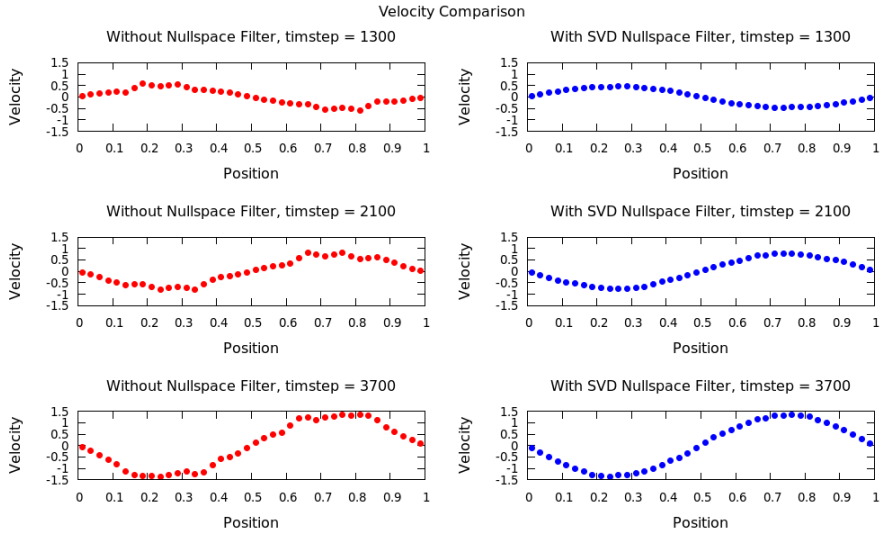

Fig. 3 1d Bar, Particle Velocities for Corrected Derivatives MPM and Corrected Derivatives MPM with Global Nullspace Filter

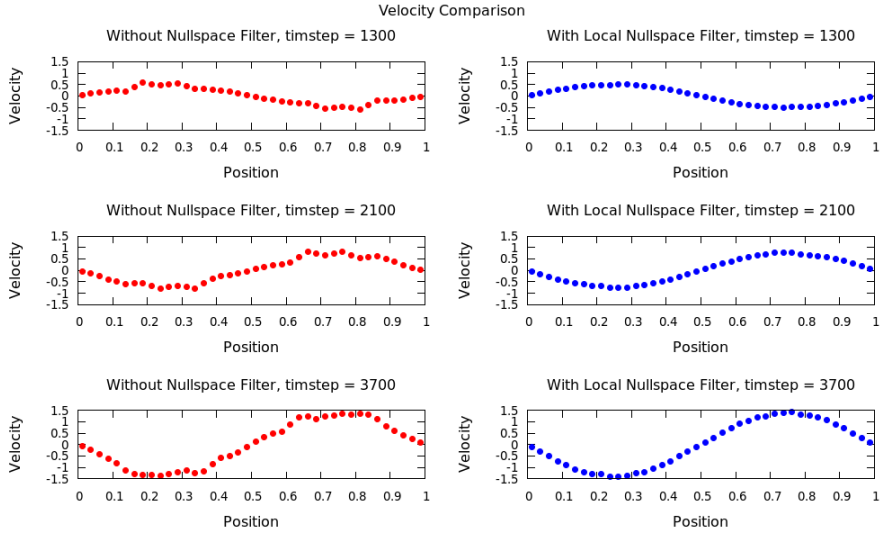

Fig. 4 1d Bar, Particle Velocities for Corrected Derivatives MPM and Corrected Derivatives MPM with Local Nullspace Filter

of the figure are velocity plots at three different timesteps where corrected derivatives MPM was used and on the left are velocity values where the global null space filter was applied. Figure 4 shows the same results but using the local filter. From these plots it can be seen that both the null space filter remove the oscillations that can arise due to the null space noise. This is clearly an area that needs further research.

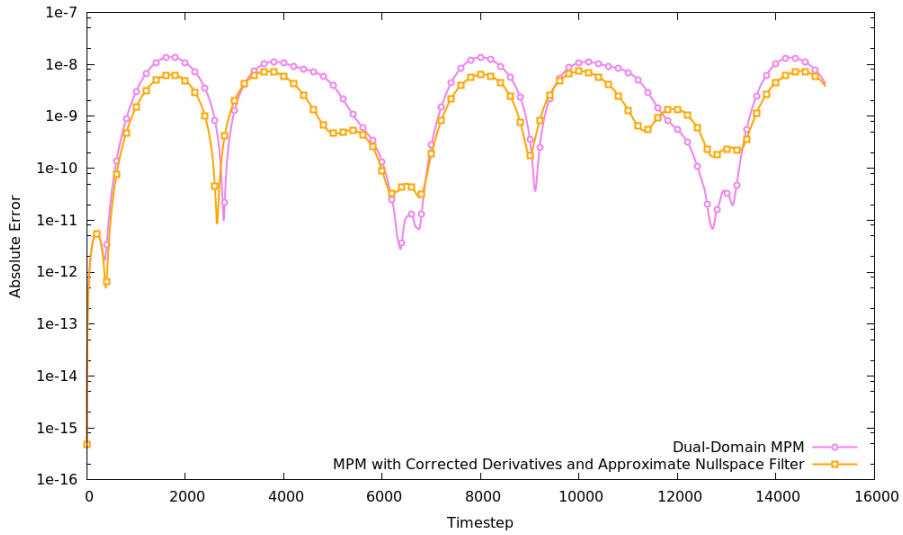

Fig. 5 1d Bar, Error vs Timesteps for MPM, Dual-Domain MPM and Corrected Derivatives MPM with Approximate Nullspace Filter

The dual domain MPM [37] increases the accuracy of MPM by improving the method of computing weight function gradients. The improved weight function gradients, as described in [37], are applied to the internal force calculation and the velocity gradient calculation steps of the MPM algorithm described in this paper. The updated version of MPM using the improved weight function gradient calculations is then used to solve the example problem described in this section. Figure 5 shows a comparison of the dual domain method compared to the corrected derivatives version with the approximated null space filter as proposed in this paper. From the plot it can be seen that the two methods are comparable in accuracy.

Further work needs to be done on the methods proposed in this paper. Currently the use of the different null space filters has been applied to the problem where particle do not cross from cell boundaries. The null space filters have not currently shown improvements in accuracy when cell crossings are involved.

7 Summary

The use of an-SVD based approach has made it possible to construct a first attempt at a local filter that addresses the challenge of null spaces in the MPM method. In preliminary experiments the new filter shows promise, but much more research is required before such approaches may be used in multi-space dimensional production codes.

Acknowledgements We would like to thank the referees for their thoughtful and helpful comments that have helped to greatly improve this paper. This research was primarily sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-12-2-0023. The views

and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The authors would like to thank ARL for their support and their colleague Mike Kirby for suggesting the use of the global SVD approach. The final publication is available at Springer via http://dx.doi.org/10.1007/s40571-016-0134-3

References

1. N.R. Aluru, *A reproducing kernel method for meshless analysis of microelectromechanical systems* Computational Mechanics, 23 (1999), pp. 324– 338.
2. S. Bardenhagen, *Energy conservation error in the material point method for solid mechanics*, Journal of Computational Physics, 180 (2002), pp. 383 – 403.
3. S. Bardenhagen and E. Kober, *The generalized interpolation material point method*, Computer Modeling in Engineering and Science, 5 (2004), pp. 477 – 495.
4. T. Belytschko, Y. Guo, W. Kam Liu and S. Ping Xiao, A unified stability analysis of meshless particle methods, International Journal for Numerical Methods in Engineering, Vol. 48, no. 9, 1359–1400,2000.
5. T. Belytschko, Y. Krongauz, J. Dolbow, and C. Gerlach, *On the completeness of meshfree particle methods*, International Journal for Numerical Methods in Engineering, 43 (1998), pp. 785–819.
6. T. Belytschko and S. Xiao, *Stability analysis of particle methods with corrected derivatives*, Computers and Mathematics with Applications, 43 (2002), pp. 329 – 350.
7. J. Brackbill and G. Lapenta, *A Method to Supress the Finite-Grid Instability in Plasma Simulations*, Journal of Computational Physics, 114 (1994), pp. 77–84.
8. J. Brackbill, *The ringing instability in particle-in-cell calculations of low-speed flow*, Journal of Computational Physics, 75 (1988), pp. 469 – 492.
9. J. Brackbill, D. Kothe, and H. Ruppel, *Flip: A low-dissipation, particle-in-cell method for fluid flow*, Computer Physics Communications, 48 (1988), pp. 25 – 38.
10. J.U. Brackbill, *On Energy and Momentum Conservation in Particle-in-Cell Simulation*, ArXiv e-prints,1510.08741, October 2015.
11. L. Chen, A.B. Langdon, C.K. Birdsall *Reduction of the Grid-Effects in Simulation Plasma's* Journal of Computational Physics, 14 (1974), pp. 200 – 222.
12. G. A. Dilts, *Moving-least-squares-particle hydrodynamicsi. consistency and stability*, International Journal for Numerical Methods in Engineering, 44 (1999), pp. 1115–1155.
13. G. H. Golub and C. F. V. Loan, *Matrix Computations*, The John Hopkins University Press, third ed., 1996.
14. C.E. Gritton, *Ringing Instabilities in Particle Methods*, M.S.Thesis in Computational Engineering and Science, August 2014, School of Computing, University of Utah.
15. C.E. Gritton, M. Berzins and R. M. Kirby, *Improving Accuracy In Particle Methods Using Null Spaces and Filters*, Proceedings of the IV International Conference on Particle-Based Methods - Fundamentals and Applications, pp 202-213,2015, Ed(s)E. Onate, M. Bischoff, D.R.J. Owen, P. Wriggers, and T. Zohdi", CIMNE, September 2015,Barcelona, Spain, ISBN 978-84-944244-7-2. http://congress.cimne.com/particles2015/frontal/doc/E-book_PARTICLES_2015.pdf,
16. F. H. Harlow*The particle-in-cell method for fluid dynamics*, Methods in Computational Physics, 3 (1964).
17. Gerhard A. Holzapfel *Nonlinear Solid Mechanics: A Continuum Approach for Engineering* Wiley 2000 470 pages April 2000. 2000 ISBN: 978-0-471-82319-3
18. G. R. Johnson and S. R. Beissel, *Normalized smoothing functions for sph impact computations*, International Journal for Numerical Methods in Engineering, 39 (1996), pp. 2725–2741.
19. Y. Krongauz and T. Belytschko, *Consistent pseudo-derivatives in meshless methods*, Computer Methods in Applied Mechanics and Engineering, 146 (1997), pp. 371 – 386.
20. G. Lapenta *Exactly Energy Conserving Implicit Moment Particle in Cell Formulation.* eprint arXiv:1602.06326, 02/2016 J. Computat. Phys., submitted
21. A. Langdon, *Effects of the spatial grid in simulation plasmas*, Journal of Computational Physics, 6 (1970), pp. 247 – 267.
22. W. K. Liu, S. Jun, S. Li, J. Adee, and T. Belytschko, *Reproducing kernel particle methods for structural dynamics*, International Journal for Numerical Methods in Engineering, 38 (1995), pp. 1655–1679.

23. J. D. Logan, *Applied Partial Differential Equations*, Springer-Verlag, New York, second ed., 2004.
24. E. Love, E. and D.L. Sulsky, An energy-consistent material-point method for dynamic finite deformation plasticity. Int. J. Numer. Meth. Engng., 2006, 65: 16081638.
25. E. Love and D. L. Sulsky, *An unconditionally stable, energymomentum consistent implementation of the material-point method*, Computer Methods in Applied Mechanics and Engineering 195, 2006,3903–3925.
26. C. M. Mast, P. Mackenzie-Helnwein, P. Arduino, G. R. Miller, and W. Shin. *Mitigating kinematic locking in the material point method.* J. Comput. Phys. 231, 16 (June 2012), 5351-5373.
27. J.J. Monaghan, *SPH without a Tensile Instability*, Journal of Computational Physics, Volume 159, Issue 2, 10 April 2000, Pages 290-311,
28. M. Ortiz, *A note on energy conservation and stability of nonlinear time-stepping algorithms*, Computers & Structures, Volume 24, 1, 1986, Pages 167-168.
29. P. Randles and L. Libersky, *Smoothed particle hydrodynamics: Some recent improvements and applications*, Computer Methods in Applied Mechanics and Engineering, 139 (1996), pp. 375 – 408.
30. A. Sadeghirad, R.M. Brannon and J.E. Guilkey Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces, International Journal for Numerical Methods in Engineering, Vol. 95, no. 11, pages = 928–952, 2013.
31. M. Steffen, P.C. Wallstedt, J.E. Guilkey, R.M. Kirby and M. Berzins, Examination and Analysis of Implementation Choices within the Material Point Method (MPM), Computer Modeling in Engineering & Sciences, Vol.31, No 2, pp.107–127, 2008.
32. M. Steffen, R. M. Kirby, and M. Berzins, *Analysis and reduction of quadrature errors in the material point method (mpm)*, International Journal for Numerical Methods in Engineering, 76 (2008), pp. 922–948.
33. A. Stomakhin, C. Schroeder, L. Chai, J.Teran and A. Selle, A Material Point Method for Snow Simulation, ACM Trans. Graph.July 2013, Vol 32, No.4 pp.102:1–102:10,ACM,New York, NY, USA.‘
34. D. Sulsky, Z. Chen, and H. Schreyer, *A particle method for history-dependent materials*, Computer Methods in Applied Mechanics and Engineering, 118 (1994), pp. 179 – 196.
35. L. N. Trefethen and I. David Bau, *Numerical Linear Algebra*, SIAM, 1997.
36. P. C. Wallstedt and J. E. Guilkey. *An evaluation of explicit time integration schemes for use with the generalized interpolation material point method.* J. Comput. Phys. 227, 22 (November 2008), 9628-9642. DOI=http://dx.doi.org/10.1016/j.jcp.2008.07.019 ‘
37. Duan Z. Zhang, Xia Ma, Paul T. Giguere, *Material point method enhanced by modified gradient of shape function*, Journal of Computational Physics, Volume 230, Issue 16, 10 July 2011, Pages 6379-6398,