Preprint ONLY
(submitted for publication)

Computational Error Estimation for The Material Point Method

Martin Berzins[1]

Received: date / Accepted: date

Abstract A common feature of many methods in computational mechanics is that there is often a way of estimating the error in the computed solution. The situation for computational mechanics codes based upon the Material Point Method is very different in that there has been comparatively little work on computable error estimates for these methods. This work is concerned with introducing such an approach for the Material Point Method. Although it has been observed that spatial errors may dominate temporal ones at stable time steps, recent work has made more precise the sources and forms of the different MPM errors. There is then a need to estimate these errors through computable estimates of the different errors in the material point method. The approach used involves linearity-preserving extensions of existing methods, which allows estimates of the different spatial errors in the Material Point Method to be derived based upon nodal derivatives of the different physical variables in MPM. These derivatives are then estimated using standard difference approximations calculated on the background mesh. The use of these estimates of the spatial error makes it possible to measure the growth of errors over time. A number of computational experiments are used to illustrate the performance of the computed error estimates for both the original MPM method and the GIMP method, when modified to preserve linearity. Finally the form of the computed estimates also makes it possible to identify the order of the accuracy of the methods in space and time. For these methods including the linearity preservation is clearly beneficial, as regards accuracy while not changing the preference for GIMP over MPM

Keywords Material Point Method, Spatial Estimation, Time Integration Error, Particle in Cell Method

[1] Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA
mb@sci.utah.edu

1 INTRODUCTION

A notable feature of many computational mechanics codes, particularly finite element codes, since the 1980s has been the availability of computable estimates of the error and the use of these estimates in accuracy checks, adaptive mesh refinement and element order selection. The Material Point Method (MPM) has proved to be invaluable for many very challenging problems. However in many ways the method is not as far advanced as such finite element methods in terms of accuracy and stability analysis and computable error estimates for errors in both space and their evolution in time. As it is observed that that the spatial error dominates for cases in which the calculation is stable [16], this suggest that emphasis should be on the spatial error.

While there is an extensive literature on Finite Element Error estimation, for standard meshes e.g [26], in the case of the methods closest to the Material Point Method the Particle Finite Element Method [9] there is comparatively little work on error estimates. One exception is [20] who look at the different approaches for mapping from particles to quadrature node. In general the Finite Element error estimators provide estimates in a particular norm, such as the energy norm. The approach adopted here is to provide estimates at every computational mesh point in both space and time, for a complex nonlinear system and following early work in this area [7,8].

This work continues the analysis of MPM methods after previous work on gas dynamics [21], null spaces and linearity preservation [11], stability [3], time integration [4], energy conservation [5] and is an extension of a previous conference paper [6] which introduced a key idea used here for estimating the mapping errors from particles to mesh and from mesh to particles without demonstrating its effectiveness in a full MPM simulation. The focus here is on a careful prototype demonstration of error estimation in MPM on a well-studied model problem, so as to provide an incentive for further study for more complex problems in higher space dimensions.

Section 2 describes related work on error estimation related to MPM. Section 3 describes the MPM method in two forms: the original MPM method of [17,**?**] and what is regarded as the improved GIMP method [2] and describes the application of these methods to a simple example problem. A notable difference from most applications of these methods is that they are modified so as to preserve linearity in mapping from particles to nodes and back again [11]. A full description of the MPM method, its errors and how they evolve in time is provided in Section 4. Simple computational estimates of the different mapping errors in MPM are derived in Section 5, based on the ideas in [6] and are used to define the order of accuracy of these errors. In Section 6 the two methods are compared on a simple model problem and the error estimating approach is applied to that problem which is widely used in the MPM literature, and which illustrates the potential of this approach. In Section 7 a discussion of the approach is given and in Section 8 the observed order of accuracy is shown when the underlying mesh spacing is varied. Concluding remarks are made together with comments on the extension of these ideas to multiple space dimensions are provided in Section 9

2 BACKGROUND ON EXISTING MPM ERROR WORK

Many of the key features of MPM methods are covered in the two recent surveys. The first of Vaucorbeil et al. [23] references the work by Wallstedt and Guilkey [25], Tran and Berzins [21] , Steffen et al. [14],[16] and Gritton and Berzins [11]. In the area of accuracy Steffen et al. look carefully at space and time errors and what degree of accuracy is seen on example problems, thus paving the way for many of the developments that followed such as the use of higher order methods in both space and time. Improved time integration methods are considered by Wallstedt and Guilkey [25]. The relationship between MPM time integration and symplectic time integration methods is considered by Berzins [4]. Such symplectic methods have good conservation properties and the well known Stormer-Verlet method has third order accuracy locally [5].

The second survey of Solowski et al [13] also describes the same methods but also points out potentially relevant analysis of particle in cell methods such as the book by Grigoriev et al. [10] and one of the first rigorous attempts to provide a solid theoretical basis for particle methods was due to Raviart [12] who provide a mathematical introduction to the vortex numerical method. All these approaches provide insight into the errors of particle methods but none of them provides computable error estimates. Perhaps the closest approach is the adaptive meshing work of Tan and Nairn [19] who use quadratic expansions to calculate second derivatives as an indicator for mesh refinement parameter as well as to estimate a mass lumping error.

This is an extended version of a conference paper [6] that described the approach that is fleshed out here in a very elementary form. The challenge with error estimation of particle methods is that such error estimates either explicitly or implicitly require spatial derivative information. Such particle information is very challenging to compute using only arbitrarily spaced particles. One major advantage of MPM over some other equally effective particle methods, such as SPH, is its background mesh that allows for the construction of derivative values on that mesh, based upon the solution values that are mapped to that mesh. These derivatives may be used directly in a Taylor series expansion on that mesh or mapped to particle positions to be used in series expansions about these points.

The central idea here is to demonstrate the potential of this approach using a relatively simple and well-studied problem. Nevertheless the basic ideas apply not only to such simple problems but extend relatively easily to more complex cases in higher dimensions. Finally although this error estimation is done in the context of MPM, many of the same ideas also apply to particle in cell methods as they share similar mappings of particles to grid and grid to particles.

3 MPM MODEL PROBLEM AND METHOD

The description of MPM used here follows [11,5] in that the model problem used here is a pair of equations connecting velocity v, displacement u and density ρ:

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

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

with a linear stress model $\sigma = E\frac{\partial u}{\partial x}$ for which Young's modulus, E, is constant, a body force b, which is initially assumed to be zero, and with appropriate boundary and initial conditions. For convenience a mesh of equally spaced $N+1$ fixed nodes X_i with intervals $I_i = [X_i, X_{i+1}]$, on on the interval $[a,b]$ is used where

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

$$h = X_i - X_{i-1}. \tag{4}$$

These fixed nodes are referred to as the i points. Nothing in the method derivation is restricted to such meshes however. It will also be assumed that periodic boundary conditions exist in that

$$\sigma(a)v(a) = \sigma(b)v(b) \tag{5}$$

together with appropriate initial conditions. While such conditions are typical for the model problem used to illustrate the results the error estimation approach used here may be extended to different boundary conditions. While the analysis of MPM for time integration error and energy conservation uses the model problem above it does apply more generally and in multiple space dimensions with a few obvious modifications, as will be discussed in Sect. 8. The computed solution at the pth particles will be written as $u_p^n = u(x_p^n, t^n)$. Suppose that there are np particles in total. The calculation of the internal forces in MPM at the nodes requires the calculation of the volume integral of the divergence of the stress [25] using

$$f_i^{int} = -\sum_p D_{pi}(x_p^n)\sigma_p V_p \tag{6}$$

The subscript pi represents a mapping from particles p to node i while the subscript ip represents a mapping from nodes i to particles p. The negative sign arises as a result of using integration by parts [11]. The mass at node i is defined by

$$m_i = \sum_p m_p S_{pi}(x_p^n) \tag{7}$$

It is important to note that the coefficients $D_{pi}(x_p^n)$ and $S_{pi}(x_p^n)$ (which here will be abbreviated to D_{pi}^n and S_{pi}^n, depend explicitly on the background mesh and the particle positions and that they also be chosen to reproduce derivatives of constant and linear functions exactly [11]. The initial volume of the particles is uniform for the n_p particles in an interval, but again nothing in the derivation explicitly requires this. Indeed after the first time step the particles will be no longer uniformly distributed and over many times teps their positions will vary greatly. The particle volumes are defined using the deformation gradient, F_p^n, and the initial particle volume,V_p^0,

$$V_p^n = F_p^n V_p^0, \text{ where } V_p^0, = \frac{h}{n_p}, \textit{ where } F_p^0 = 1 \tag{8}$$

From (7) the form of acceleration equation in the MPM method in this case is

$$a_i(t) = \frac{-1}{m_i} \sum_p D_{pi}(x_p(t)) \sigma_p(t) F_p(t) V_p^0 \tag{9}$$

While for the simple example here uniform masses are assumed there is nothing in the derivation that precludes non-uniform masses. The equation to update velocity at the nodes, as denoted by v_i^n is then given by

$$\dot{v}_i = a_i \tag{10}$$

The equation for the update of the particle velocities is then

$$\dot{v}_p = a_p \tag{11}$$

where the value of the acceleration at a point x_p^n is given by interpolation based upon nodal values of acceleration

$$a_p = \sum_i S_{ip}(x_p(t)) a_i \tag{12}$$

The equation for the particle position updates is

$$\dot{x}_p = v_p \tag{13}$$

The update of the deformation gradients is given using

$$\frac{\partial v}{\partial x}(x_p(t)) = \sum_i D_{ip}(x_p(t)) v_i \tag{14}$$

The deformation update equation is

$$\dot{F}_p = \frac{\partial v}{\partial x}(x_p(t), t) F_p \tag{15}$$

While the stress update equation is, using the appropriate constitutive model and Young's Modulus, E,

$$\dot{\sigma}_p = E \frac{\partial v}{\partial x}(x_p(t)) \tag{16}$$

In the above general formulation the coefficients S_{ip} and D_ip are determined by the choice of MPM method and will now be given for the original MPM method and the GIMP method, see [27].

3.1 Original MPM Method Coefficients

In the case of the original MPM method [17,18] the method coefficients are given by

$$S_{ip} = \begin{cases} \frac{x_p - X_{i-1}}{X_i - X_{i-1}}, X_{i-1} \le x_p \le X_i \\ \frac{X_{i+1} - x_p}{X_{i+1} - X_i}, X_i \le x_p \le X_{i+1} \\ 0 \; otherwise \end{cases} \tag{17}$$

$$D_{ip} = \begin{cases} \frac{1}{X_i - X_{i-1}}, X_{i-1} \le x_p \le X_i \\ \frac{-1}{X_{i+1} - X_i}, X_i \le x_p \le X_{i+1} \\ 0 \; otherwise \end{cases} \tag{18}$$

3.2 GIMP MPM Method Coefficients

In the case of the GIMP MPM method [2] the method coefficients are given by P.72 of[27]

$$S_{ip} = \begin{cases} 0 | x_p - X_i | \ge h + l_p \\ \frac{(h + l_p + (x_p - X_i))^2}{4hl_p}, -h - l_p \le x_p - X_i \le -h + l_p \\ 1 + \frac{(x_p - X_i)}{h}, -h + l_p \le x_p - X_i \le -l_p \\ 1 - \frac{((x_p - X_i))^2 + l_p^2}{2hl_p}, -l_p \le x_p - X_i \le +l_p \\ 1 - \frac{(x_p - X_i)}{h}, l_p \le x_p - X_i \le -l_p \\ \frac{(h + l_p - (x_p - X_i))^2}{4hl_p}, L - l_p \le x_p - X_i \le h + l_p \end{cases} \tag{19}$$

In this case h is the mesh spacing and l_p is the initial particle width which here is V_p^0 as defined in equation (8).

$$D_{ip} = \begin{cases} 0 | x_p - X_i | \ge h + l_p \\ \frac{(h + l_p + (x_p - X_i))}{2hl_p}, -h - l_p \le x_p - X_i \le -h + l_p \\ \frac{1}{h}, -h + l_p \le x_p - X_i \le -l_p \\ -\frac{(x_p - X_i)}{hl_p}, -l_p \le x_p - X_i \le +l_p \\ -\frac{1}{h}, l_p \le x_p - X_i \le -l_p \\ \frac{(h + l_p - (x_p - X_i))}{2hl_p}, h - l_p \le x_p - X_i \le h + l_p \end{cases} \tag{20}$$

3.3 Linearity Preserving MPM and GIMP

In order to estimate the errors of the different parts of MPM methods it is important to define the accuracy of the mappings defined by the S and D matrices and the transposes S^T and D^T. The approach used here with both MPM and GIMP is that of [11] which is used to modify the mapping S_{ip} so that it is linearity preserving i.e.

$$\sum_p S_{pi} = 1 \tag{21}$$

and

$$\sum_p S_{pi} x_p = X_i. \tag{22}$$

and for the D matrix which differentiates, the exact differentiation of a linear function is required, giving the equations

$$\sum_p D_{pi} = 0 \tag{23}$$

and

$$\sum_p D_{pi} x_p = 1. \tag{24}$$

The same operations are also applied to the transpose matrices S^T and D^T. In all the experiments below the GIMP method is used as modified for linearity preservation [11]. The original form of MPM will be referred to as MPM, while the linearity preserving form will be referred to as LPMPM. While numerous experiments have shown that GIMP is to be preferred over MPM [2,16,25] (and this will be confirmed below), the additional of linearity preservation to MPM to give the LPMPM method improves its performance greatly, as will also be shown below in Sect. 6, though not to the point where it is to be preferred over GIMP.

4 Stress Last MPM and Global Errors in Space and Time

For any choice of the mappings $S_i p$ and $D_i p$ it is now possible to define the equations used to advance the method and estimate its errors. In solving the system of equations defined above by equations (6) to (16) one standard approach used is to order the equations in a certain order and then to solve them in turn using explicit methods. Differences in how the equations are solved corresponds to whether or not the stress is updated first or last in a time step, a choice that is discussed at length by [1]. These two different choices are related to the use of the semi-implicit Euler A or B method [3]. Following Bardenhagen [1] it is preferable to increment stress last. In this case it is assumed that at time t^n a consistent set of particle positions x_p^n, velocities v_p, stresses σ_p^n and deformation gradients F_p^n are available. The description here of the method and the sources of error follows that in [4]. It is also assumed that the GIMP method is used for spatial discretization.

In contrast to the local errors described in [6] the analysis here is concerned with the evolution of the full global error.

The nodal velocity v_i is calculated using a standard MPM momentum based mapping.

$$v_i^n = \sum_p S_{pi}^n \frac{m_p}{m_i} v_p^n \tag{25}$$

with an associated nodal velocity error $Ev_i^n = v_{i,true}^n - v_i^n$ which is defined in terms of the existing particle errors Ev_p^n and the interpolation error Ev_{pi}^n by

$$Ev_i^n = \sum_p S_{pi}^n \frac{m_p}{m_i} Ev_p^n + Ev_{pi}^n \tag{26}$$

where Ev^n_{pi} is the interpolation or mapping error associated with the coefficients S^n_{pi} which will be defined as follows:

$$Ev^n_{pi} = \tilde{v}^n_i - \sum_p S^n_{pi} \frac{m_p}{m_i} v^n_p \tag{27}$$

where $\tilde{v}^n_i$ is the nodal value consistent with the values v^n_p. This error is estimated in Sect 5. below, equations (52) to (53).

The nodal acceleration is updated by using the stresses and deformation gradients at the current grid points and the body forces

$$a^n_i = \frac{-1}{m_i} \sum_p D^n_{pi} \sigma^n_p + b(X_i, t^n) \tag{28}$$

where the nodal mass is is defined by equation (7). The equation to update velocity at the nodes is then given by

$$v^{n+1}_i = v^n_i + dt a^n_i \tag{29}$$

The global error in this forward Euler step at time t^{n+1} is given by Ev^{n+1}_i and whose evolution may be approximated by

$$Ev^{n+1}_i = Ev^n_i + dtEa^n_i + \frac{dt^2}{2} \frac{d^2 v^n_i}{dt^2} \tag{30}$$

where the rightmost term is the local time error contribution and where Ea^n_i is the spatial error from using the approximation in equation (28), as defined by

$$Ea^n_i = a^n_{i,true} - a^n_i \tag{31}$$

and which will be estimated as defined in Sect. 5 below. The time derivative term is a simple approximation to the local time error which may be estimated by

$$\frac{dt^2}{2} \frac{d^2 v^n_i}{dt^2} \approx \frac{dt}{2} (a^n_i - a^{n-1}_i) \tag{32}$$

The equation for the update of the particle velocity is then

$$v^{n+1}_p = v^n_p + dt \sum_i S^n_{ip} a^n_i \tag{33}$$

The associated global error is defined by

$$Ev^{n+1}_p = v^{n+1}_{p,true} - v^{n+1}_p \tag{34}$$

whose evolution may be approximated by

$$Ev^{n+1}_p = Ev^n_p + dt \sum_i S^n_{ip} Ea^n_i + dtEa^n_{ip} + \frac{dt^2}{2} \frac{d^2 v^n_p}{dt^2} \tag{35}$$

where the rightmost term is the local time error contribution. In this case the error Ea^n_{ip} is the approximation error caused by the mapping coefficients S^n_{ip} and is estimated as described in Sect. 5 below using the same approach as in equations (61) and

(62) in that section. Furthermore the time derivative term is the local time error which may be estimated by

$$\frac{dt^2}{2}\frac{d^2v_p^{n+1}}{dt^2} \approx \frac{dt}{2}(a_p^n - a_p^{n-1}) \tag{36}$$

In the computational experiments the following estimate was used after considerable numerical testing as equation (35) gave computed error estimates that appeared to grow too quickly.

$$Ev_p^{n+1} = \sum_i S_{ip}^n Ev_i^{n+1} + dtEa_{ip}^n \tag{37}$$

From equations (26) (30) this may be written as

$$Ev_p^{n+1} = \sum_i S_{ip}^n \sum_p S_{pi}^n Ev_p^n + dtEa_{ip}^n + \sum_i S_{ip}^n \left(dtEa_i^n + Ev_{pi}^n + \frac{dt^2}{2}\frac{d^2v_i^n}{dt^2} \right) \tag{38}$$

where the rightmost term is the local time error contribution. The velocity gradients at particles are calculated using the formula

$$\frac{\partial v^{n+1}}{\partial x}(x_p) = \sum_i D_{ip}^n v_i^{n+1} \tag{39}$$

with an associated derivative approximation error as denoted by Ev_{xp}^{i+1}, defined by

$$Ev_{xp}^{i+1} = \frac{\partial v_{true}^{n+1}}{\partial x}(x_p) - \sum_i D_{ip}^n v_i^{n+1} \tag{40}$$

These velocity gradients are used to update the stress and deformation gradients at particles

$$F_p^{n+1} = F_p^n + dt\frac{\partial v^{n+1}}{\partial x}(x_p^n, t_n)F_p^n dt \tag{41}$$

The associated global error is defined by

$$EF_p^{n+1} = F_{p,true}^{n+1} - F_p^{n+1} \tag{42}$$

For the deformation gradient F_p^n the error evolution may be approximated by

$$EF_p^{n+1} = EF_p^n + dtF_p^n \left(Ev_{xp}^n + \sum_i D_{ip}^n Ev_i^{n+1} \right) - \frac{dt^2}{2}\frac{d^2F_p}{dt^2} \tag{43}$$

where the rightmost term is the local time error contribution and this second time derivative term corresponds to the local error from a semi-implicit Euler step with updated particle velocity derivatives at t^{n+1}. The semi-implicit method local time error has the opposite sign to the explicit. Stress is updated using the appropriate constitutive model and Young's Modulus, E,

$$\sigma_p^{n+1} = \sigma_p^n + dtE\frac{\partial v^{n+1}}{\partial x}(x_p^n) \tag{44}$$

The associated global error is defined by

$$E\sigma_p^{n+1} = \sigma_{p,true}^{n+1} - \sigma_p^{n+1} \tag{45}$$

In this case the stress global time and space error approximately evolves according to

$$E\sigma_p^{n+1} = E\sigma_p^{n+1} + dtE\left(Ev_{xp}^n + \sum_i D_{ip}^n Ev_i^{n+1}\right) - \frac{dt^2}{2}\frac{d^2\sigma_p^{n+1}}{dt^2} \tag{46}$$

in a similar way as for the deformation gradient, where again the rightmost term is the local time error contribution. In the computational experiments the term $\sum_i D_{ip}^n Ev_i^{n+1}$ was not used as it caused the stress error to grow too quickly, in part due to the presence of the constant E. The time derivative term is the local time error which may be estimated by .

$$\frac{dt^2}{2}\frac{d^2\sigma_p^{n+1}}{dt^2} \approx \frac{dtE}{2}\left(\frac{\partial v^{n+1}}{\partial x}(x_p^n) - \frac{\partial v^n}{\partial x}(x_p^n)\right) \tag{47}$$

However this term is not significant when the spatial error dominates and so was not included in the experiments. The equation for the particle position update is

$$x_p^{n+1} = x_p^n + dtv_p^{n+1} \tag{48}$$

The associated global error is defined by

$$Ex_p^{n+1} = x_{p,true}^{n+1} - x_p^{n+1} \tag{49}$$

For the particle update the error is given by

$$Ex_p^{n+1} = Ex_p^n + dt(Ev_p^n + Ev_{pi}) - \frac{dt^2}{2}\frac{d^2x_p^{n+1}}{dt^2} \tag{50}$$

where again the rightmost term is the local time error contribution. When implementing this it was found that the key error source term was Ev_{pi}^n which is estimated as in equations (61) and (62), and the term Ev_p^n was dropped, as its inclusion resulted in this error being over estimated. The time derivative term is the local time error which may be calculated by

$$\frac{dt^2}{2}\frac{d^2x_p^p}{dt^2} \approx \frac{dt^2}{2}a_p^n \tag{51}$$

Of course, it should be noted that in this analysis the error in evolving the error equations is neglected. As an example if the second time derivatives of the errors in a particle quantity are greater than the second time derivatives of the quantity itself then these errors must also somehow be estimated.

5 ESTIMATING THE SPATIAL ERROR TERMS

The above derivations illustrate how the spatial and temporal errors associated with MPM combine to give the overall error. Steffen et al. [16] observed these errors experimentally and arrived at the conclusion that for a stable time step with the methods they considered that temporal errors are dominated by spatial errors. This suggests that the starting point is to focus on the spatial errors terms in the derivation of the error in the MPM method in the previous section.

The error framework presented in the previous section makes it possible to derive estimates for the individual parts of MPM associated with the mapping matrix S_{ip} and the differentiation matrix D_{ip} (and their transposes) and to thus provide computable estimates for the error. The estimates derived in this section were first shown in a conference paper [6]. There are two parts to this process. The first part is to estimate the error in mapping from particles to the grid nodes. The second part is to estimate the error in mapping from the grid nodes back to particles. These are now considered in turn.

5.1 Particles to Nodes

In the case of the mapping defined by equation (25),(ignoring the contributions of the masses for the moment)

$$v_p^n = v_i^n + (x_p - X_i)\frac{\partial v}{\partial x}(X_i,t^n) + \frac{(x_p-X_i)^2}{2}\frac{\partial^2 v}{\partial x^2}(X_i,t^n) + \frac{(x_p-X_i)^3}{6}\frac{\partial^3 v}{\partial x^3}(X_i,t^n) + ... \tag{52}$$

Using the approach of [11] it is assumed that the mapping S_{ip} is linearity preserving i.e. $\sum_p S_{pi} = 1$ and $\sum_p S_{pi}x_p = X_i$. Approximating the true value of the error at the nodal velocity as in equation (27) by a local solution based upon computed solution values and multiplying (52) by S_{ip} and using linearity preservation gives

$$Ev_{pi}^n \approx -\frac{\partial^2 v}{\partial x^2}(X_i,t^n)\sum_p S_{pi}^n\frac{(x_p-X_i)^2}{2} - \frac{\partial^3 v}{\partial x^3}(X_i,t^n)\sum_p S_{pi}^n\frac{(x_p-X_i)^3}{6} \tag{53}$$

which requires the estimation of second and third derivatives at the nodes. Those derivatives are estimated with the present solution, rather than the true solution.

In the case of acceleration matters are more complicated. Consider the mapping defined by equation (28). The first step is to expand the stress at a particle about the node by using a simple Taylor expansion.

$$\sigma_p^n = \sigma_i^n + (x_p - X_i)\frac{\partial \sigma}{\partial x}(X_i,t^n) + \frac{(x_p-X_i)^2}{2}\frac{\partial^2 \sigma}{\partial x^2}(X_i,t^n) + \frac{(x_p-X_i)^3}{6}\frac{\partial^3 \sigma}{\partial x^3}(X_i,t^n) + ... \tag{54}$$

Substituting this in equation (28) gives after assuming that the coefficients D_{pi} exactly differentiate linear functions ($\sum_p D_{pi} = 0$ and $\sum_p D_{pi}x_p = 1$), see [11] who also provide a procedure for this. Define the stress derivative error as

$$E\sigma_{xi}^n = \frac{\partial \sigma}{\partial x}(X_i,t^n) + \sum_p D_{pi}^n\sigma_p^n \tag{55}$$

then using a Taylor series gives

$$E\sigma_{xi}^n = \sum_p D_{pi}^n \left[\frac{(x_p - X_i)^2}{2} \frac{\partial^2 \sigma}{\partial x^2}(X_i, t^n) + \frac{(x_p - X_i)^3}{6} \frac{\partial^3 \sigma}{\partial x^3}(X_i, t^n) + ... \right] \tag{56}$$

The stress derivatives at nodes are difficult to estimate and so nodal acceleration derivatives are used instead. This is complicated by the body forces contribution to acceleration so that if

$$\lambda_i^n = \frac{b(x_i, t^n)}{a_i^n} \tag{57}$$

then

$$\frac{\partial \sigma}{\partial x}(X_i, t^n) = (1 - \lambda_i^n) a_i^n \tag{58}$$

Assuming that the same approximation may be used for higher derivatives so that as $Ea_i = E\sigma_{xi}/(1-\lambda_i^n)$ then

$$Ea_i^n = \frac{-1}{\tilde{m}_i} \sum_p D_{pi}^n \left[\frac{(x_p - X_i)^2}{2} \frac{\partial a}{\partial x}(X_i, t^n) + \frac{(x_p - X_i)^3}{6} \frac{\partial^2 a}{\partial x^2}(X_i, t^n) + ... \right] \tag{59}$$

As the D_{pi} coefficients differentiate, it is possible to do this directly to get

$$Ea_i^n \approx \frac{-1}{\tilde{m}_i} \sum_p \left[(x_p - X_i) \frac{\partial a}{\partial x}(X_i, t^n) + \frac{(x_p - X_i)^2}{3} \frac{\partial^2 a}{\partial x^2}(X_i, t^n) + ... \right] \tag{60}$$

which is the estimate used in the experiments. The estimation of these spatial acceleration derivatives at the nodes is described below in Sect 5.3.

5.2 Nodes to Particles

In this case we have to consider the mapping from nodal values of accelerations to accelerations at particles, given by equation (33) and the mapping from nodal velocities to velocity derivatives at particles. It is assumed that the transposes of the mapping matrices S and D (as denoted by switching the subscript pi to ip) satisfy the same equations as above for preserving linearity in the mapping and for differentiating linear functions exactly, e.g. using the procedure of Gritton [11]. In both these cases we expand the nodal values about particles. Consider the mapping equation as used for velocity

$$Ev_p^n = v_{p,true}^n - \sum_i S_{ip}^n v_i^n \tag{61}$$

where the true particle velocity $v_{p,true}^n$ is again approximated using a Taylor series as in equation (52). Using the fact that that the sum $\sum_i S_{ip}^n = 1$ then

$$\begin{aligned} &Ev_p^n \approx \\ &-\sum_i S_{ip}^n \left(\frac{(x_p - X_I)^2}{2} \frac{\partial^2 v}{\partial x}(X_i, t^n) + \frac{(x_p - X_I)^3}{6} \frac{\partial^3 v}{\partial x^3}(X_i, t^n) \right) \end{aligned} \tag{62}$$

Where higher order than three derivatives are neglected in the Taylor expansion. A similar approach may be used for the approximation error to do with mapping the acceleration from nodes to particles Ea_{ip}^n

In the case of estimating the error in derivatives at particles the approach is similar to estimate the mapping error

$$Ev_{xp}^{n+1} = \frac{\partial v_{true}^{n+1}}{\partial x}(x_p) - \sum_i D_{ip}^n v_i^{n+1}, \tag{63}$$

Expanding about X_i gives

$$Ev_{xp}^{n+1} \approx -\sum_i D_{ip}^n \left(\frac{(X_i - x_p)^2}{2} \frac{\partial^2 v}{\partial x^2}(x_p, t^{n+1}) + \frac{(X_i - x_p)^3}{6} \frac{\partial^3 v}{\partial x^3}(x_p, t^{n+1}) \right) \tag{64}$$

This expression requires velocity derivatives at the particles. These values may be approximated by interpolating from the nodal derivatives so that, for instance,

$$\frac{\partial^2 v}{\partial x^2}(x_p, t^{n+1}) \approx \sum_j S_{ip}^n \frac{\partial^2 v}{\partial x^2}(X_j, t^{n+1}) \tag{65}$$

to get

$$Ev_{xp}^{n+1} \approx -\sum_i D_{ip}^n \left(\frac{(X_i - x_p)^2}{2} \sum_j S_{jp} \frac{\partial^2 v}{\partial x^2}(X_j, t^{n+1}) + \frac{(X_i - x_p)^3}{6} \sum_j S_{jp} \frac{\partial^3 v}{\partial x^3}(X_j, t^{n+1}) \right) \tag{66}$$

Alternatively the expansions

$$\frac{\partial^2 v}{\partial x^2}(x_p, t^{n+1}) = \frac{\partial^2 v}{\partial x^2}(X_i, t^{n+1}) + (x_p - X_i) \frac{\partial^3 v}{\partial x^3}(X_i, t^{n+1}) \tag{67}$$

and

$$\frac{\partial^3 v}{\partial x^3}(x_p, t^{n+1}) = \frac{\partial^3 v}{\partial x^2}(X_i, t^{n+1}) + (x_p - X_i) \frac{\partial^4 v}{\partial x^4}(X_i, t^{n+1}) \tag{68}$$

May be truncated and used in equation (64) to get

$$Ev_{xp}^{n} \approx -\sum_i D_{ip}^n \left(\frac{(X_i - x_p)^2}{2} \frac{\partial^2 v}{\partial x^2}(X_i, t^{n+1}) + \frac{2(X_i - x_p)^3}{3} \frac{\partial^3 v}{\partial x^3}(X_i, t^{n}) + \frac{(X_i - x_p)^4}{6} \frac{\partial^4 v}{\partial x^4}(X_i, t^{n+1}) \right) \tag{69}$$

This involves less computation and in the experiments the third and fourth-order terms were not included as in the computational experiments it was sufficient to just use the approximation

$$Ev_{xp}^{n+1} \approx -\sum_i D_{ip}^n \left(\frac{(X_i - x_p)^2}{2} \frac{\partial^2 v}{\partial x^2}(X_i, t^{n+1}) \right) \tag{70}$$

5.3 Estimating the Spatial Derivatives

The first spatial derivative of the stress $\sigma(X_i,t)$ or any other quantity at the nodes, such as acceleration or velocity is straightforwardly estimated using finite differences of nodal stress values.

$$\frac{\partial \sigma}{\partial x}(X_i,t) \approx \frac{\sigma_{i+1}-\sigma_{i-1}}{2h} \tag{71}$$

and the second derivative of the stress is straightforwardly estimated using finite differences of nodal stress values.

$$\frac{\partial^2 \sigma}{\partial x^2}(X_i,t) \approx \frac{\sigma_{i+1}-2\sigma_i+\sigma_{i-1}}{h^2} \tag{72}$$

and the third derivative similarly as

$$\frac{\partial^3 \sigma}{\partial x^3}(X_i,t) \approx \frac{\sigma_{i+2}-2\sigma_{i+1}+2\sigma_{i-1}+\sigma_{i-2}}{h^3} \tag{73}$$

With appropriate modifications at the boundaries that need to include the boundary conditions.

5.4 Order of Accuracy of MPM.

The error estimates derived in Sect. 5.1 and 5.2 above make it possible to define the order of accuracy of the above estimated error. There are two forms of error to consider. The first is the error due to the use of the mapping coefficients S_{pi} and the second is the error due to the use of the differentiation mapping coefficients D_{pi}. These are considered in turn. Consider the first term of the error defined by equation (62)

$$Ev_p^n = -\frac{\partial^2 v}{\partial x^2}(X_i,t^n)\sum_i S_{ip}^n \frac{(x_p-X_i)^2}{2} - \frac{\partial^3 v}{\partial x^3}(X_i,t^n)\sum_i S_{ip}^n \frac{(x_p-X_i)^3}{6} + h.o.t. \tag{74}$$

and let

$$c_{ip} = (x_p - X_i)/h \tag{75}$$

Then equation (74) may be written as

$$Ev_p^n = -\frac{h^2}{2}\,\frac{\partial^2 v}{\partial x}(X_i,t^n)\sum_i S_{ip}^n c_{ip}^2 - \frac{h^3}{6}\frac{\partial^3 v}{\partial x^3}(X_i,t^n)\sum_i S_{ip}^n c_{ip}^3 + h.o.t. \tag{76}$$

Or as

$$Ev_p^n = -\frac{h^2}{2}\,C_{Evp1}(t) - \frac{h^3}{6}\,C_{Evp2}(t) + h.o.t. \tag{77}$$

where

$$C_{Evp1}(t) = \frac{\partial^2 v}{\partial x^2}(X_i,t^n)\sum_i S_{ip}^n c_{ip}^2 \tag{78}$$

$$C_{Evp2}(t) = \frac{\partial^3 v}{\partial x^3}(X_i, t^n) \sum_i S_{ip}^n c_{ip}^3 \tag{79}$$

and as $-1 \leq c_{ip} \leq 1$ using equation (21)

$$\sum_i S_{ip}^n c_{ip}^2 \leq 1. \tag{80}$$

and

$$-1 \leq \sum_i S_{ip}^n c_{ip}^3 \leq 1. \tag{81}$$

For evenly spaced meshes and evenly spaced particles about nodes,$C_{Evp2}(t))$ may even be zero. For the second case consider the first term of the error defined by equation (56) written as

$$Ea_i^n = \frac{-1}{\tilde{m}_i} \sum_p \left[(x_p - X_i) \frac{\partial a}{\partial x}(X_i, t^n) + \frac{(x_p - X_i)^2}{3} \frac{\partial^2 a}{\partial x^2}(X_i, t^n) + ... \right] \tag{82}$$

and use equation (75 to write it as

$$Ea_i^n = \frac{-1}{\tilde{m}_i} \sum_p \left[h c_{ip} \frac{\partial a}{\partial x}(X_i, t^n) + h^2 \frac{c_{ip}^2}{3} \frac{\partial^2 a}{\partial x^2}(X_i, t^n) + ... \right] \tag{83}$$

This equation may be written as

$$Ea_i^n = \frac{-1}{\tilde{m}_i} \left[h C_{Eai1}(t) + \frac{h^2}{2} C_{Eai2}(t) \right] \tag{84}$$

where

$$C_{Eai1}(t) = \frac{\partial a}{\partial x}(X_i, t^n) \sum_p c_{ip} \tag{85}$$

and where $\sum_i c_{ip}$ may be "small" due to cancellations of positive and negative values about a node, and

$$C_{Eai2}(t) = \frac{\partial^2 a}{\partial x^2}(X_i, t^n) \sum_p c_{ip}^2 \tag{86}$$

In this case in contrast

$$\sum_p c_{ip}^2 \geq 0 \tag{87}$$

It is observed that this term is substantially larger than $\sum_i c_{ip}$ and that all the sums involving C_{ip} grow with the number of particles per call.

The expressions for the other spatial errors may be written in the same way. Part of the difficulty in assigning a clear order to these results is that some terms may be close to zero for some particle distributions. This means that the order will vary depending on the particle distributions to some extent. This error dependence on particle position was noted by Steffen et al. [14] who observed larger errors with randomized particle distributions.

6 COMPUTATIONAL EXPERIMENTS

While it is straightforward to write down the equations for evolution of the errors there are many subtleties and challenges in implementing such an approach. The first major challenge comes from the coupled nature of all the errors in the system of error equations defined by MPM. In particular this means that over or under estimation of a particular error propagates through the whole system. Thus with over estimation the error equations at increase at an unrealistic rate. After considerable experimentation the stated algorithm given above appears to a good compromise with regard to estimating the errors in all the components.

In order to test the estimates derived above the vibrating bar example that is often a standard MPM benchmark problem is used, e.g. [11]. The problem considered is a 1D bar problem of unit length, that is used in many MPM papers as a starting point for testing algorithms, see for example [15,11,3] and numerous others. The stress equation is

$$\sigma = P = E\frac{\partial u}{\partial X} = E(F-1), \tag{88}$$

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

$$\dot{\sigma} = E(\dot{F}), \tag{89}$$
$$= E(lF), \tag{90}$$

where l is the velocity gradient in the spatial description. The analytic solutions for displacement and velocity defined in the material description are:

$$u(X,t) = Asin(\frac{2\pi X}{l})sin(\frac{c\pi t}{l}), \tag{91}$$
$$v(X,t) = = \frac{Ac\pi}{l}sin(\frac{2\pi X}{l})cos(\frac{c\pi t}{l}), \tag{92}$$

where $c = \sqrt{E/\rho}$ for a density ρ,the length of the bar $l = 1$ and A is the maximum displacement. The constitutive model is defined in Equation 88 and the body force is,

$$b(X,t) = 3A(c\pi)^2 u(X,t). \tag{93}$$

The initial spatial discretization is on the spatial domain of $[0,1]$ as the length of the bar, $l = 1$. The periodic nature of the analytic solution means that both periodic boundary conditions and zero Dirichlet boundary conditions are both appropriate. Figure 1 shows one example mpm particle distributions and a uniform background mesh when the particles have moved from their uniform initial distributions, [16]

The initial conditions for the updated Lagrangian description of the particles are:

$$F = 1, \tag{94}$$
$$x_p = X_p^0, \tag{95}$$
$$V_p = V_p^0. \tag{96}$$

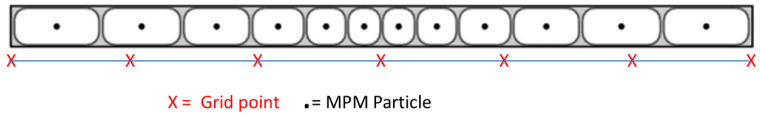

Fig. 1 1d Bar example particle distribution

Suppose that the error in the stress at a spatial point x is given by $e\sigma(x)$, with similar definitions for the other errors, then the following error vector definitions are needed.

$$Ex_p^n = \left[Ex_1^n, ..., Exp_{np}^n\right]^T \tag{97}$$

$$Ev_N^n = [Ev_1^n, ..., Ev_N^n]^T \tag{98}$$

$$Ev_p^n = \left[Evp_1^n, ..., Evp_{np}^n\right]^T \tag{99}$$

$$EA_N^n = [EA_1^n, ..., EA_N^n]^T \tag{100}$$

$$E\sigma_p = \left[E\sigma_1^n, ..., E\sigma_{np}^n\right]^T \tag{101}$$

These errors are those whose evolution is described by equations (50), (30), (38), (60) and (46) respectively.

In thgis section for this problem two main cases are considered to illustrate the evolution of the errors and their estimates in time. Case 1 uses a cell width is $h = 10^{-2}$ and two evenly spaced particles per cell. Case 2 uses a cell width is $h = 0.510^{-2}$ and four evenly spaced particles per cell. The material density is $\rho_0 = 1$, and Young's modulus is varied from $E = 4$ to $E = 1000$, maximum displacement is $A = 0.1$ and the timestep is varied. It should be noted that with the use of the above parameters there are many particle crossings, ranging from about 1000 with E=4 to about 8000 with E= 1000. Examples of the physical parameters in SI units for such test problems are given by Tran et al. [22]. In Section 7 in order to illustrate the order of accuracy of the different methods the cell width is varied from $h = 0.1$ to $h = 0.0015625$.

6.1 Original MPM vs GIMP

The first computational experiments re-iterate the relatively poor performance of the original MPM method vs GIMP, as described and referenced in Section 3.1. Included here is the linearity preserving MPM method, denoted as LPMPM. In Tables 1 and 2 the Maximum Error norms shown are the L_2 vector norms of the errors of particles averaged over the square root of the number of particles

$$||EX_p|| = \sqrt{\frac{1}{np}\sum_{i=1}^{np} Ex_i^2} \tag{102}$$

Table 1 Case 1 Maximum of error norms calculated at the end of every time step for particles displacement, velocity, acceleration and stress, h = 0.05

E	Method	dt	Err X_p	Err V_p	Err A_N	Err σ_p
1000	MPM	1.0e-3	3.2e-2	9.8	5.4e+3	3.7e+2
	LPMPM	1.0e-3	2.2e-2	1.5	9.4e+1	4.3e+2
	GIMP	1.0e-3	8.9e-3	7.8e-1	5.8e+1	1.7e+2
1000	MPM	1.0e-4	4.3e-2	7.2	8.5e+1	4.1e+2
	LPMPM	1.0e-4	2.4e-2	3.7	1.7e+1	3.0e+2
	GIMP	1.0e-4	2.1e-3	4.9e-1	7.6e+1	2.1e+1
1000	MPM	1.0e-5	7.2e-1	1.6e+2	8.9e+5	8.9e+2
	LPMPM	1.0e-5	5.4e-2	1.1e+1	2.9e+3	6.4e+2
	GIMP	1.0e-5	2.5e-3	6.3e-1	2.1e+2	4.1e+1
64	MPM	1.0e-3	2.7e-2	1.3	2.1e+2	1.3e+1
	LPMPM	1.0e-3	4.4e-3	1.7e-1	1.8e+1	4.4
	GIMP	1.0e-3	1.9e-3	8.9e-2	4.2	2.4
64	MPM	1.0e-4	2.6e-2	2.0e-2	4.2e+2	2.8e+1
	LPMPM	1.0e-4	7.3e-3	2.9e-1	4.2e+1	5.2
	GIMP	1.0e-4	2.3e-3	1.1	5.8	2.6
64	MPM	1.0e-5	3.8e-2	2.0	6.3e+4	3.0e+1
	LPMPM	1.0e-5	7.9e-3	3.8e-1	6.3	5.1
	GIMP	1.0e-5	2.4e-3	1.1e-1	7.2	2.7
4	MPM	1.0e-3	8.9e-3	2.3e-1	1.1e+1	6.1e-1
	LPMPM	1.0e-3	2.5e-3	3.1e-2	8.9e-2	2.6
	GIMP	1.0e-3	2.0e-3	2.1e-2	2.6e-2	1.5e-1
	MPM	1.0e-4	8.9e-3	3.1e-1	1.5e+1	8.5e-1
4	LPMPM	1.0e-4	2.7e-3	3.6e-2	1.7	2.0e-1
	GIMP	1.0e-4	2.0e-3	2.1e-2	2.8e-1	1.5e-1
4	MPM	1.0e-5	8.9e-3	3.1e-1	1.5e+1	8.4e-1
	LPMPM	1.0e-5	2.7e-3	3.7e-2	1.4	1.9e-1
	GIMP	1.0e-5	2.0e-3	2.1e-2	2.8e-1	1.5e-1

In the case of quantities at the nodes such as acceleration, the averaging is over the number of nodal points

$$||EA_N|| = \sqrt{\frac{1}{N}\sum_{i=0}^{N} EA_i^2} \tag{103}$$

These results show that the original MPM has errors that are often close to an order of magnitude large than the GIMP method. What is more unexpected is that the linearity preserving version of the original MPM method (LPMPM) often does surprisingly well by comparison. This suggests that linearity preservation is an important property for MPM methods to have.

In comparing Tables 1 and 2 the extra particles in Case 2 almost always lead to more accurate results. One exception is with MPM and LPMPM and the case E = 1000 which has many more grid crossings and for which the extra grid crossing errors appear to give worse results with Case 2 than Case 1.

6.2 Error Estimation Experiments

In evaluating error estimators it is common to use an error index that is the ratio of the estimated error norm divided ' by the actual error. As above the L_2 vector norm

Table 2 Case 2 Maximum of error norms calculated at the end of every time step for particles displacement, velocity, acceleration and stress, *=failure when particles attempt to leave the grid, h = 0.05.

E	Method	dt	Err X_p	Err V_p	Err A_N	Err σ_p
1000	MPM	1.0e-3	*	*	*	*
	LPMPM	1.0e-3	*	*	*	*
	GIMP	1.0e-3	*	*	*	*
1000	MPM	1.0e-4	4.3e-3	1.3e-1	3.3e+2	1.7e+1
	LPMPM	1.0e-4	1.4e-4	3.9e-2	1.9e+1	2.5
	GIMP	1.0e-4	6.6e-5	2.4e-2	4.6	1.4
1000	MPM	1.0e-5	7.8e-4	2.3e-1	4.9e+2	1.8e+1
	LPMPM	1.0e-5	8.7e-5	7.5e-2	3.2e+1	2.1
	GIMP	1.0e-5	7.3e-5	2.3e-2	6.2	1.4
64	MPM	1.0e-3	5.1e-4	3.0e-2	1.9e+1	8.9e-1
	LPMPM	1.0e-3	6.7e-5	4.2e-3	1.2	1.1e-1
	GIMP	1.0e-3	6.5e-5	3.8e-3	2.3e-1	8.9e-2
64	MPM	1.0e-4	4.3e-4	4.8e-2	2.7e+1	8.9e-1
	LPMPM	1.0e-4	7.9e-5	5.5e-3	1.7	1.3e-1
	GIMP	1.0e-4	6.6e-5	3.3e-5	3.8e-1	8.9e-2
64	MPM	1.0e-5	4.8e-4	8.0e-2	3.4e+1	1.1
	LPMPM	1.0e-5	8.9e-5	7.2e-3	2.3	1.3e-1
	GIMP	1.0e-5	6.6e-5	3.3e-3	4.2e-1	8.9e-2
4	MPM	1.0e-3	4.5e-4	1.4e-2	1.4	3.6e-2
	LPMPM	1.0e-3	6.7e-5	8.9e-4	8.5e-2	6.3e-3
	GIMP	1.0e-3	6.6e-5	6.7e-4	1.3e-2	5.7e-3
4	MPM	1.0e-4	4.1e-4	1.6e-2	2.0	4.7e-2
	LPMPM	1.0e-4	7.0e-5	1.2e-3	1.3e-1	6.6e-3
	GIMP	1.0e-4	6.6e-5	6.6e-4	1.3e-2	5.7e-3
4	MPM	1.0e-5	4.1e-4	1.8e-2	2.3	4.7e-2
	LPMPM	1.0e-5	7.0e-5	1.3e-3	1.5e-1	6.8e-3
	GIMP	1.0e-5	6.6e-5	6.6e-4	8.9e-3	5.7e-3

divided by the square root of the number of sample points is used. The error index of the estimated error norm is given by

$$ErrXIAv = \frac{\sum_{k=1}^{nsteps} ||Ex_p^k||_2}{\sum_{k=1}^{nsteps} ||E_{true}x_P^k||_2} \tag{104}$$

$$ErrVIAv = \frac{\sum_{k=1}^{nsteps} ||Ev_p^k||_2}{\sum_{k=1}^{nsteps} ||E_{true}v_P^k||_2} \tag{105}$$

$$ErrAIAv = \frac{\sum_{k=1}^{nsteps} ||EA_N^k||_2}{\sum_{k=1}^{nsteps} ||E_{true}A_N^k||_2} \tag{106}$$

$$Err\sigma IAv = \frac{\sum_{k=1}^{nsteps} ||E\sigma_p^k||_2}{\sum_{k=1}^{nsteps} ||E_{true}\sigma_P^k||_2} \tag{107}$$

Where the index k refers to the time at which particular error quantity that is being estimated and the subscript "true" refers to the actual error. These results show that the error estimators we have developed initially appear to do a good job of estimating the errors as is shown by the error indices. While the error indices described above show the average behavior of the error estimates it is good to also have more detailed

Table 3 Case 1 GIMP Average Error Indices for Particles Displacement, Velocity, Acceleration and Stress

E	dt	Err X IAv	Err V IAv	Err A IAv	Err σ IAv
1.0e+3	1.0e-3	0.621	0.886	0.882	0.75
	1.0e-4	0.604	1.380	1.151	1.89
	1.0e-5	0.403	1.280	1.302	1.63
64	1.0e-3	0.844	1.53	1.01	1.45
	1.0e-4	0.411	2.01	1.27	1.22
	1.0e-5	0.392	2.05	1.38	1.19
4	1.0e-3	0.470	2.09	1.139	1.20
	1.0e-4	0.418	2.32	1.266	1.18
	1.0e-5	0.416	2.42	1.283	1.18

Table 4 Case 1 LPMPM Average Error Indices for Particles Displacement, Velocity, Acceleration and Stress (*=failure)

E	dt	Err X IAv	Err V IAv	Err A IAv	Err σ IAv
1.0e+3	1.0e-3	*	*	*	*
	1.0e-4	0.44	1.380	1.151	1.89
	1.0e-5	0.38	1.1	1.5	2.8
64	1.0e-3	0.50	1.2	1.4	1.2
	1.0e-4	0.22	0.89	1.6	1.2
	1.0e-5	0.20	0.85	1.6	1.2
4	1.0e-3	0.40	1.7	1.6	1.0
	1.0e-4	0.34	1.7	1.6	0.99
	1.0e-5	0.33	1.9	1.6	0.99

Table 5 Case 2 GIMP Average Error Indices for Particles Displacement, Velocity, Acceleration and Stress (*=failure)

E	dt	Err X IAv	Err V IAv	Err A IAv	Err σ IAv
1.0e+3	*	*	*	*	*
	1.0e-4	1.16	1.17	1.363	2.89
	1.0e-5	0.52	1.85	1.404	2.55
64	1.0e-3	2.52	1.78	1.43	2.83
	1.0e-4	0.59	2.83	1.73	2.52
	1.0e-5	0.50	3.32	1.55	2.49
4	1.0e-3	0.80	1.78	1.44	2.83
	1.0e-4	0.52	2.83	1.37	2.52
	1.0e-5	0.51	3.32	1.55	2.49

information on the time evolution of the errors. Three cases are considered, E=4 E = 64 and E =1000. Figures 1 to 3 show the evolution of actual and estimated errors in particle displacement and velocity and nodal acceleration for Case 1. Figures 4 to 6 show the evolution of actual and estimated errors in particle displacement and velocity and nodal acceleration for Case 2.

These plots illustrate the challenges of estimating the errors in the complex system of MPM equations and also show that the error estimation approach provides order of magnitude estimates of the error.

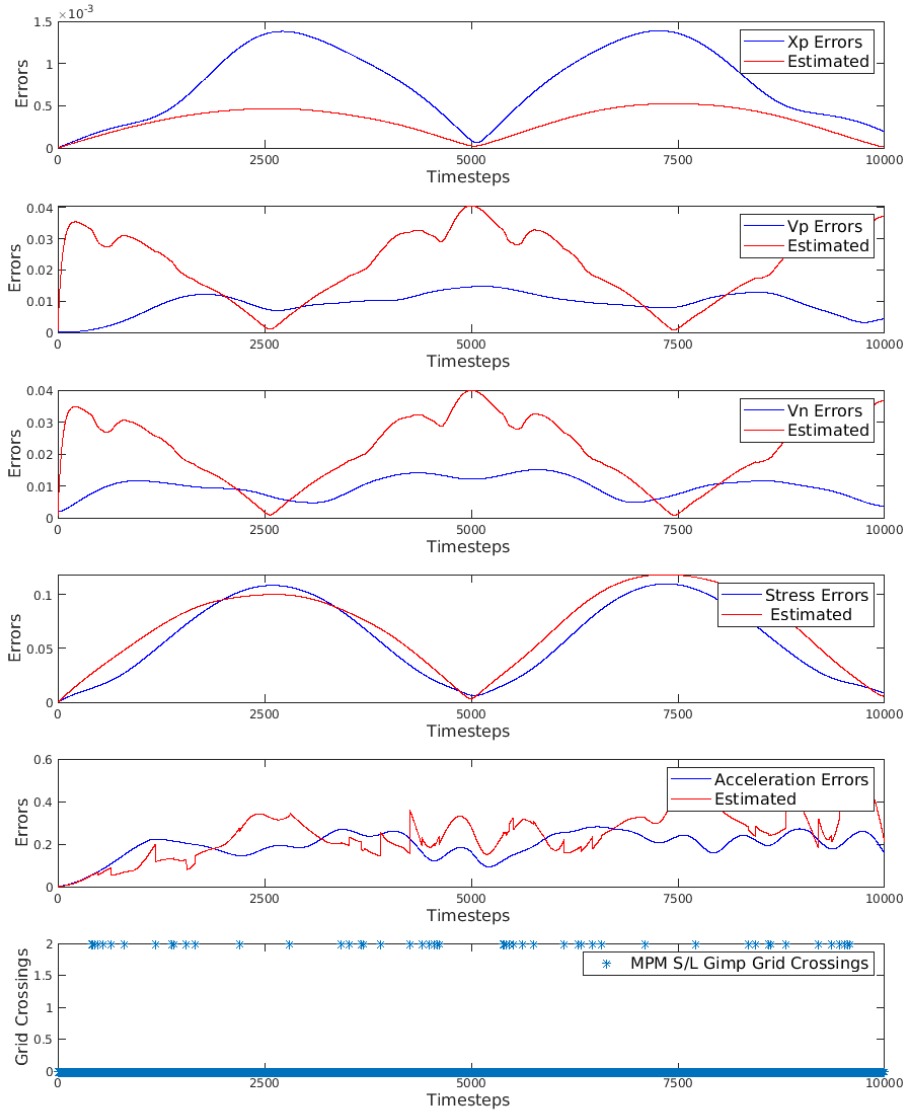

Fig. 2 1d Bar, Case 1:Plots of GIMP Error Norms in Displacement, Particle Velocities, Nodal Velocities, Particle Stresses, Nodal Accelerations and Numbers of Particle Grid Crossings for E=4 dt=1e-4

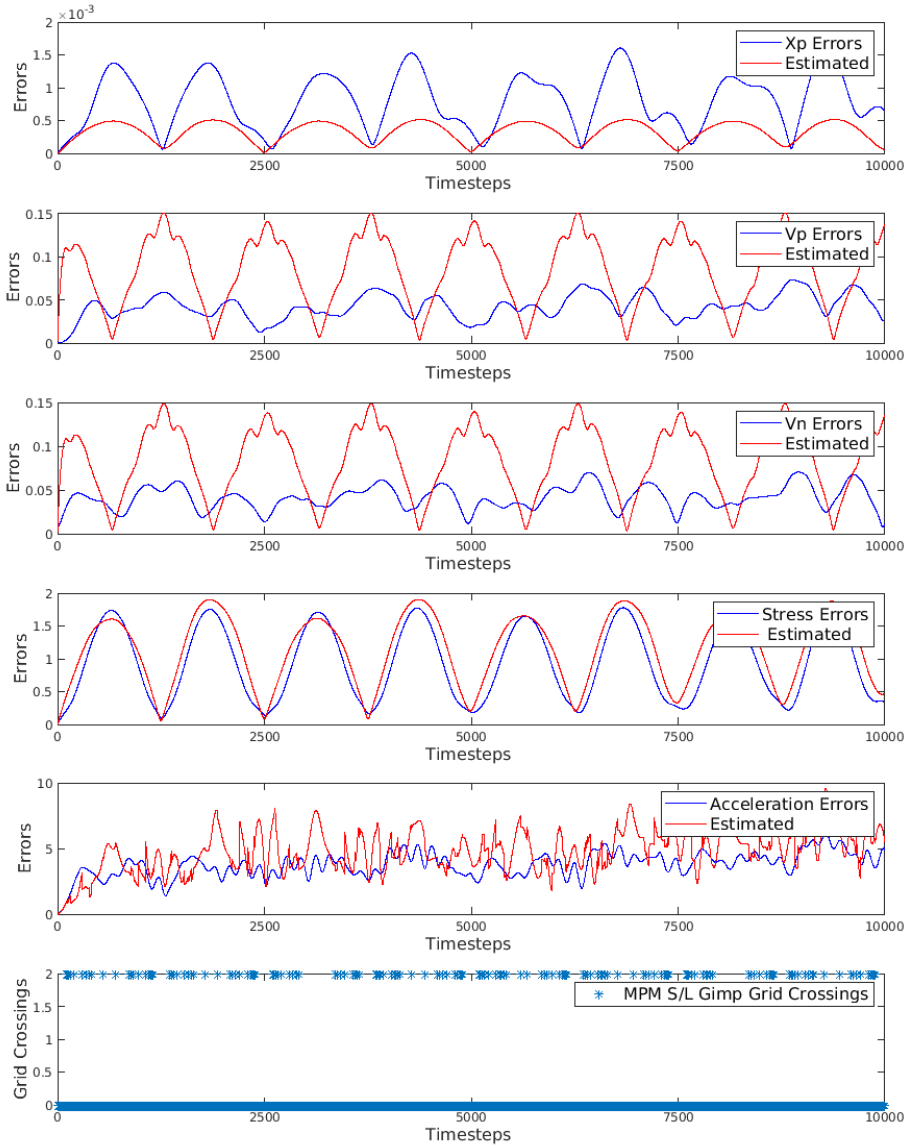

Fig. 3 1d Bar, Case 1:Plots of GIMP Error Norms in Displacement, Particle Velocities, Nodal Velocities, Particle Stresses, Nodal Accelerations and Numbers of Particle Grid Crossings for E=64 dt=1e-4

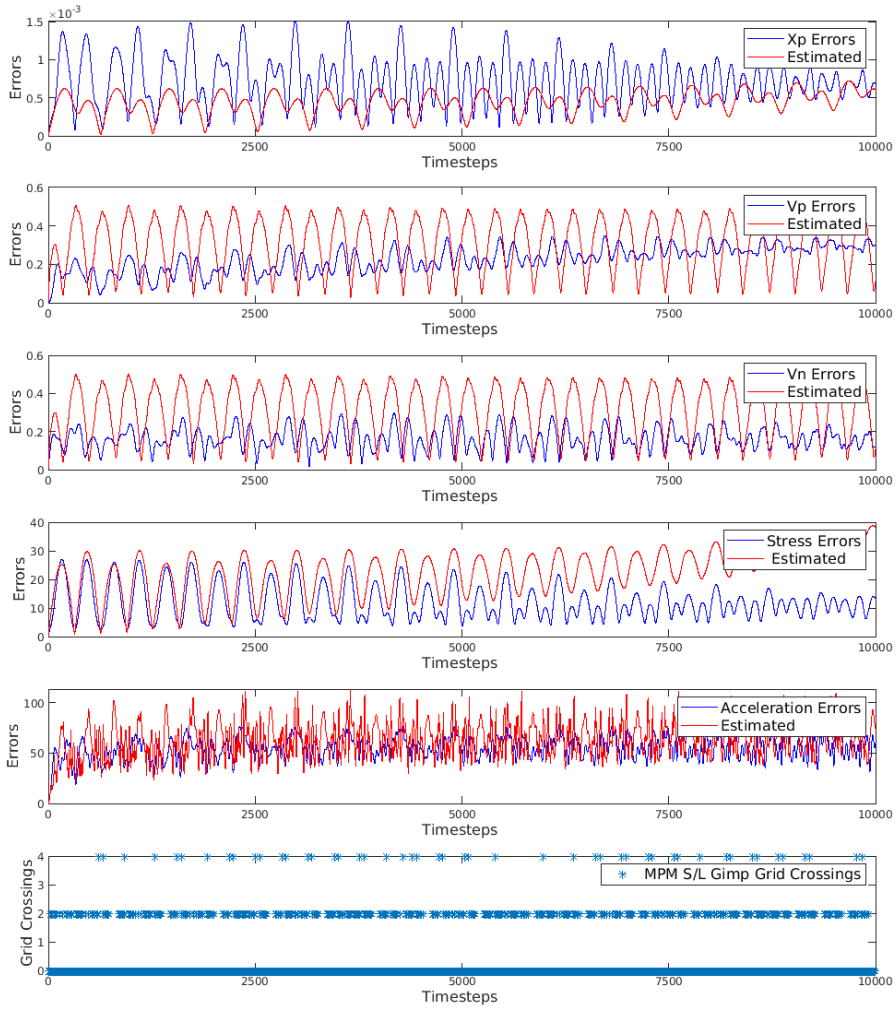

Fig. 4 1d Bar, Case 1:Plots of GIMP Error Norms in Displacement, Particle Velocities, Nodal Velocities, Particle Stresses, Nodal Accelerations and Numbers of Particle Grid Crossings for E=1000 dt=1e-4

7 DISCUSSION OF RESULTS

The complexity of estimating the errors in all the coupled components of MPM presents a more challenging problem than might be expected even for a relatively simple problem such as the one considered here. It is perhaps worth noting that there

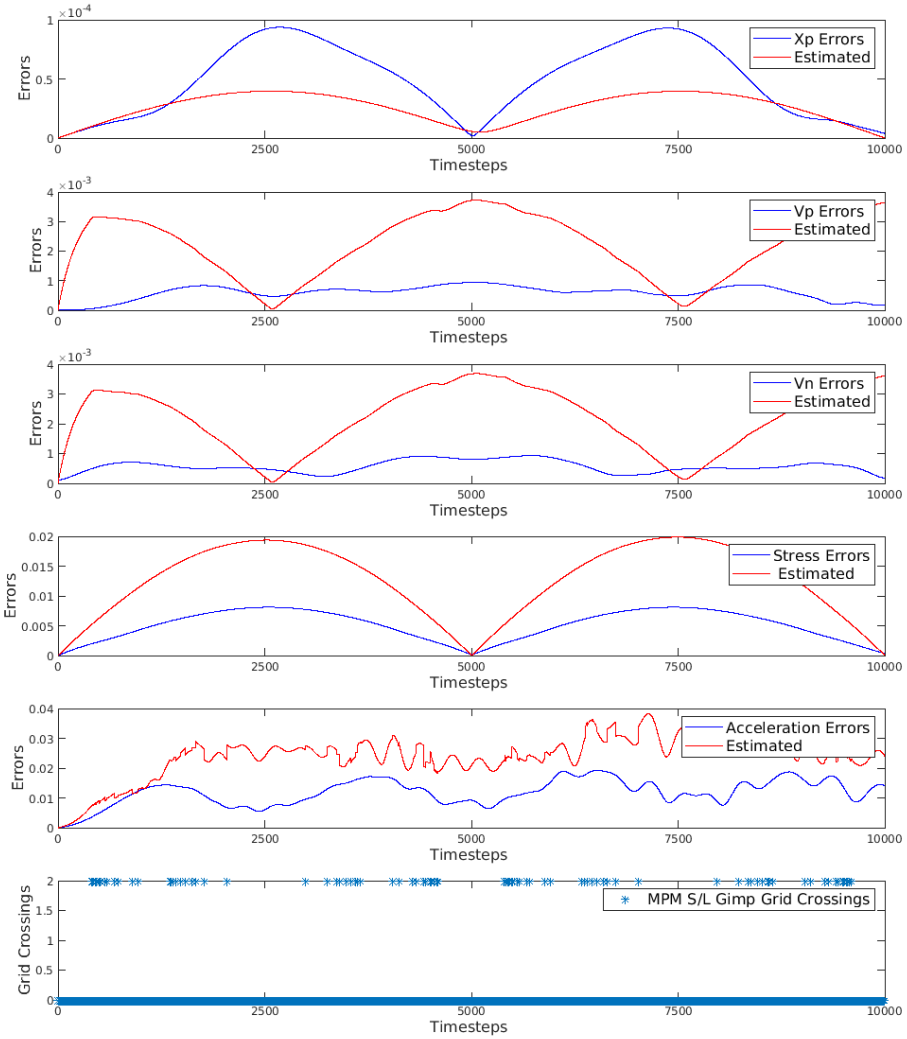

Fig. 5 1d Bar, Case 2:Plots of GIMP Error Norms in Displacement, Particle Velocities, Nodal Velocities, Particle Stresses, Nodal Accelerations and Numbers of Particle Grid Crossings for E=4 dt=1e-4

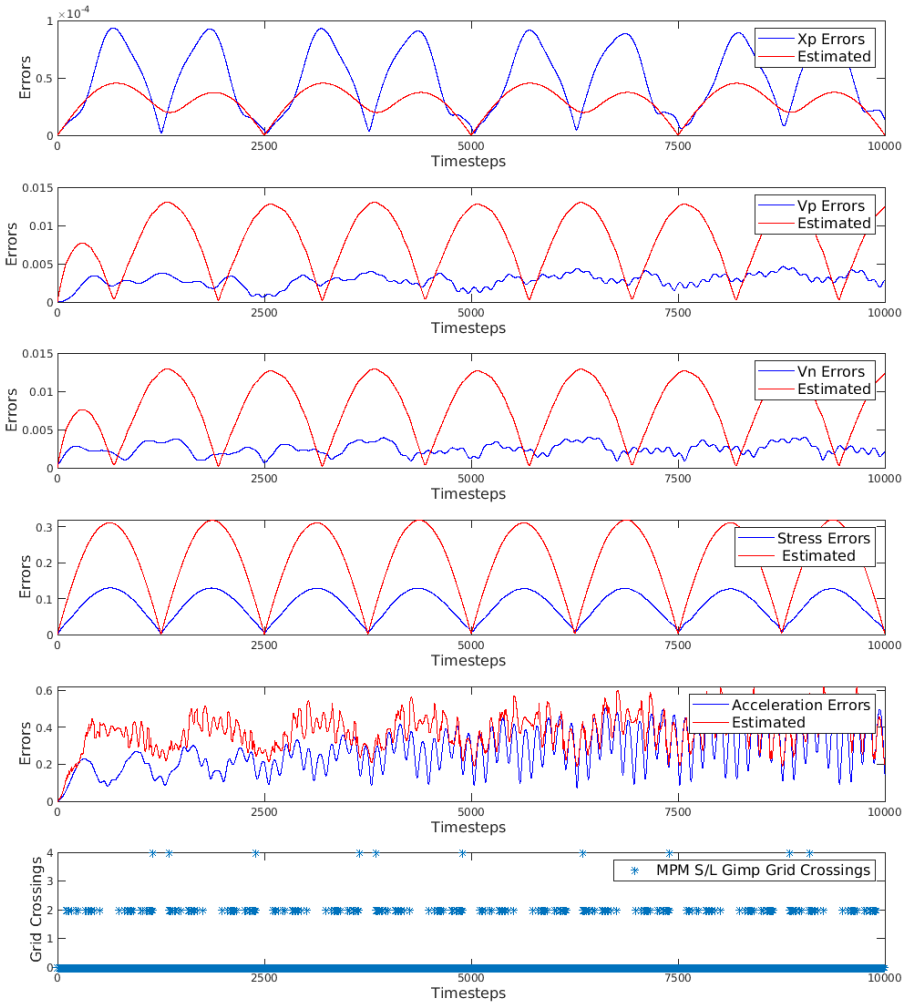

Fig. 6 1d Bar, Case 2:Plots of GIMP Error Norms in Displacement, Particle Velocities, Nodal Velocities, Particle Stresses, Nodal Accelerations and Numbers of Particle Grid Crossings for E=64 dt=1e-4

are relatively few attempts to follow error growth in time for complex systems such as those that arise in MPM and none that the author is aware of for particle methods.

Perhaps the most striking aspect of this is that while the actual errors are bounded in time for the smaller values of the parameter E, it is easy to produce error estimates that in the course of an integration that has thousands of time steps over or underes-

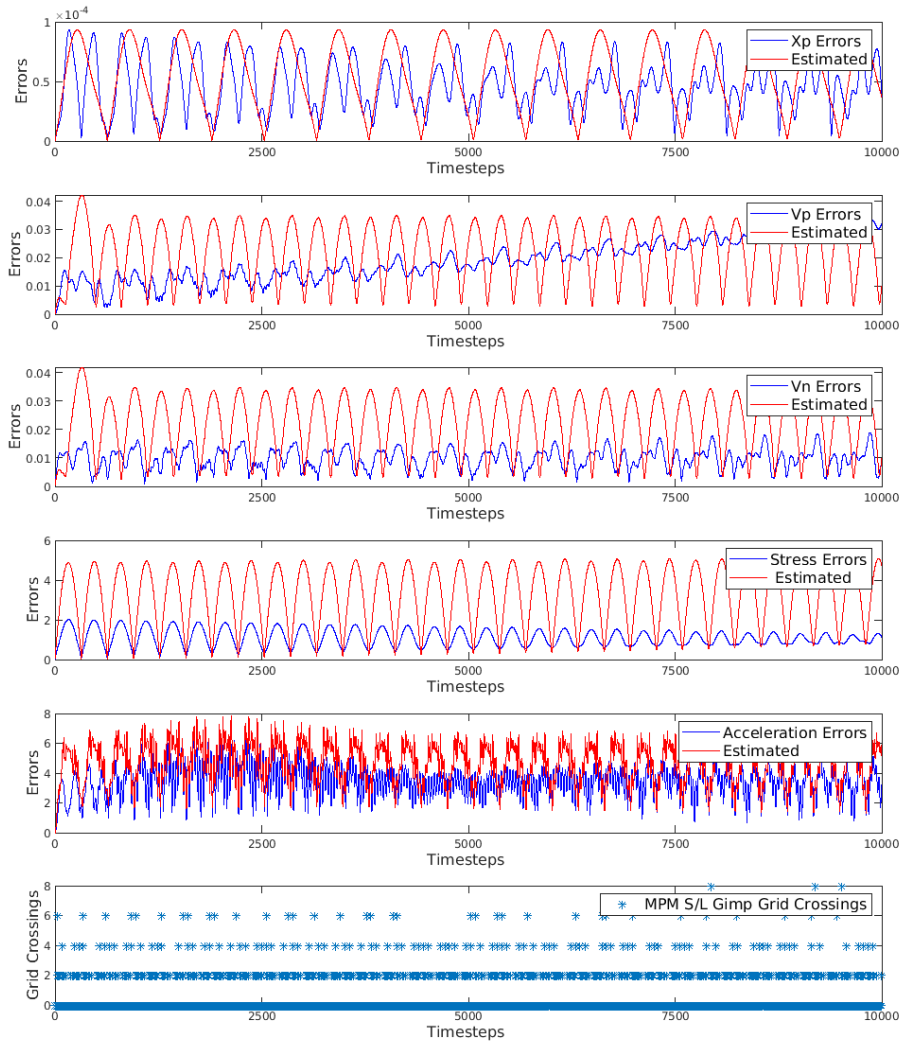

Fig. 7 1d Bar, Case 2:Plots of GIMP Error Norms in Displacement, Particle Velocities, Nodal Velocities, Particle Stresses, Nodal Accelerations and Numbers of Particle Grid Crossings for E=1000 dt=1e-4

Table 6 Case 2 LPMPM Average Error Indices for Particles Displacement, Velocity, Acceleration and Stress (*=failure)

E	dt	Err X IAv	Err V IAv	Err A IAv	Err σ IAv
1.0e+3	*	*	*	*	*
	1.0e-4	0.60	0.88	2.7	1.7
	1.0e-5	0.40	1.2	3.2	1.8
64	1.0e-3	2.0	1.8	3.0	2.1
	1.0e-4	0.46	1.6	3.2	1.9
	1.0e-5	0.39	1.6	3.7	1.8
4	1.0e-3	0.75	1.7	3.6	2.3
	1.0e-4	0.46	2.5	4.1	2.2
	1.0e-5	0.45	2.5	4.1	2.1

timate the error. As the error estimators involves approximations at every level and coupled this is very easy to do and so required the selection of terms that are of appropriate magnitude. This aspect of the error estimates is perhaps the one that requires further research.

Part of the explanation is that the errors are bounded in this case in the same way as the energy is bounded [5]. This means that the discrete exact solution and the computed solution both satisfy an approximate energy conservation law as shown in [5]. In equation (88) in [5] it is shown that the change in energy per step is $O(dt^3)$ as given by

$$\Delta \mathscr{E}\mathscr{E}_{err} = \frac{dt}{2}\sum_q (\frac{(v_q^{n+1} - v_q^n)}{2}) m_q \left(a_q^{n+1} - a_q^n\right) \tag{108}$$

This equation does not take into account errors in the solution. Assuming that the true solution mapped onto a grid satisfies a similar equation (which only requires differentiability of that solution we get the equation

$$\Delta \mathscr{E}\mathscr{E}_{err}^{true} = \frac{dt}{2}\sum_q (\frac{(v_{q,true}^{n+1} - v_{q,true}^n)}{2}) m_q \left(a_{q,true}^{n+1} - a_{a,true}^n\right) \tag{109}$$

As the errors at particles and nodal values are the differences between the true and computed values are the errors, then by subtraction the errors in velocity ev_q^n and acceleration ea_q^n satisfy an approximate conservation like equation given by

$$\Delta \mathscr{E}\mathscr{E}_{err}^{true} - \Delta \mathscr{E}\mathscr{E}_{err} = \frac{dt}{2}\sum_q (\frac{(ev_q^{n+1} - ev_q^n)}{2}) m_q \left(a_q^{n+1} - a_q^n\right) \tag{110}$$

$$+(\frac{(v_q^{n+1} - v_q^n)}{2}) m_q \left(ea_q^{n+1} - ea_q^n\right) + (\frac{(ev_q^{n+1} - ev_q^n)}{2}) m_q \left(ea_q^{n+1} - ea_q^n\right) \tag{111}$$

What may happen is that any approach that independently follows the evolution of the error (as in this paper) may not satisfy this condition and this may result in the error estimate blowing up when the true error does not. This issue of error estimation in the presence of energy conservation is one that needs further investigation.

One unexpected aspect of the results obtained is that the introduction of linearity preservation, so as to make error estimation possible to the original MPM to give

Table 7 Convergence Properties of GIMP LPMPM and MPM, E = 64 dt =1.0e-4, 2 particles per cell initially, Ratios are those of an error divided by the entry below it.

Method	h	Err X_p	Err V_p	Err A_N	Ratio Ex_p	Ratio Ev_p	Ratio Ea_N
GIMP	1/10	2.8e-2	2.22	1.3e+2	46	111	32
	1/20	6.4e-4	2.6e-2	4.8	13	5	3.7
	1/40	4.7e-5	5.0e-3	1.34	12	10.8	44
	1/80	3.7e-6	4.6e-4	2.8e-2	5.2	9.6	8.2
	1/160	7.0e-7	4.8e-5	3.4e-3	4.11	9.6	5.5
	1/320	1.7e-7	5.3e-6	6.2e-4	3.9	8.4	6.8
	1/640	4.8e-8	6.7e-7	9.1e-5			
LPMPM	1/10	3.3e-2	1.93	1.3e+2	15	10	8.6
	1/20	2.2e-3	1.96e-1	2.5e+1	11.6	14	9.8
	1/40	1.9e-4	1.3e-2	2.55	3.6	2.4	2.8
	1/80	5.3e-5	5.4e-3	9.3e-1	4	7.7	4.5
	1/160	1.3e-5	6.9e-4	2.0e-1	3.1	3	2.4
	1/360	4.1e-6	2.3e-4	8.3e-2	2.6	2.4	1.7
	1/640	1.4e-6	9.5e-5	4.9e-2			
MPM	1/10	1.3e-2	1.55	7.0e+1	1.4	1.2	0.3
	1/20	9.5e-3	1.30	2.3e+2	1.4	2.6	1.8
	1/40	6.8e-3	5.0e-1	1.3e+2	4	6.4	3.6
	1/80	1.7e-3	7.8e-2	3.6e+1	2.4	2.9	2.8
	1/160	7.1e-4	2.7e-2	1.3e+1	1.6	2.7	2.5
	1/320	4.5e-4	1.0e-2	5.1	2.4	1.9	1.1
	1/640	1.9e-4	5.2e-3	4.7			

LPMPM improves the performance of the method greatly, but not so much as to be an improvement over GIMP. The error estimating approach used here also appears to perform equally well for LPMPM and the GIMP method with linearity preservation. One surprising result in Table 7 is that on the very coarsest mesh the original MPM method performs best, although in every other case the GIMP method is superior.

8 Observed Order of Accuracy

The form of the error derived in Sect 5.3 now makes it possible to derive theoretical error estimates for the linearity preserving MPM methods considered here. It is clear from Sect.5 that the relationship between the error and the width of interval h is a complex one. In order to show the observed error a series of runs were done with all three methods considered here and with h being successively halved. The errors in particle displacement, velocity and nodal acceleration are shown in Table 7. The ratios in the Table show how much the error is reduced by when the mesh is halved. Thus ratios of 2,4, and 8 correspond to first, second and third order errors.

The results in Table 7 show the poor first performance of the original MPM algorithm in that moving to a mesh that is twice as fine only reduces the error by two. The linearity preserving version of it is closer to second order as on the finer mesh the error goes down by four or more sometimes. The linearity preserving version of GIMP appears to give certainly second and almost third order accuracy.

It is natural to ask if the error estimators still hold on the finer meshes. Figure 8 shows the evolution of the error and of the error estimates with $h = 1/320$ and two

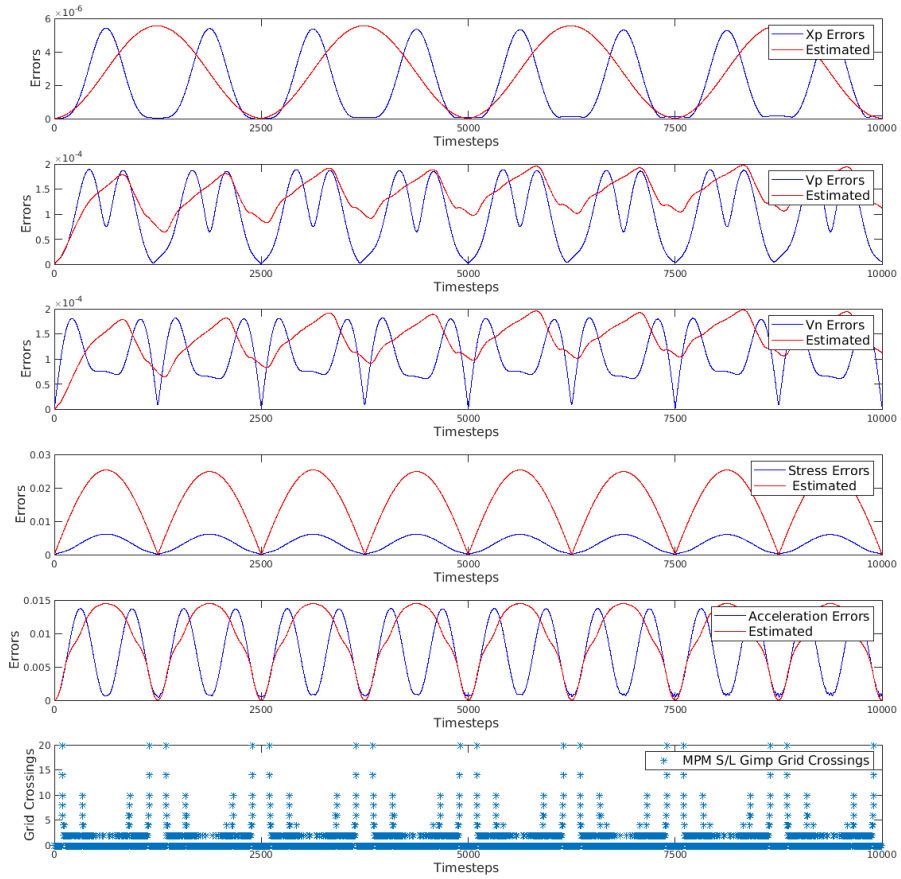

Fig. 8 1d Bar, Plots of GIMP Error Norms in Displacement, Particle Velocities, Nodal Velocities, Particle Stresses, Nodal Accelerations and Numbers of Particle Grid Crossings for E=64, h = 1/320, dt=1e-4

particles per interval. The error estimates derived here estimate all the errors very closely, apart from an overestimate of the stress error.

These results in Table 8 show the error indices for the GIMP method and the LPMPM method over the range of integration as defined in equations (104) to (107). Where results are not shown for a grid size the run failed because of particles leaving the domain. Overall the computed error estimation indices indicate that the error estimate is the right order of magnitude. Based on previous we can really only expect error indices between 0.25 and 5 for a complex coupled problem with a large number of time steps. The error estimates presented here meet these expectations. Overall the acceleration errors are estimated well for all grid sizes. The Stress error is over

Table 8 Properties of Error Estimator for GIMP and LPMPM, E = 4,64,1000 dt =1.0e-4, 2 particles per cell initially, Error Indices quoted are the averages defined in equations (104) to (107).

Method	h	ErrX_pIAvg	Err V_pIAvg	Err A_NIAvg	Errσ_pIAvg
GIMP	1/10	0.27	0.01	2.2	0.5
E=4	1/20	0.42	0.04	1.3	1.2
	1/40	0.37	0.09	0.9	1.7
	1/80	0.28	0.22	1.0	2.4
	1/160	0.27	0.30	1.2	3.4
	1/320	0.38	0.5	1.3	4.9
	1/640	0.7	1.23	1.4	6.9
LPMPM	1/10	0.2	0.01	2.0	0.5
E=4	1/20	0.4	0.04	1.7	0.9
	1/40	0.3	0.1	1.7	1.5
	1/80	0.25	0.29	1.9	2 1
	1/160	0.24	0.48	1.8	3.0
	1/360	0.34	0.67	1.6	4.2
	1/640	0.64	1.3	1.5	6.0
GIMP	1/10	0.4	0.03	1.8	1.1
E=64	1/20	0.4	0.14	1.3	1.2
	1/40	0.4	0.28	0.86	1.7
	1/80	0.46	0.46	0.97	2.5
	1/160	0.73	0.62	1.2	3.5
	1/320	1.4	1.4	1.3	5.0
	1/640	2.8	2.9	1.4	7.1
LPMPM	1/10	0.3	0.03	1.75	1.1
E=64	1/20	0.2	0.1	1.6	1.2
	1/40	0.2	0.3	1.7	1.2
	1/80	0.3	0.5	1.6	4 5
	1/160	0.6	0.6	1.5	2.1
	1/360	1.1	0.9	1.2	2.7
	1/640	2.2	1.6	1.0	3.4
GIMP	1/20	0.6	0.30	1.1	1.9
E=	1/40	0.85	0.40	0.7	2.7
1000	1/80	1.6	0.70	1.0	3.2
	1/160	3.0	1.3	1.2	4.4
LPMPM	1/20	0.4	0.20	1.5	3.8
E=	1/40	0.2	0.4	1.5	1.0
1000	1/80	0.4	0.5	1.4	0.8
	1/160	0.7	0.4	0.7	0.8

estimated and the velocity and displacement errors are underestimated for coarser meshes. The results suggest that perhaps more work is needed to estimate the errors in velocities and displacements. Table 7 for E =64 shows that the error drops very fast for the coarser meshes. It is perhaps not surprising then the errors from $h = 1/40$ downwards appear to be closer to what is expected in terms of asymptotic behavior, the estimates are more reasonable there too, this is particularly true from $h = 1/80$ downwards where we see second and third order accuracy from GIMP and first and second order accuracy from LPMPM with mostly good error indices. The error estimators do not apply for the original MPM method as it is not linearity preserving. In some cases when the mesh size is reduced by a factor of two the relative error index in veocity and displacement errors grows by 2 also, thus suggesting that further investigation of the estimation of these errors is perhaps needed.

9 CONCLUSIONS AND FUTURE WORK

The result of the approach presented here is that a decomposition of the different errors in MPM has been used to derive estimates for the mappings inherent in MPM between particles and nodes and vice versa. These simple estimates have been shown to work well in the simple demonstration case used here. There are differences in that reliability of the error estimates for the different solution components and one topic for future study is why the error estimate may grow much more quickly than the actual error. A possible explanation for this in terms of energy conservation is provided. Further work could also be done to revise the estimates of individual parts of some of the errors, particularly in velocity and displacement. The use of linearity preservation to derive error estimates also appears to improve the accuracy of MPM and GIMP, but GIMP is still more accurate.

An obvious extension of this work work involves applying this approach in a full MPM simulation and for two and three space dimensions. As the estimates derived here involve approximating derivatives of solution values on a regular mesh, this would seem to be entirely possible and is ongoing work using the method of manufactured solutions problems in [25]. In order to implement linearity preservation in multiple dimensions the approach of Vidal et al. [24] may be used, or any of the approaches mentioned in [11]. While the error framework and also linearity preservation can be used in multiple dimensions significant challenges in practical engineering problems such as complex constitutive laws and addressing fracture will likely prove challenging for some time to come.

ACKNOWLEDGEMENTS

The original GIMP code written by Chris Gritton for [11] was used as the starting point for the code developed here. This research was partially sponsored by the Army Research Laboratory under Cooperative Agreement Number W911NF-12-2-0023. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the US Government.

Finally the author would like to thank the reviewers for their helpful and insightful comments that have helped to improve this paper

Conflict of Interest

The author states that there is no conflict of interest.

References

1. S. Bardenhagen, Energy conservation error in the material point method for solid mechanics, *Journal of Computational Physics*, 180, 2002, 383-403.
2. Bardenhagen S. and Kober E., The generalized interpolation material point method, *Computer Modeling in Engineering and Science*, 5 (2004), 477-495.

3. Berzins M. Nonlinear Stability and time step selection in the material point method. *Computational Particle Mechanics* January 2018.
4. Berzins M. Symplectic Time Integration Methods for the Material Point Method, Experiments, Analysis and Order Reduction, *WCCM-ECCOMAS2020 virtual Conference* proceedings January, 2021.
5. Berzins M. Energy Conservation and Accuracy of Some MPM Methods. *Computational Particle Mechanices* February 04 2022.
6. Berzins M. Time Stepping with Space and Time Errors and Stability of the Material Point Method. *VII International Conference on Particle-Based Methods PARTICLES 2021* (Eds) P. Wriggers, M. Bischoff, E. O nate, M. Bischoff, A. Duster and T. Zohdi proceedings (to appear) 2022. `http://www.sci.utah.edu/publications/Ber2021c/Berzins_particles2021.pdf`.
7. Berzins M. Global Error Estimation in the Method of Lines for Parabolic Equations, *SIAM Journal on Scientific Computing*, Vol. 9, pp. 687–703. 1988.
8. Berzins, M., "Temporal Error Control for Convection-Dominated Equations in Two Space Dimensions", *SIAM Journal on Scientific Computing*, 16,3, 558-580,1995.
9. Cremonesi, M., Franci, A., Idelsohn, S. et al. A State of the Art Review of the Particle Finite Element Method (PFEM). *Arch Computat. Methods Eng.* 27, 17091735 (2020). https://doi.org/10.1007/s11831-020-09468-4
10. Grigoryev, Y. N., Vshivkov, V. A., and Fedoruk, M. P. "Numerical Particle-in-cell methods: Theory and applications." **2012** De Gruyter https://doi.org/10.1515/9783110916706
11. Gritton C. and Berzins M. Improving accuracy in the MPM method using a null space filter, *Computational Particle Mechanics*, 4, pp. 131142 2017.
12. Raviart, P. A. (1985). "An analysis of particle methods". In F. Brezzi (Ed.), i *Numerical Methods in Fluid Dynamics. Lecture Notes in Mathematics, vol 1127*. 1985 243324. Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0074532
13. Wojciech T. Solowski , Martin Berzins, William M. Coombs, James E. Guilkey, Matthias Moller, Quoc Anh Tran, Tito Adibaskoro, Seyedmohammadjavad Seyedan, Roel Tielen, and Kenichi Soga Material point method: Overview and challenges ahead Chapter 2 of *Advances in Applied* Vol. 54. Eds. Stephane P.A. Bordas and Daniel S, Balint. November 23 2021, ISBN: 9780323885195.
14. Steffen M., Kirby R.M., Berzins M. Analysis and Reduction of Quadrature Errors in the Material Point Method (MPM), *Int. J. for Numer. Meths. in Engnng*,**76**, 6, 922–948, 2008.
15. Steffen M., Wallstedt P.C., Guilkey J.E. , Kirby R.M. and Berzins M., Examination and analysis of implementation choices within the Material Point Method (MPM), *Computer Modeling in Engineering & Sciences*, 2008, **31**, 2, 107-127.
16. Steffen M., Kirby R.M., Berzins M. Decoupling and Balancing of Space and Time Errors in the Material Point Method (MPM), *Int. J. for Numer. Meths. in Engnng*, **82**, No. 10, pp. 1207–1243. 2010.
17. Sulsky D., Chen Z. and Schreyer H.L. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering* 118 (1994):179-196.
18. Sulsky D., Zhou S.-J. and Schreyer H.L. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* 87 (1995):236-252.
19. Honglai Tan, John A. Nairn "Hierarchical, adaptive, material point method for dynamic energy release rate calculations" *Computer Methods in Applied Mechanics and Engineering* **191**, 2123-2137.
20. Thielmann, M., May, D.A. and Kaus, B.J.P. Discretization Errors in the Hybrid Finite Element Particle-in-cell Method. i *Pure Appl. Geophys.* **171**, 21652184 (2014).
21. Tran, L. T., Kim, J., and Berzins, M. (2010). "Solving time-dependent PDEs using the material point method, a case study from gas dynamics." *International Journal for Numerical Methods in Fluids*. 23 March 2009 https://doi.org/10.1002/fld.2031
22. Tran Q. A., Solowski W., Berzins M., and Guilkey J. (2019), "A convected particle least square interpolation material point method", *International Journal for Numerical Methods in Engineering*, 2019, Wiley, October 2019,
23. de Vaucorbeil A., Nguyen V.P., Sinaie S., Wu J.Y, Chapter Two - Material point method after 25 years: Theory, implementation, and applications, Editor(s): Stphane P.A. Bordas, Daniel S. Balint, *Advances in Applied Mechanics,* Elsevier, 2020 **53**, 2020, 185-398.
24. Vidsal Y., Bonet J., and Huerta A. "Stabilized updtaed Lagrangian corrected SPG+H for explicit dynamics methods." *International Journal for Numerical Methods in Engineering* . **69**, 13, 2687-2710, 2007.
25. Wallstedt P.C. and Guilkey J.E., An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *Journal of Computational Physics* 2008, **227**, 22, 9628-9642.

26. Tong Zhang, ShiShun Li, "A posteriori error estimates of finite element method for the time-dependent NavierStokes equations", *Applied Mathematics and Computation*, Volume 315, 2017, Pages 13-26, ISSN 0096-3003,
27. Xiong Zhang and Zhen Chen and Yan Liu "The Material Point Method." **2017** Academic Press.