VIII International Conference on Particle-Based Methods
PARTICLES 2023
U. Perego, M. Cremonesi, A. Franci (Eds)

COMPUTATIONAL ERROR ESTIMATION FOR THE MATERIAL POINT METHOD IN 1D AND 2D.

Martin Berzins *

* SCI Institute and School of Computing
University of Utah
72. S.Central Campus Drive, Salt Lake City USA
e-mail: mb@sci.utah.edu, web page: http://sci.utah.edu/people/mb.html

Abstract: The Material Point Method (MPM) is widely used for challenging applications in engineering, and animation but lags behind some other methods in terms of error analysis and computable error estimates. The complexity and nonlinearity of the equations solved by the method and its reliance both on a mesh and on moving particles makes error estimation challenging. Some preliminary error analysis of a simple MPM method has shown the global error to be first order in space and time for a widely-used variant of the Material Point Method. The overall time dependent nature of MPM also complicates matters as both space and time errors and their evolution must be considered thus leading to the use of explicit error transport equations. The preliminary use of an error estimator based on this transport approach has yielded promising results in the 1D case. One other source of error in MPM is the grid-crossing error that can be problematic for large deformations leading to large errors that are identified by the error estimator used. The extension of the error estimation approach to two space higher dimensions is considered and together with additional algorithmic and theoretical results, shown to give promising results in preliminary computational experiments.

Keywords. Material Point Method Error Estimation

1 INTRODUCTION

The Material Point Method (MPM) continues to be an evolving method that has proved to be capable of solving many challenging problems in computational mechanics [10, 16, 9]. While for many methods in computational mechanics computational error estimates are available, they are very much lacking for the Material Point Method. In part this is due to the complex system of time-dependent non-linear equations and the need to track the error as it evolves in time, [5].

In the case of the Material Point Method, the first step is analyze the sources of different MPM errors see [5]. Estimating these errors may involve a finite element approach [1] or may use a linearity preservation formulation [7] to estimate the errors as in [5]. One problem is that grid crossing may affect the error decomposition derived by [5], particularly in the case of large deformations, see also p.348-349 of [9], resulting in a lack of convergence for some variants of MPM. As shown below in numerical experiments MPM errors may not

always have a clear asymptotic behavior in this case and this makes explicitly modeling the error equations attractive using the error transport approach [2] which has its origins in [4, 6]. Two fundamental errors in MPM are in the mappings from particles to nodes and back again. This includes differentiating a function defined at particles to get derivatives at nodes and vice versa from nodes to particles. As the mapping from stress particles values to acceleration values at nodes involves the evaluation of an integral this error is sometimes referred to as a quadrature error. Some theoretical results [1, 12, 13] show that the error in this integral may be constant or even $O(1/h)$. The use of linearity preserving extensions to MPM allows the overall error in a modified form of this mapping to be $O(h)$, see [5]. This result is further clarified here. The other significant error comes from particles crossing grids and is known as the grid-crossing error [13]. This error can reduce the accuracy if there are large numbers of grid crossing. Experiments described below also show that the error estimator of [5] still works under these circumstances. While the results in [5] are encouraging, it is important to consider higher space dimensions. In this case it is possible to use the ideas behind the error estimator of [5] to produce a simple error estimator in 2D. This derivation is shown for the model problem of [11] and preliminary error estimates used to estimate displacement errors.

2 Model Problem

A standard MPM model used here follows [5, 7] in being a pair of equations connecting velocity v, displacement u, stress σ and density ρ (here assumed constant):

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

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

with a linear stress model $\sigma = \hat{E}\frac{\partial u}{\partial x}$ for which Young's modulus, E, is constant, a body force b and with appropriate boundary and initial conditions. 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

$$a = x_0 < x_1 < ... < x_N = b, \tag{3}$$

$$h = x_i - x_{i-1}. \tag{4}$$

It will also be assumed that periodic boundary conditions exist, together with appropriate initial conditions. The approach used corresponds to a stress-last [3] or a Symplectic Euler method [5]. Following [3] it is preferable to increment stress last and to use the GIMP method for spatial discretization. It will also be assumed that are n_p particles initially 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 displacement at the pth particles will be written as $u_p^n = u(x_p^n, t^n)$. The initial volume of the particles is uniform for the n_p particles in an interval. The particle volumes are defined using the deformation gradient, F_p^n, and the initial particle volume,V_p^0,

$$V_p^n = F_p^n V_p^0, \text{ where } V_p^0, = \frac{h}{n_p}, \ where\ F_p^0 = 1 \tag{5}$$

At the beginning of each subsequent step it is assumed that particle positions and velocities x_p^n and v_p^n, stresses and deformation gradients σ_p^n and F_p^n exist. There are two key mappings in MPM. The first is an interpolation mapping from particles to grid (and back again). For example for the grid (v_i^n) and particle velocities (v_p^n) at time t_n possible mappings are:

$$v_i^n = \sum_p S_{pi}^n v_p^n \quad (and\ v_p^n = \sum_i S_{ip}^n v_i^n) \tag{6}$$

In this case 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 second mapping uses particle values to compute a derivative at the nodes (or nodal values to compute a derivative at the particles). In other words that

$$\frac{\partial v}{\partial x}(x_i, t_n) = \sum_p D_{pi}^n v_p^n, \quad (or\ \frac{\partial v}{\partial x}(x_p, t_n) = \sum_i D_{ip}^n v_i^n) \tag{7}$$

Examples of how these mappings are used in MPM are in calculating acceleration and velocity at mesh nodes using particle values. The calculation of the acceleration in MPM at the nodes requires the calculation of the volume integral of the divergence of the stress using the equations (and ignoring external forces)

$$a_i^n = \frac{-1}{m_i} \sum_p D_{pi}^n \sigma_p^n F_p^n V_p^0 \tag{8}$$

The negative sign arises as a result of using integration by parts [7]. The nodal velocity v_i is defined by a mapping using existing particle velocities v_p^n by

$$v_i = \sum_p S_{pi}^n \frac{m_p}{m_i} v_p^n \tag{9}$$

in what is a momentum weighted interpolation mapping in which nodal mass is also calculated using the S_{ip}^n coefficients [5] and so gives a partition of unity mapping in that the sum of the effective mapping coefficients is one.

$$\sum_p S_{pi}^n \frac{m_p}{\sum_p S_{pi}^n m_p} = 1 \tag{10}$$

For this interpolation mapping linearity preservation further means that $\sum_p S_{pi}^n x_p^n = x_i$. The interpolation error in this mapping from particles to nodes, as denoted by EIv_i^n, may be approximately written as [5] (where errors in v_p^n are ignored)

$$EIv_i^n \approx C_{EVi} h^2 + h.o.t. \tag{11}$$

for some constant C_{EVi}.

3 Acceleration Error and Quadrature Errors

In the case of the acceleration error we have the result [5] that

$$Ea_i^n = C_h h^k - \frac{1}{m_i}\sum_p D_{pi}^n (E\sigma_p^n F_p^n + \sigma_p^n E F_p^n) V_p^0 \tag{12}$$

Where $E\sigma_p^n$ and EF_p^n are the stress and deformation gradient errors at time t_n and $k = 1 or 2$ depending on the particle distribution [5]. This order h error is different to that of [12, 13, 1] and indeed the coefficients D_{pi}^n is $O(1/h)$. However if the functions $E\sigma_p^n F_p^n$ and $\sigma_p^n EF_p^n$ are differentiable then the right hand term will not be $O(1/h)$ as multiplying by the coefficients D_{pi}^n will give an approximation to the derivative of these terms.

The standard form of the quadrature error for the force, E_f, as used by [12, 13, 1] is

$$E_f = \int_{Omega} \sigma(x) \frac{d\phi_i}{dx} d\Omega - \sum_p \sigma_p \frac{d\phi_{ip}}{dx} V_p \tag{13}$$

Where $\phi_i(x)$ is the mesh basis function associated with the spatial mesh point x_i at which the force is estimated. Steffen et al. with a simple example based upon constant stresses and Baum both show that this error is $O(\Delta x/h)$ thus giving a constant error. Baum also show errors of $O(1/h)$ are possible. Both approaches consider particle volumes that may partially sit outside the range of the basis function $\phi_i(x)$. The analysis in both these cases is correct and so an alternative is to consider the full form of the force error:

$$E_f^{full}(x_i, t) = \frac{\partial \sigma}{\partial x}(x_i, t) - \sum_p -\sigma_p \frac{d\phi_{ip}}{dx} V_p \tag{14}$$

The coefficients used may be modified by using linearity preservation which requires that a constant differentiate to one and a function x differentiates to 1. So if the original mapping is given by

$$\frac{\partial \sigma}{\partial x}(x_i, t) \approx \sum_p D_{pi}^* \sigma_p \tag{15}$$

where $D_{pi}^* = -\frac{d\phi_{ip}}{dx} V_p$ then the linearity preserving mapping is given by

$$\frac{\partial \sigma}{\partial x} \approx \sum_p D_{pi} \sigma_p \tag{16}$$

where

$$D_{pi} = \frac{D_{pi}^* - S_{pi} \sum_p D_{pi}^*}{\sum_p (D_{pi}^* - S_{pi} \sum_p D_{pi}^*) x_p} \tag{17}$$

So that differentiating the function x defined at the particle points x_p gives the value of one at x_i from

$$1 = \sum_p D_{pi} x_p \tag{18}$$

and that the differential of a constant is zero

$$0 = \sum_p D_{pi} \tag{19}$$

between them these equations give at least first order accuracy, [5] for the differential operator.

4 Summary of Existing Error Results

One challenge of attempting analysis for the MPM is that the order of accuracy may vary greatly depending on how much particles move away from their (often) evenly spaced initial distribution. The following results from [5] do not explicitly include the impact of grid crossing errors which may add an $O(dt^2)$ error [12, 13] per grid crossing, but nevertheless seem to work in this case also, as will be shown in Section 5.1 below. The velocity update equation at particles is given in terms of the nodal acceleration a_i^n by

$$v_p^{n+1} = v_p^n + dt \sum_i S_{ip}^n a_i^n \tag{20}$$

The associated global error is Ev_p^{n+1}

$$Ev_p^{n+1} = Ev_p^n + dt \sum_i S_{ip}^n Ea_i^n + dtTES_{ip}^n + lev_p^n \tag{21}$$

where lev_i^n is the $O(dt^2)$ error associated with time integration and where TES_{ip}^n is the $O(h^2)$ error associated with the mapping coefficients S_{ip}. Ignoring the two rightmost terms as being of higher order and realizing that this implies that Ev_p^n is $O(ndt\ h)$ or $Ev_p^n \approx Cv_p^n(ndt\ h) + h.o.t$ then leads to the equation

$$CEv_p^{n+1}((n+1)dt\ h) \approx CEv_p^n(n\ dt\ h) + CEa_i^n(dt\ h) \tag{22}$$

where CEv_p^n, CEv_p^n and CEa_i^n are appropriate constants. Hence at the end of of integration the total number of steps dt add up to the range of integration giving

$$Ev_p^{nsteps} = CEv_p^{nsteps}\ h + h.o.t. \tag{23}$$

The velocity update equation at nodes is given by

$$x_p^{n+1} = x_p^n + dtv_p^{n+1} \tag{24}$$

The global errors in the nodal velocity Ex_p^{n+1} associated with this equation are given by:

$$Ex_p^{n+1} = Ex_p^n + dtEv_p^{n+1} + lex_p^n \tag{25}$$

where lex_p^n is the $O(dt^2)$ local error associated with time integration. Ignoring this higher order term and substituting from equation (22) gives

$$Ex_p^n \approx CEx_p^n(\frac{1}{2}(n(n+1))dt^2h) + h.o.t \tag{26}$$

as

$$CEx_p^{n+1}(\frac{1}{2}(n+1)(n+2)dt^2h) \approx CEx_p^n(\frac{1}{2}(n(n+1))dt^2h) + CEV_p^{n+1}((n+1)dt^2h) \quad (27)$$

Which again using the sum over the total number of steps $nsteps$ gives

$$Ex_p^{nsteps} = CEx_p^{nsteps}(h) + h.o.t \quad (28)$$

This analysis also applies if the acceleration error is a different power of h, as is the case in Section 5.1 in which the particle displacements are small from an initial even mesh.

5 MPM MODEL 1D PROBLEM

In order to test the estimates derived above the vibrating bar example that is often a standard MPM benchmark problem is used, e.g. [7]. 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 [12, 7, 5] and numerous others. The stress equation is

$$\sigma = P = E\frac{\partial u}{\partial x} = E(F-1), \quad (29)$$

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

$$\dot{\sigma} = E(\dot{F}), \quad (30)$$
$$= E(lF), \quad (31)$$

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

$$u(x,t) = Asin(\frac{2\pi x}{l})sin(\frac{c\pi t}{l}), \quad (32)$$
$$v(x,t) = = \frac{Ac\pi}{l}sin(\frac{2\pi x}{l})cos(\frac{c\pi t}{l}), \quad (33)$$

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 29 and the body force is,

$$b(x,t) = 3A(c\pi)^2u(x,t). \quad (34)$$

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.

5.1 Observed Order of Accuracy

The error index of the estimated error norm computed from the error estimation approach in [5] is given by

$$E.Indx_p = \frac{\sum_{k=1}^{nsteps} ||Ex_p^k||_2}{\sum_{k=1}^{nsteps} ||E_{true}x_p^k||_2} \quad (35)$$

and $E.Indv_pi$ and $E.Ind\sigma_p$ are similarly defined. 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. The form of the error stated above now makes it possible to check the derived error accuracy against the observed error. The model problem used in that stated above in [7, 5] except that the parameter A that multiplies the error does not depend on the mesh spacing h as in [5] and so the errors are one power of h lower than in that paper. The errors are measured by taking the L2 norm at each time step in each quantity and then finding its maximum value over all the time steps. These results show the

Table 1: Errors of MPM GIMP for dt =2.5e-5 with 4 particles per cell.

	E=4 A=2.5e-4				E=1000 A=2.5e-2			
h	Err x_p	Err v_p	Err σ_p	NGrX	Err x_p	Err v_p	Err σ_p	NGrX
1/20	3.6e-6	5.8e-5	3.9e-4	8	6.6e-4	2.6e-1	8.6	1.7e+4
1/40	8.7e-7	1.4e-5	1.9e-4	8	3.4e-4	5.6e-2	6.3	6.5e+4
1/80	2.4e-7	3.7e-6	9.4e-5	8	3.4e-4	4.5e-2	6.1	2.6e+5
1/160	6.1e-8	9.9e-7	4.5e-5	8	3.4e-4	5.2e-2	6.1	1.0e+6
h	E.Indx_p	Er.Indv_p	E.Indσ_p	NGrX	E. Indx_p	E.Indv_p	E.Indσ_p	NGrX
1/20	0.89	0.14	1.6	8	0.78	0.52	3.4	1.7e+4
1/40	1.2	0.56	2.2	8	0.86	1.12	2.24	6.5e+4
1/80	1.7	1.9	3.1	8	0.72	0.59	1.72	2.6e+5
1/160	2.6	5.3	4.4	8	0.68	0.51	1.2	1.0e+6

second order accuracy of the displacement, velocity and acceleration errors (not shown) and first order accuracy of the stress error as expected in the case of small displacement (A=0.25e-4). In the large displacement case (A=0.25e-2), for the finer meshes there is little improvement in accuracy, because of the increased number of grid crossings (NGrX) and hence additional errors introduced , see [12]. For large value of the Young's Modulus $\hat{E}$ the Stress (and acceleration - not shown for brevity) errors are much larger than for smaller values of the modulus $\hat{E}$ [5].

6 A TWO-DIMENSIONAL PROBLEM

In extending the above ideas to two and more dimensions, the starting point is a simple expanding and contracting ring problem considered in Section 4.2 of [11]. A snapshot of the solutions of this problem given by the code from [11] is given by Figure 1. In this problem the equation for accelerations in x and $y, (a_x, a_y)$ at the kth node x_i, y_j is given in terms of differentiation matrices in x and y, D_x, D_y and the stresses $\sigma_{i,j}$ at a particle position (x_p, y_p)

$$\begin{bmatrix} a_x \\ a_y \end{bmatrix} = \frac{1}{m_k} \begin{bmatrix} D_x & 0 & D_x \\ 0 & D_y & D_y \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} \tag{36}$$

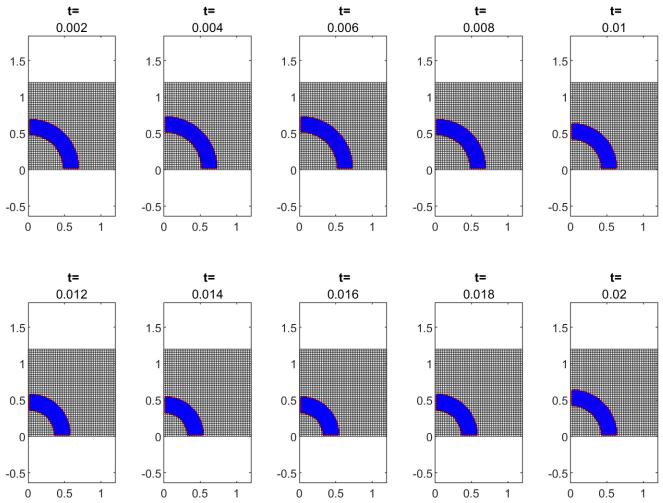

Figure 1: Example Run for 2D Ring problem

where m_k is the mass at node k. The stresses themselves are defined by

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} = \lambda \; log(det(F))/det(F) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \mu \; det(F) \left[FF^T + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right] \tag{37}$$

Where F is the deformation gradient tensor defined by in tensor notation as

$$F_{ij} = \delta_{ij} + u_{ij} \tag{38}$$

where $u_{ij} = \frac{\partial u_i}{\partial x_j}$ and u is the displacement and x is the reference configuration. The update equation for F is given by

$$F(t + dt) = [I + dtL(t + dt)] \, F(t) \tag{39}$$

where $L = \frac{\partial v_i}{\partial x_j}$ is matrix of velocity derivatives calculated from the particle velocities. Further details of the problem are given by [11] and [15].

6.1 Accuracy Linearity Preservation and Least Squares in 2D

Linearity preserving approach for Smooth particle Hydrodynamics Methods are described perhaps in the most detail by [8]. They are introduced for MPM by [7] and used extensively by [5] for accuracy and error estimation in 1D. One advantage of linearity preservation is that it naturally provides a way to estimate errors by looking at the quadratic terms. For example, the acceleration equations obtained by expanding equation (36) may be be written as

$$a_x = \sum_p \hat{D}^x_{pi} f_x(x_p, y_p) \tag{40}$$

$$a_y = \sum_p \hat{D}^y_{pi} f_y(x_p, y_p) \tag{41}$$

For linearity preservation the conditions required are [8]

$$\sum_p D^x_{pi} = 0 \sum_p D^x_{pi} x_p = 1 \sum_p D^x_{pi} y_p = 0 \tag{42}$$

$$\sum_p D^y_{pi} = 0 \sum_p D^y_{pi} y_p = 1 \sum_p D^y_{pi} x_p = 0 \tag{43}$$

Currently the code of [11] does not implement linearity preservation, but the methodology used in [5] still allows the key errors to be estimated, as will be shown below. Another approach that similarly makes it possible to estimate errors is the least squares approach used in a number of MPM codes, [15, 14, 9]. This method represents the solution locally as say a linear polynomial in 2D as $ax + by + c$. The error then consists of quadratic and higher terms. There is clearly scope for using both these approaches in error estimation for MPM in 2D and 3D as it is clear what the next terms are that make up the error.

6.2 Observed Order of Accuracy

The results in [11] show the order of accuracy for this problem in a qualitative way the results in Table 2 show the observed order of accuracy. These results show that

Table 2: Errors of MPM METHODS dt =2.0e-5. at end of integration $t = 0.02$

	UGIMP			CPGIMP			CPDI		
Mesh	Err x_p	Err v_p	Err A	Err x_p	Err v_p	Err A	Err x_p	Err v_p	Err A
12x12	4.1e-4	5.5e-1	1.9e+4	4.4e-4	5.4e-1	3.5e+5	4.1e-4	0.5	2.5e+2
24x24	2.1e-4	2.0e-1	3.9e+3	1.6e-4	1.9e-1	7.0e+2	1.7e-4	1.9e-1	1.9e+2
48x48	1.0e-4	6.4e-2	1.4e+3	7.8e-5	5.5e-2	3.9e+2	6.2e-5	5.8e-2	1.6e+2
96x86	4.1e-5	1.7e-2	1.1e+3	2.6e-5	1.4e-2	2.1e+2	8.9e-6	1.4e-2	7.6e+1

displacement errors are first (UGIMP) or a mixture of first or a mixture of first and second order (CPDI and CPGIMP) while velocity errors are first order except for CPDI and acceleration errors are first order at best.

6.3 Error Estimation of Acceleration Errors

The general approach is as in [5] in that nodal acceleration derivatives are used to estimate the error in the acceleration at nodes. The starting point is a simple Taylor expansion of the x component of the force f_x as an example.

$$f_x(x_p, y_p) = f_x(x_i, y_j) + (x_p - x_i)\frac{\partial f_x}{\partial x}(x_i, y_j) + (y_p - y_i)\frac{\partial f_x}{\partial y}(x_i, y_y) +$$

$$\frac{(x_p - x_i)^2}{2}\frac{\partial^2 f_x}{\partial x^2}(x_i, y_j, t^n) + \frac{(y_p - y_j)^2}{2}\frac{\partial^2 f_x}{\partial y^2}(x_i, y_j, t^n) + \frac{(x_p - x_i)(y_p - y_j)}{2}\frac{\partial^2 f_x}{\partial x \partial y}(x_i, y_j, t^n) + ... \tag{44}$$

substituting from equation (40) gives

$$a_x^n(x_i,y_j) = \sum_p DS_{pi}^n[f_x(x_i,y_j) + (x_p - x_i)\frac{\partial f_x}{\partial x}(x_i,y_j)$$
$$+(y_p - y_i)\frac{\partial f_x}{\partial y}(x_i,y_j) + \frac{(x_p-x_i)^2}{2}\frac{\partial^2 f_x}{\partial x^2}(x_i,y_j,t^n)+$$
$$\frac{(y_p - Y_i)^2}{2}\frac{\partial^2 f_x}{\partial y^2}(x_i,y_j,t^n) + \frac{(x_p-x_i)(y_p-y_j)}{2}\frac{\partial^2 f_x}{\partial x \partial y}(x_i,y_j,t^n) + ...] \quad (45)$$

Subtracting $a_x^n(x_i,y_j) = \frac{\partial f_x}{\partial x}(x_i,y_i)$ after only keeping the first two terms gives

$$Ea_{ix}^n = \sum_p DS_{pi}^n f_x(x_i,y_j) + [\sum_p DS_{pi}^n(x_p - x_i) - 1]a_x^n(x_i,y_j) \quad (46)$$

A similar estimate for the acceleration in y is used

$$Ea_{iy}^n = \sum_p DS_{pi}^n f_x(x_i,y_j) + [\sum_p DS_{pi}^n(y_p - y_j) - 1]a_y^n(x_i,y_j,t^n) \quad (47)$$

As the acceleration errors dominate the propagation of the particle velocity error is neglected

$$Ev_p^{n+1} = dt\sum_i S_{pi}^n[Ea_{ix}^n, Ea_{iy}^n]^T, \quad (48)$$

and estimate the two components of the displacement error by using

$$Ex_p^{n+1} = Ex_p^n + dtEv_p^{n+1} \quad (49)$$

7 COMPUTATIONAL ERROR ESTIMATION RESULTS

Table 3: Average Errors Indices of MPM GIMP dt =2.0e-5.

	UGIMP	CPGIMP	CPDI
Mesh	E.Indx_p	E. Indx_p	E.Indx_p
12x12	0.69	0.72	10
24x24	1.2	1.33	1.8
48x48	2.0	2.2	3.5
96x96	3.3	3.4	7.6

Table 3 shows the error indices for the displacement error in the 2D case for the three methods in the code of [11]. Overall, and perhaps surprisingly given its simplicity, the error estimator described above does remarkably well in terms of estimating the displacement error. The performance of the error is shown in Figure 2.

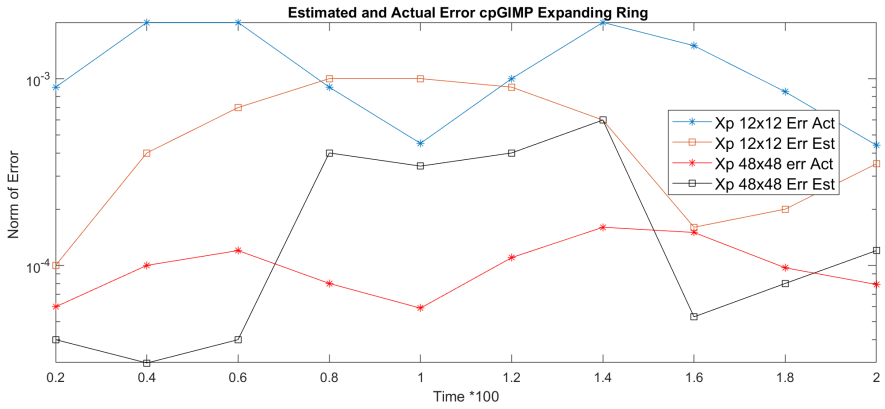

Figure 2: CPGIMP Example Error Estimates for 2D Ring problem

8 CONCLUSIONS

In this paper an introductory discussion has been provided with regard to the errors in the material point method with a view to understanding better the error estimation approaches that may be used in the future to build on available theoretical results. While clearly there is much more work to be done the present approach is a promising start. There is still more work to be done in estimating the errors in all the different solution components. Stress errors being the most challenging.

REFERENCES

[1] Baumgarten A.S. Kamrin K. Analysis and mitigation of spatial integration errors for the Material Point Method. *Int J Numer Methods Eng.* 2023;149

[2] Banks J.W., Hittinger J.M. Connors J.M. Woodward C.S. Numerical error estimation for nonlinear hyperbolic PDEs via nonlinear error transport. *Comput. Meths. Appl. Mech Engnrg.* 213-216, 2012, 1-15.

[3] Bardenhagen S. Kober E., The generalized interpolation Material Point Method, *Computer Modeling in Engineering and Science*, 5 2004, 477-495.

[4] Berzins M. Global Error Estimation in Parabolic Equations Using the Method of Lines. *SIAM J. on Sci. Comput.* 9,4,1988.

[5] Berzins M. Computational error estimation for the Material Point Method. *Computational Particle Mechanics* 10, pages 865-886 (2023).

[6] Berzins, M., "Temporal Error Control for Convection-Dominated Equations in Two Space Dimensions", *SIAM Journal on Scientific Computing*, 16,3, 558-580,1995.

[7] Gritton C. Berzins M. Improving accuracy in the MPM method using a null space filter, *Computational Particle Mechanics*, 4, pp. 131-142 2017.

[8] Huerta, A., Vidal, Y. Bonet, J., Updated Lagrangian formulation for corrected Smooth Particle Hydrodynamics, *Int. J.of Computat. Meths*,3, 4, 383-399, 2006

[9] Nguyen V.P. , de Vaucorbeil A. Bordas S. The Material Point Method Theory, Implementations and Applications Springer Cham, book series: Scientific Computation, ISBN 978-3-031-24069-0 Published: 12 April 2023

[10] Solowski W.T., Berzins M., Coombs W.H, Guilkey J.E, Moller M., Tran Q.A., Adibaskoro T, Seyedan A., Tielen R., and Soga K. Material Point Method: overview and challenges ahead. Chapter 2 of *Advances in Applied Mech.* Vol. 54, 113-204, 2021.

[11] Sadeghirad A., Brannon R.M. Burghardt J. A convected particle domain interpolation technique to extend applicability of the Material Point Method for problems involving massive deformations *Int. J. Numer. Meth. Engng* 86, 1435-1456, 2011.

[12] 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 Engnrng*, 76, 6, 922–948, 2008.

[13] 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.

[14] Thielmann, M., May, D.A., Kaus, B.J.P. Discretization errors in the hybrid finite element particle-in-cell method. *Pure Appl. Geophys.* 171, 21652184 (2014).

[15] Q. A. Tran, W. Solowski, M. Berzins, J. Guilkey. A convected particle least square interpolation Material Point Method, *Int. J.for Numer. Meths. in Engnrng.*, Wiley, October, 2019.

[16] 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): S.P.A. Bordas, D. S. Balint, *Adv. in Appl. Mech.,* Elsevier, 2020 **53**, 2020, 185-398.