Tran, Quoc; Berzins, Martin; Sołowski, Wojciech Tomasz

An improved moving least squares method for the Material Point Method

Published in:
Proceedings of the 2nd International Conference on the Material Point Method for Modelling Soil-Water-Structure Interaction (MPM 2019)

Published: 10/01/2019

Document Version
Peer reviewed version

2nd International Conference on the Material Point Method for Modelling Soil-Water-Structure Interaction

Anura3D
MPM Research Community

Paper number: mpm2019.xxx
www.anura3d.com
www.mpm2019.eu

An improved moving least squares method for the Material Point Method

Quoc-Anh Tran[1*], Martin Berzins[2], Wojciech Sołowski[1]

[1]Department of Civil Engineering, Aalto University, Finland
[2] SCI Institute, University of Utah, Salt Lake City, USA

[*] E-mail: tran.quocanh@aalto.fi

ABSTRACT

The paper presents an improved moving least squares reconstruction technique for the Material Point Method. The moving least squares reconstruction (MLS) can improve spatial accuracy in simulations involving large deformations. However, the MLS algorithm relies on computing the inverse of the moment matrix. This is both expensive and potentially unstable when there are not enough material points to reconstruct the high-order least squares function, which leads to a singular or an ill-conditioned matrix. The shown formulation can overcome this limitation while retain the same order of accuracy compared with the conventional moving least squares reconstruction. Numerical experiments demonstrate the improvements in the accuracy and comparison with the original Material Point Method and the Convected Particles Domain Interpolation method.

KEY WORDS: improved moving least squares, Material Point Method

INTRODUCTION

The Material Point method (MPM) (Sulsky et al., 1994) has been widely used to simulate the large deformation behaviour of solid continuum-based materials. In MPM, the continuum body is discretized into Lagrangian elements referred to "material points". These material points are placed in a background grid which is used to solve the equations of motion. Although MPM is a powerful tool in computational mechanics, MPM still suffered from shortcomings such as grid-crossing and quadrature errors. These errors are reduced in the newer formulations of MPM, such as the Generalized Interpolation Material Point Method (Badenhagen and Kober, 2004), the MPM using B-spline shape functions (Steffen et al., 2008a; Tielen et al., 2017; Gan et al., 2017) or the Convected Particles Domain Interpolation (CPDI) (Sadeghirad et al., 2011). Many studies have shown that when grid-crossing occurs, the MPM solutions can either be non-convergent or reduce the convergence rate when refining grids with the spatial convergence rate varying between first and second order (Wallstedt and Guilkey, 2007; Steffen et al., 2008a; Steffen et al., 2008b; Charlton et al., 2017).

To improve the accuracy of MPM, Wallstedt and Guilkey (2009) and Sulsky and Gong (2015) showed that the moving least squares function reconstruction (MLS) could be used to achieve the higher accuracy in large deformation. However, the formulation requires the inversion of the moment matrix on the solution of a system of equations, therefore, this is both expensive and there is a possibility that the moment matrix may be singular or ill-conditioned. Wallstedt and Guilkey (2009) showed that the nodes in the background grid can have spurious values due to the ill-conditioning of the moment matrix and proposed a method to detect the ill-conditioned nodes. In this paper, we propose an alternative method for function reconstruction within the MPM framework, this method is based on the improved moving least squares (IMLS), which has been applied to the element-free Galerkin method framework (Liew et al., 2006; Zhang et al., 2014; Zhang and Liew, 2014), to overcome the instability of the conventional MLS. The new formulation does not require the inversion of the moment matrix so it can avoid the singlar or ill-conditioned matrix in MLS. To evaluate the proposed formulation, the method of manufactured solutions is used to compare the spatial convergence rate of the solution with MPM and CPDI interpolation.

IMPROVED MOVING LEAST SQUARES RECONSTRUCTION

Improved moving least squares shape function

The MLS approximation of an unknown function $u_h(\mathbf{x})$ is:

$$u_h(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x})\, a_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{1}$$

In which **p(x)** is the polynomial basis vector and **a(x)** is the unknown coefficient vector and m is number of basis functions. Given a set of N data points with position $\{x\}_{p=1}^{N}$ and values of these points $\{u\}_{p=1}^{N}$, the coefficient vector can be computed by minimizing the weighted least square error J given by:

$$J = \sum_{p=1}^{N} w(\mathbf{x} - \mathbf{x}_p)\left[\mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) - u_p\right]^2 \tag{2}$$

where $w(\mathbf{x} - \mathbf{x}_p) > 0$ is a positive weight function. In this paper, a quadratic Bspline function (eq.(27)) below will be used for density, body forces and stress reconstruction while a linear basis function is selected for the velocity reconstruction. Then, differentiating the weighted least square error J with respect to **a(x)**, we obtain:

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{U}_p \tag{3}$$

where **A** the moment matrix, the vector **B** and the data vector $\mathbf{U}_p$ are defined by:

$$\mathbf{A}(\mathbf{x}) = \sum_{p=1}^{N} w(\mathbf{x} - \mathbf{x}_p)\mathbf{p}(\mathbf{x}_p)\mathbf{p}^T(\mathbf{x}_p) \tag{4}$$

$$\mathbf{B}(\mathbf{x}) = [w(\mathbf{x} - \mathbf{x}_1)\mathbf{p}(\mathbf{x}_1) \quad w(\mathbf{x} - \mathbf{x}_2)\mathbf{p}(\mathbf{x}_2) \quad \dots \quad w(\mathbf{x} - \mathbf{x}_N)\mathbf{p}(\mathbf{x}_N)] \tag{5}$$

$$\mathbf{U}_p = [u_1\, u_2 \dots u_N]^T \tag{6}$$

For $\forall$ f(**x**) and g(**x**), in MPM applying least squares reconstruction (Wallstedt and Guilkey, 2009; Sulsky and Gong, 2015), an inner product of two polynomials f and g is computed using approximate quadrature rule given by:

$$\langle f, g\rangle = \int f(\mathbf{x})g(\mathbf{x})dx \approx \sum_{p=1}^{N} w(\mathbf{x} - \mathbf{x}_p)f(\mathbf{x}_p)g(\mathbf{x}_p) \tag{7}$$

From eq.(7), eq.(3) can be rewritten in the matrix form as:

$$\begin{bmatrix} \langle p_1, p_1\rangle & \langle p_1, p_2\rangle & \dots & \langle p_1, p_m\rangle \\ \langle p_2, p_1\rangle & \langle p_2, p_2\rangle & \dots & \langle p_2, p_m\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle p_m, p_1\rangle & \langle p_m, p_2\rangle & \dots & \langle p_m, p_m\rangle \end{bmatrix} \begin{bmatrix} a_1(\mathbf{x}) \\ a_2(\mathbf{x}) \\ \vdots \\ a_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \langle p_1, u_p\rangle \\ \langle p_2, u_p\rangle \\ \vdots \\ \langle p_m, u_p\rangle \end{bmatrix} \tag{8}$$

If the basis function set $p_i(\mathbf{x})$ consists of weighted orthogonal function, the orthogonality condition is satisfied such that:

$$\begin{cases} \langle p_i, p_j\rangle \neq 0, & i = j \\ \langle p_i, p_j\rangle = 0, & i \neq j \end{cases} \tag{9}$$

Then, the eq.(8) becomes:

$$\begin{bmatrix} \langle p_1, p_1\rangle & 0 & \dots & 0 \\ 0 & \langle p_2, p_2\rangle & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \langle p_m, p_m\rangle \end{bmatrix} \begin{bmatrix} a_1(\mathbf{x}) \\ a_2(\mathbf{x}) \\ \vdots \\ a_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \langle p_1, u_p\rangle \\ \langle p_2, u_p\rangle \\ \vdots \\ \langle p_m, u_p\rangle \end{bmatrix} \tag{10}$$

Then, the coefficient **a(x)** can be calculated as:

$$a_i(\mathbf{x}) = \frac{\langle p_i, u_p \rangle}{\langle p_i, p_i \rangle}; i = [1, m] \tag{11}$$

Or in matrix form:

$$\mathbf{a}(\mathbf{x}) = \bar{\mathbf{A}}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{U} = \begin{bmatrix} \frac{1}{\langle p_1, p_1 \rangle} & 0 & \dots & 0 \\ 0 & \frac{1}{\langle p_2, p_2 \rangle} & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\langle p_m, p_m \rangle} \end{bmatrix} \mathbf{B}(\mathbf{x})\mathbf{U} \tag{12}$$

Substituting for **a**(x) in equation (1), we get:

$$u_h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\bar{\mathbf{A}}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{U}_p = \mathbf{\Phi}(\mathbf{x})\mathbf{U}_p = \sum_{p=1}^{N} \phi(\mathbf{x})\, u_p \tag{13}$$

where $\mathbf{\Phi}(\mathbf{x})$ is the vector of IMLS shape function and the shape function $\phi(\mathbf{x})$ is computed from:

$$\phi(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x})\left(\bar{\mathbf{A}}(\mathbf{x})\mathbf{B}(\mathbf{x})\right)_{\mathbf{ip}} = \mathbf{p}^T(\mathbf{x})(\bar{\mathbf{A}}\mathbf{B})_{\mathbf{p}} \tag{14}$$

Orthogonal basis functions

In the weighted least squares method (Wallstedt and Guilkey, 2009; Sulsky and Gong, 2015), the monomial polynomials, denoted by **q,** are employed to construct the polynomial basis vector. For example, the 2D monomial basis functions are:

$$\mathbf{p} = [1, x, y, x^2, xy, y^2] \tag{15}$$

In MPM, applying least squares reconstruction, the density, velocity, stress and body forces are constructed from data at material points. However, MPM suffers from quadrature errors because the positions of material points are arbitrary and may not coincide with quadrature points. Therefore, the integration in eq.(7) does not always satisfy the orthogonality condition given by eq.(9). To build a set of polynomials to satisfy the orthogonality condition, the set of polynomials is updated in every time step and depends on the material points coordinates using the Gram Schmidt orthogonalization method. For example, the 2D orthogonal polynomials **p** can be computed from the monomial polynomial **q** in eq. (15) as follows:

$$p_i = q_i - \sum_{k=1}^{i-1} \frac{\langle q_i, p_k \rangle}{\langle p_k, p_k \rangle} p_k \tag{16}$$

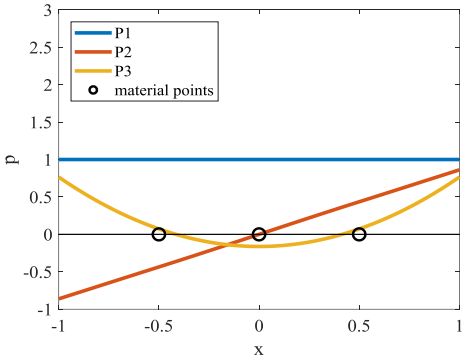

Figure 1 Polynomials in equal distribution of 3 material points with weighted quadratic Bspline

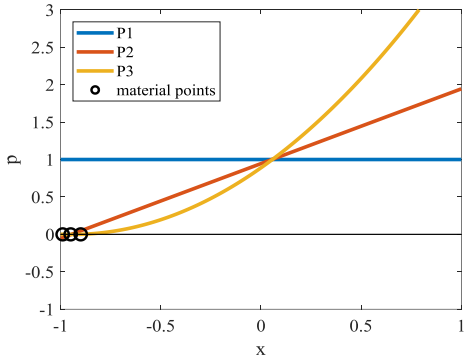

Figure 2 Polynomials in near-corner distribution of 3 material points with weighted quadratic Bspline

Figure 1 and Figure 2 show the change of polynomials in 1D corresponding to 2 different material points coordinates in the interval [-1,1] including equal distribution and near-corner distribution. In this example, the weight function is a quadratic Bspline. In case there is not enough material points data to reconstruct high order function, then the moment matrix A may be singular. Then, given rank(A) = *r* < *m*, the inner products of orthogonal basis functions with order higher

than r become zeros such that:

$$\langle p_i, p_i \rangle = 0, \qquad r < i \leq m \tag{17}$$

In such cases, the coefficient **a(x)** is only calculated up to order *r* using eq.(11). Figure 3 shows the construction of polynomials for 2 material points, using eq.(16). In this case, because $p_3(x_p) = 0$ with $\forall x_p$, using eq. (7) leading to $\langle p_3, p_3 \rangle = 0$. The formulation automatically applies to only the 2 first orthogonal basis functions for the function reconstruction.

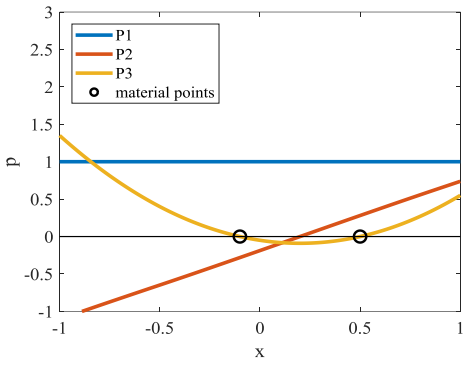

Figure 3 Polynomials for 2 material points, rank(A) = 2 then $\langle p_3, p_3 \rangle = 0$

MATERIAL POINT METHOD WITH IMLS FUNCTION RECONSTRUCTION

Governing equations

The governing equations read:

$$\rho \frac{d\mathbf{v}}{dt} = \nabla . \boldsymbol{\sigma} + \rho \, \mathbf{b} \quad \text{for } \mathbf{x} \in \Omega \tag{18}$$

where $\boldsymbol{\sigma}$ is the stress tensor, ρ is the density, **b** is the body acceleration, **v** is the velocity vector and **x** is the coordinate vector. The Dirichlet and Neumann boundary conditions are:

$$\mathbf{u} = \hat{\mathbf{u}} \quad \text{for } \mathbf{x} \in \Omega_u \tag{19}$$

$$\sigma . \mathbf{n} = \tau \quad \text{for } \mathbf{x} \in \Omega_\tau \tag{20}$$

where $\hat{\mathbf{u}}$ is the pre-described displacement vector in the boundary Ω_u, **n** is the normal vector in the boundary Ω_τ and τ is the pre-described traction. The weak form of the governing equations can be written as:

$$\int_\Omega \rho \frac{d\mathbf{v}}{dt} . \mathbf{w} d\Omega = -\int_\Omega \boldsymbol{\sigma} : \nabla \mathbf{w} d\Omega + \int_\Omega \rho \, \mathbf{b} . \mathbf{w} d\Omega + \int_{\Omega_\tau} \boldsymbol{\tau} . \mathbf{w} \, d\Omega_\tau \tag{21}$$

where **w** is a test function. Denoting the acceleration vector **a** and the terms in the right-hand side of the eq.(21) respectively as $\mathbf{f}^{int}$, $\mathbf{f}^{b}$, $\mathbf{f}^{ext}$ the discrete equations in the background grid can be written in a simple form as follows:

$$\mathbf{Ma} = \mathbf{f}^{int} + \mathbf{f}^{b} + \mathbf{f}^{ext} \tag{22}$$

In this paper, the subscript *i* denotes the nodes of the background grid while the subscript *p* denotes material points.

Time integration equations

The numerical solution of the momentum balance eq.(22) is obtained at discrete time steps using explicit time integration (Sulsky et al 1994) where Δt is the time step and N_t is the total number of time steps. Superscript 'L' denotes the solution of nodes at the end of the time step. The solution at both nodes and material points in the current time step is denoted by a superscript 't' and the next time step at the end of the calculation is:

$$t^{t+1} = t^t + dt \tag{23}$$

Initially, the state variables for material points including mass m_p, position $\mathbf{x}_p$, velocity $\mathbf{v}_p$, volume V_p, stress $\boldsymbol{\sigma}_p$, body force $\mathbf{b}_p$, deformation gradient $\mathbf{F}_p$ are generated. At the beginning of the time step, the approximations to the density ρ_p^t, stress $\boldsymbol{\sigma}_p^t$ and body forces $\mathbf{b}_p^t$ are reconstructed using IMLS technique and evaluated at the Gauss quadrature points 'g' as follows:

$$\rho_g^t = \sum_p \phi_{gp} \rho_p^t \tag{24}$$

$$\mathbf{b}_g^t = \sum_p \phi_{gp} \mathbf{b}_p^t \tag{25}$$

$$\boldsymbol{\sigma}_g^t = \sum_p \phi_{gp} \boldsymbol{\sigma}_p \tag{26}$$

where ρ_g^t, $\mathbf{b}_g^t$, σ_g^t are the value of density, body forces, stress at Gauss points. $\phi_{gp} = \phi_g(\mathbf{x}_p)$ is the IMLS shape function of Gauss point 'g' evaluating at position of the material point $\mathbf{x}_p$, computed from eq.(14). For the IMLS shape function of Gauss point, the weight function for the cell size l_e is the quadratic spline function:

$$w(x - x_i) = \begin{cases} \frac{3}{4} - \left(\frac{x - x_p}{2l_e}\right)^2 & \left(\frac{x - x_p}{2l_e}\right) \le \frac{1}{2} \\ \frac{1}{2}\left(\frac{3}{2} - \left(\frac{x - x_p}{2l_e}\right)\right)^2 & \frac{1}{2} \le \left(\frac{x - x_p}{2l_e}\right) \le \frac{3}{2} \\ 0 & \left(\frac{x - x_p}{2l_e}\right) \ge \frac{3}{2} \end{cases} \tag{27}$$

Then, the lumped nodal mass m_i^t, internal forces $\mathbf{f}_i^{b,t}$ and external forces $\mathbf{f}_i^{int,t}$ are computed using Gauss quadrature integration:

$$m_i^t = \sum_g N_{ig} \rho_g^t \omega_g \tag{28}$$

$$\mathbf{f}_i^{b,t} = \sum_g N_{ig} \mathbf{b}_g^t \omega_g \tag{29}$$

$$\mathbf{f}_i^{int,t} = -\sum_g \nabla N_{ig} \boldsymbol{\sigma}_g^t \omega_g \tag{30}$$

where ω_g is the quadrature weight at the Gauss points and $N_{ig} = N_i(\mathbf{x}_g)$, $\nabla N_{ig} = \nabla N_i(\mathbf{x}_g)$ are the linear shape function and the gradient of the shape function of the node 'i' evaluated at Gauss points $\mathbf{x}_g$. The nodal velocity is computed based on the nodal IMLS shape function evaluated at material points $\phi_{ip} = \phi_i(\mathbf{x}_p)$ as:

$$\mathbf{v}_i^t = \sum_p \phi_{ip} \mathbf{v}_p^t \tag{31}$$

where $\mathbf{v}_i^t$ is the nodal velocity, $\mathbf{v}_p^t$ is the material point's velocity and ϕ_{ip} is the IMLS shape function with the weight function being the linear basis function. After that, the nodal acceleration $\mathbf{a}_i^t$ and velocity at the next time step $\mathbf{v}_i^{t+1}$ are calculated from:

$$\mathbf{a}_i^t = \frac{\mathbf{f}_i^{int,t} + \mathbf{f}_i^{b,t} + \mathbf{f}_i^{ext,t}}{m_i^t} \tag{32}$$

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \mathbf{a}_i^t dt \tag{33}$$

After solving the motion equation, the velocities and positions of the material points are updated:

$$\mathbf{v}_p^{t+1} = \mathbf{v}_p^t + \sum_i N_{ip}\mathbf{a}_i^t dt \tag{34}$$

$$\mathbf{x}_p^{t+1} = \mathbf{x}_p^t + \sum_i N_{ip}\mathbf{v}_i^{t+1} dt \tag{35}$$

where $N_{ip} = N_i(\mathbf{x}_p)$is the linear shape function of the node 'i' evaluating material point $\mathbf{x}_p$. The nodal velocity at the end of the time step $\mathbf{v}_i^L$ is calculated using IMLS shape function $\phi_{ip} = \phi_i(\mathbf{x}_p)$ as:

$$\mathbf{v}_i^L = \sum_p \phi_{ip}\mathbf{v}_p^{t+1} \tag{36}$$

The velocity gradients, using the gradient of the shape function, can be written as:

$$\nabla\mathbf{v}_p^{t+1} = \sum_i N_{ip}\mathbf{v}_i^L \tag{37}$$

Subsequently, the quantities such as deformation gradient $\mathbf{F}_p^{t+1}$, volume of the material point V_p^{t+1} and density of the material points ρ_p^{t+1} are updated:

$$\mathbf{F}_p^{t+1} = (\mathbf{I} + \nabla\mathbf{v}_p^{t+1} dt)\mathbf{F}_p^t \tag{38}$$

$$V_p^{t+1} = \det\left(\mathbf{F}_p^{t+1}\right) V_p^o \tag{39}$$

$$\rho_p^{t+1} = \frac{m_p}{V_p^{t+1}} \tag{40}$$

Finally, the constitutive model stress point algorithm updates the stress $\boldsymbol{\sigma}_p^{t+1}$ at the material points based on the deformation gradient $\mathbf{F}_p^{t+1}$. At that moment, the grid configuration is reset.

METHOD OF MANUFACTURED SOLUTIONS

To validate the proposed formulation, several numerical solutions are constructed with the method of manufactured solutions. The governing equation (18) can be written in the total and updated Lagrangian forms as follows:

$$\rho_o\mathbf{a} = \nabla.\mathbf{P} + \rho_o\mathbf{b} \tag{41}$$

where ρ_o is the density in reference configuration, **a** is the acceleration vector, **P** is the 1st Piola-Kichhoff Stress tensor, **b** is the body force vector. In this paper, the hyper-elastic neo-Hookean model is used to test the non-linear behaviour. The 1st Piola-Kichhoff stress to the reference configuration, denoted as **X,** can be written as:

$$\mathbf{P} = \lambda\ln(J)\mathbf{F}^{-1} + \mu\mathbf{F}^{-1}(\mathbf{F}\mathbf{F}^T - \mathbf{I}) \tag{42}$$

where J is the determinant of deformation gradient **F**, μ and λ are the shear modulus and Lame constant. The Cauchy stress to the current configuration, denoted as **x** is:

$$\boldsymbol{\sigma} = \frac{\mathbf{P}\mathbf{F}^T}{J} = \frac{\lambda\ln(J)}{J}\mathbf{I} + \frac{\mu}{J}(\mathbf{F}\mathbf{F}^T - \mathbf{I}) \tag{43}$$

To manufacture the solution, a non-linear time-dependent displacement solution **u** is defined. The displacement and velocity field are:

$$\mathbf{u}(X,t) = A\sin\left(2\frac{\pi X}{L}\right)\sin\left(\frac{c\pi t}{L}\right) \tag{44}$$

$$\mathbf{v}(X,t) = \frac{d\mathbf{u}}{dt} = Ac\pi\sin\left(2\frac{\pi X}{L}\right)\cos\left(\frac{c\pi t}{L}\right) \tag{45}$$

where A is the maximum amplitude of displacement, L is size of the unit square (L=1m), X and Y are the position of material points in the reference configuration, c is the wave speed ($c = \sqrt{\frac{E}{\rho}}$). Then the deformation gradient is computed as:

$$\mathbf{F}(X,t) = \mathbf{I} + \frac{\delta \mathbf{u}}{\delta \mathbf{X}} = 1 + 2\frac{A\pi}{L}\cos\left(2\frac{\pi X}{L}\right)\sin\left(\frac{c\pi t}{L}\right) \tag{46}$$

After that, the 1st Piola-Kichhoff stress is computed from the deformation gradient. The acceleration is computed by twice differentiating the displacement **u** by time. Then, substituting the given stress and acceleration in the governing equation eq.(41), the body force can be found. The body forces are used as the source term. The source term is manufactured as the input for the model and the given displacement solution is considered as the exact solution for the spatial convergence rate computation. The body forces required to obtain the displacement and velocity field above are:

$$\mathbf{b}(X,t) = \frac{\pi \mathbf{u}(X,t)}{L^2}\left(4\frac{\mu}{\rho_o} - c^2 - 4\frac{\lambda[\ln(\mathbf{F}(X,t)) - 1] - \mu}{\rho_o \mathbf{F}(X,t)^2}\right) \tag{47}$$

In order to study the spatial convergence of MPM, a 1m-length bar is discretized into 8, 16, 32, 64, 128 cells with 3 material points per cell. This discretization corresponds to cell sizes of 0.125m, 0.0625m, 0.03125m, 0.015625m and 0.0078125m respectively. The material properties include the shear modulus μ = 3.85e^{6}Pa and the Lame constant λ = 5.77e^{6}Pa, initial density ρ_o = 1000 kg/m^3. The time step dt = 1e^{-5}s and the final time of the simulation is 0.02s. The displacement amplitude is 0.1m to generate the grid-crossings. The root-mean-square errors (RMS) for numerical simulations are defined as:

$$\mathrm{RMS} = \sqrt{\frac{\sum_{p=1}^{N_p}\left\|\mathbf{f}_{\mathrm{numerical}}(\mathbf{x}_p) - \mathbf{f}_{\mathrm{exact}}(\mathbf{x}_p)\right\|^2}{N_p}} \tag{48}$$

Figure 4, Figure 5 and Figure 6 show the spatial convergence rates of displacement, velocity and stress of material points respectively. We observe that MPM is non-convergent when grid-crossing occurs. The CPDI interpolation (Sadeghirad et al., 2011) showed a second order of accuracy with the number of grid cells of 8 and 16. However, refining the grid causes a loss of convergence rate as the grid-crossing occurs more frequently with the finner mesh. Our proposed formulation using MPM with IMLS function reconstruction shows second order accuracy, which, appears to be better than the CPDI method. Futher numerical experiments are needed to explain this behaviour.

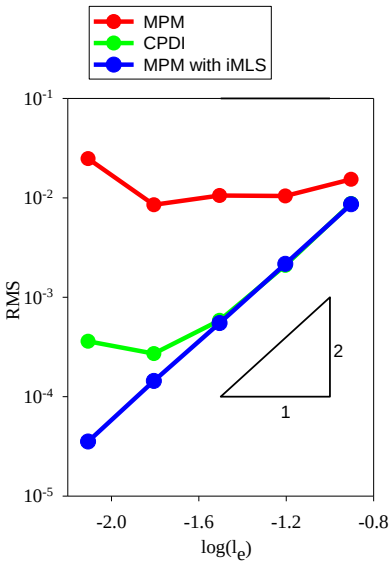

Figure 4 Spatial convergence rate of displacement (A = 0.1m)

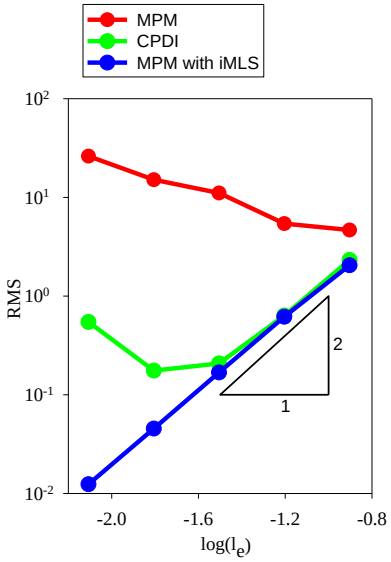

Figure 5 Spatial convergence rate of velocity (A = 0.1m)

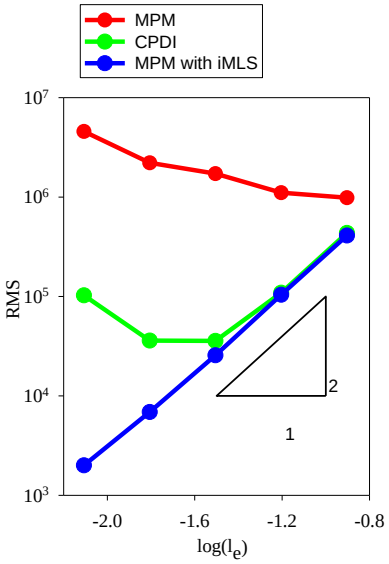

Figure 6 Spatial convergence rate of stress (A = 0.1m)

CONCLUSION

This paper has presented a framework for improving the accuracy of the MPM. In this framework, the data at material points are reconstructed on a background grid using the improved moving least squares (IMLS) function reconstruction. The IMLS reconstruction does not require the inversion of the moment matrix, therefore, it avoids the singular or ill-conditioned problems in the original moving least squares method. After the function is reconstructed on the background grid, it is integrated using Gauss integration, in a way that is similar to the approach used in the Finite Element method. To evaluate the convergence rate of the proposed formulation, a solution is manufactured to obtain the exact solutions of displacement, velocity and stress in the large deformation. Clearly, our method has better accuracy than CPDI method and obtains second order accuracy in the given example. Future work will focus on the performance of the method in the large rotation problems, for example, validating with the generalized vortex problem using the method of manufactured solutions (Kamojjala et al., 2015).

REFERENCES

Badenhagen, S. G., & Kober, E. M. (2004). The generalized interpolation material point method. *Computer Modeling in Engineering & Science, 5*, 477-495.

Charlton, T. J., Coombs, W. M., & Augarde, C. E. (2017). iGIMP: an implicit generalized interpolation material point method for large deformations. *Computers and Structures, 190*, 108-125.

Gan, Y., Sun, Z., Chen, Z., Zhang, X., & Liu, Y. (2017). Enhancement of the material point method using B-spline basis functions. *International Journal for numerical methods in engineering, 113*(3), 411-431.

Gong, M. (2015). *Improving the Material Point Method.* PhD thesis, the University of New Mexico.

Kamojjala, K., Brannon, R., Sedeghirad, A., & Guilkey, J. (2015). Verification tests in solid mechanics. *Engineering with COmputers, 31*(2), 193-213.

Liew, K. M., Cheng, Y., & Kitipornchai, S. (2006). Boundary element-free method and its application to two dimensional elasticity problems. *International journal for numerical methods in engineering, 65*, 1310-1332.

Sadeghirad, A., Brannon, R., & Burghardt, J. (2011). A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. *International Journal for numerical methods in Engineering, 86*, 1435-1456.

Steffen, M., Kirby, R. M., & Berzins, M. (2008). Analysis and reduction of quadrature errors in the material point method. *International Journal for numerical methods in engineering, 76*, 922-948.

Steffen, M., Wallstedt, P. C., Guilkey, J. E., Kirby, R. M., & Berzins, M. (2008). Examination and analysis of implementation coices within the Material Point Method (MPM). *Computer Modeling in Engineering and Sciences, 31*(2), 107-127.

Sulsky, D., & Gong, M. (2016). Improving the Material Point Method. In *Innovative Numerical Approaches for Multi-Field and Multi-Scale Problems* (pp. 217-240).

Sulsky, D., Chen, Z., & Schreyer, H. L. (1994). A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering, 118*(1-2), 179-196.

Tielen, R., Wobbes, E., Möller, M., & Beuth, L. (2017). A high order material point method., (pp. 265-272).

Wallstedt, P. C., & Guilkey, J. E. (2007). Improved Velocity Projection for the Material Point Method. *Computer Modeling in Engineering and Sciences, 19*(3), 223-232.

Wallstedt, P. C., & Guilkey, J. E. (2010). A weighted least squares particle-in-cell method for solid mechanics. *International journal for numerical methods in engineering, 85*(13), 1687-1704.

Zhang, D., Ma, X., & Giguere, P. (2011). Material Point Method enhanced by modified gradienet of shape function. *Journal of Computational Physics, 230*, 6379-6398.

Zhang, L., & Liew, K. (2014). An improved moving least-square Ritz method for two-dimensional elasticity problems. *Applied Mathematics and Computation, 246*, 268-282.

Zhang, L., Deng, Y., & Liew, K. (2014). An improved element-free Galerkin method for numerical modelling of the biological population problems. *Engineering analysis with boundary elements, 40*, 181-188.