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

Anura3D
MPM Research Community

Paper number: mpm2019.07
www.anura3d.com
www.mpm2019.eu

Moving least squares reconstruction for B-spline Material Point Method

Quoc-Anh Tran[1*], Elizaveta Wobbes[2], Wojciech Sołowski[1], Matthias Möller[2], Cornelis Vuik[2]

[1]Department of Civil Engineering, Aalto University, Finland
[2]Department of Applied Mathematics, Delft University of Technology, the Netherlands

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

ABSTRACT

The paper shows a moving least squares reconstruction technique applied to the B-spline Material Point Method (B-spline MPM). It has been shown previously that B-spline MPM can reduce grid-crossing errors inherent in the original Material Point Method. However, in the large deformation regime where the grid crossing occurs more frequently, the convergence rate of B-spline MPMis lower. In this paper, moving least squares reconstruction is employed to retrieve the expected convergence rate. The proposed improvement is examined in terms of the spatial convergence using the methods of manufactured solutions for large deformations.

KEY WORDS: moving least squares, function reconstruction, B-spline basis function, Material Point Method.

INTRODUCTION

Since the introduction of Material Point Method (MPM) by Sulsky et al. (1994), MPM has been used to simulate dynamic problems that involve large deformation. However, the original MPM algorithm suffers from numerical inaccuracies when material points move from one cell to another. To overcome these inaccuracies, known as grid-crossing errors, Bardenhagen and Kober (2004) proposed Generalized Interpolation Material Point Method (GIMP). GIMP defines the shape functions of material points with a characteristic function. The shape-function gradients in GIMP are continuous, which reduce the grid-crossing errors. Later, Zhang et al. (2011) enhanced the MPM with the modified gradient of the shape function, while Steffen et al. (2008) adopted B-spline basis functions for MPM and showed that B-spline MPM could decrease the cell-crossing errors. The accuracy of B-spline MPM was further improved by advanced function reconstruction technique to project data from material points to the background grid (Tielen et al., 2017; Gan et al., 2017; Wobbes et al., 2018).
The paper investigates the spatial convergence rate of B-spline MPM and uses a moving least squares (MLS) function reconstruction technique to improve its performance. Such technique has been applied to the original MPM (Sulsky and Gong, 2016) and Generalized Internpolation Material Point Method (Wallstedt and Guilkey, 2009). The paper investigates the influence of the MLS function reconstruction on the numerical accuracy of B-spline MPM using an example from Sadeghirad et al. (2011). The results show that the use of B-spline basis functions considerably reduces the error made by MPM and can lead to the second order of convergence. However, for finer meshes, a lack of convergence is observed. However, B-Spline formulation coupled with the moving least squares reconstruction technique, converges well for finer meshes and therefore improves the accuracy of B-spline MPM significantly.

B-spline BASIS FUNCTIONS

The construction of B-spline basis functions starts from a so-called knot vector, denoted by $\Xi = \{\xi_1, \xi_{2,\ldots,}\xi_{n+p},\xi_{n+p+1}\}$, where $\xi_1 \leq \xi_2 \leq \ldots \leq \xi_{n+p} \leq \xi_{n+p+1}$ are knots with n and p being the number of basis functions and polynomial order, respectively. If knots are distributed equidistantly, the knot vector is uniform. Otherwise, it is non-uniform. A knot vector is open if the first and last knots are repeated p + 1 times. Polynomial order p = 0, 1, 2, 3 … refer to constant, linear, quadratic, and cubic B-spline shape functions, respectively. Given a knot vector, the B-spline basis function at the i-th knot is denoted as $N_{i,p}$. The zeroth order basis is constant (p = 0):

$$N_{i,0} = \begin{cases} 0, & \xi_i \leq \xi < \xi_{i+1} \\ 1, & otherwise \end{cases} \tag{1}$$

For $p \geq 1$, the basis functions are defined recursively by the (Cox-de Boor, 1971) recursion formula:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{2}$$

B-spline basis functions satisfy the partition of unity property and are non-negative over their entire support. The derivative of the i-th basis function, $N'_{i,p}(\xi)$, can be calculated as follows:

$$N'_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{3}$$

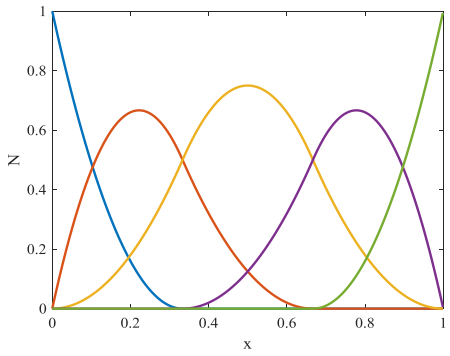

Figure 1 Quadratic B-spline for an open, uniform knot vector $\Xi = \{0,0,0,1/3,2/3,1,1,1\}$

Figure 1 presents the quadratic B-spline basis functions (p = 2) for the open, uniform knot vector $\Xi = \{0,0,0,1/3,2/3,1,1,1\}$. Multivariate B-spline basis functions in two and three dimensions are constructed as the tensor product of one-dimensional, that is, univariate basis functions. Let $\Xi = \{\xi_1, \xi_{2,\ldots,}\xi_{n+p,}\xi_{n+p+1}\}$ and $\Theta = \{\eta_1, \eta_{2,\ldots,} \eta_{m+q,} \eta_{m+q+1}\}$, where $\xi_1 \leq \xi_2 \leq \ldots \leq \xi_{n+p} \leq \xi_{n+p+1}$ and $\eta_1 \leq \eta_2 \leq \ldots \leq \eta_{m+q} \leq \eta_{m+q+1}$ be two knot vectors with polynomial orders p and q and the number of basis functions in each direction being n and m, respectively. The two-dimensional B-spline basis function and its gradient can then be written as:

$$\mathbf{N}_{(i,j),(p,q)}(\xi,\eta) = N_{i,p}(\xi).N_{j,q}(\eta) \tag{4}$$

$$\nabla \mathbf{N}_{(i,j),(p,q)}(\xi,\eta) = \left\{N'_{i,p}(\xi).N_{j,q}(\eta)\ ,\ N_{i,p}(\xi).N'_{j,q}(\eta)\right\} \tag{5}$$

MOVING LEAST SQUARES SHAPE FUNCTION

The moving least squares approximation of an unknown function $u(\mathbf{x})$ is:

$$u_h(\mathbf{x}) = \sum_{i=1}^{n_b} p_i(\mathbf{x})\, a_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{6}$$

in which $\mathbf{p(x)}$ is the column vector of the polynomial basis functions, $\mathbf{a(x)}$ is the column vector containing the unknown coefficients, and n_b is the number of basis functions. Given a set of n_p data points $\{x\}_{p=1}^{n_p}$ and values of these points $\{u\}_{p=1}^{n_p}$, the vector $\mathbf{a(x)}$ can be computed by using a weighted function $w(\mathbf{x} - \mathbf{x}_p) \neq 0$ as follows:

$$J(\mathbf{x}) = \sum_{p=1}^{n_p} w(\mathbf{x} - \mathbf{x}_p)\left[\mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) - u_p\right]^2 \tag{7}$$

Take the derivative of J with respect to each component of **a(x)** and set it to zero to obtain:

$$\sum_{p=1}^{n_p} w(\mathbf{x}-\mathbf{x}_p)\mathbf{p}(\mathbf{x})\left[\mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x})-u_p\right]=0 \tag{8}$$

This leads to the following expression:

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x})=\mathbf{B}(\mathbf{x})\mathbf{U}_p \tag{9}$$

where the matrix **A** and **B** can be written as:

$$\mathbf{A}(\mathbf{x})=\sum_{p=1}^{n_p} w(\mathbf{x}-\mathbf{x}_p)\mathbf{p}(\mathbf{x}_p)\mathbf{p}^T(\mathbf{x}_p) \tag{10}$$

$$\mathbf{B}(\mathbf{x})=\left[w(\mathbf{x}-\mathbf{x}_1)\mathbf{p}(\mathbf{x}_1)w(\mathbf{x}-\mathbf{x}_2)\mathbf{p}(\mathbf{x}_2)\quad\ldots\quad w\left(\mathbf{x}-\mathbf{x}_{n_p}\right)\mathbf{p}\left(\mathbf{x}_{n_p}\right)\right] \tag{11}$$

$$\mathbf{U}_p=\left[u_1\,u_2\,\ldots\,u_{n_p}\right]^T \tag{12}$$

Solving eq.(9) for **a(x)** and substituting it into eq.(6)leads to:

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

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

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

In this paper, **p(x)** is defined as follows:

$$\mathbf{p}(\mathbf{x})=[1,x,y]^T \tag{15}$$

Considering a rectangular grid with the element size l_x x l_y, we denote $r_x=\frac{x-x_p}{2l_x}$, $r_y=\frac{y-y_p}{2l_y}$. Then, the weight function $w(\mathbf{x}-\mathbf{x}_p)$ is:

$$w\left(\mathbf{x}-\mathbf{x}_p\right)=w(r_x)w\left(r_y\right) \tag{16}$$

The quadratic spline function $w(r)$ can be written as:

$$w(r)=\begin{cases}\frac{3}{4}-r^2 & r\le\frac{1}{2}\\ \frac{1}{2}\left(\frac{3}{2}-r\right)^2 & \frac{1}{2}\le r\le\frac{3}{2}\\ 0 & r\ge\frac{3}{2}\end{cases} \tag{17}$$

B-SPLINE MATERIAL POINT METHOD WITH FUNCTION RECONSTRUCTION

The algorithm can be divided into 3 phases. Phase 1 consists of the projection from the material points to the background grid. Phase 2 solves the discrete equation on the grid. Phase 3 updates the solutions from the background grid to the material points and reset the grid configuration. In the phase 1, the data of material points including density ρ_p^t, momentum $\rho\mathbf{v}_p^t$, stress $\boldsymbol{\sigma}_p^t$ and body forces $\mathbf{b}_p^t$ are reconstructed using moving least square

technique and evaluated at the Gauss quadrature points'g' as follows:

$$\rho_g^t = \sum_p \phi_{gp} \rho_p^t \tag{18}$$

$$(\rho \mathbf{v})_g^t = \sum_p \phi_{gp} \rho_p^t \mathbf{v}_p^t \tag{19}$$

$$\mathbf{b}_g^t = \sum_p \phi_{gp} \mathbf{b}_p^t \tag{20}$$

$$\boldsymbol{\sigma}_g^t = \sum_p \phi_{gp} \boldsymbol{\sigma}_p^t \tag{21}$$

where ρ_g^t, $(\rho \mathbf{v})_g^t$, $\mathbf{b}_g^t$, σ_g^t are the value of density, momentum, body forces, stress at Gauss points. |Here, $\phi_{gp} = \phi_g(\mathbf{x}_p)$ is the moving least square shape function of Gauss point 'g' evaluating at position of the material point $\mathbf{x}_p$, computed from eq.(14). Then, the nodal lumped mass m_i^t, nodal momentum $(m\mathbf{v})_i^t$, internal forces $\mathbf{f}_i^{b,t}$ and external forces $\mathbf{f}_i^{int,t}$ are computed using Gauss quadrature integration as follows:

$$m_i^t = \sum_g N_{(i,j),(p,q)}(\mathbf{x_g}) \rho_g^t \, \omega_g \tag{22}$$

$$(m\mathbf{v})_i^t = \sum_g N_{(i,j),(p,q)}(\mathbf{x_g}) (\rho \mathbf{v})_g^t \, \omega_g \tag{23}$$

$$\mathbf{f}_i^{b,t} = \sum_g N_{(i,j),(p,q)}(\mathbf{x_g}) \mathbf{b}_g^t \, \omega_g \tag{24}$$

$$\mathbf{f}_i^{int,t} = -\sum_g \nabla N_{(i,j),(p,q)}(\mathbf{x_g}) \boldsymbol{\sigma}_g^t \, \omega_g \tag{25}$$

where ω_g is the weight of the Gauss point and $N_{(i,j)(p,q)}(\mathbf{x}_g)$ is the B-spline shape function evaluating at position of Gauss point $\mathbf{x}_g$**.** After that, in phase 2 the nodal total force $\mathbf{f}_i^t$, nodal acceleration $\mathbf{a}_i^t$ and nodal momentum in the next time stepare calculated:

$$\mathbf{f}_i^t = \mathbf{f}_i^{int,t} + \mathbf{f}_i^{b,t} + \mathbf{f}_i^{ext,t} \tag{26}$$

$$\mathbf{a}_i^t = \frac{\mathbf{f}_i^t}{m_i^t} \tag{27}$$

$$(m\mathbf{v})_i^{t+1} = (m\mathbf{v})_i^t + \mathbf{f}_i^t \, dt \tag{28}$$

After solving the motion equation, phase 3 updates the velocities and positions of the material points:

$$\mathbf{v}_p^{t+1} = \mathbf{v}_p^t + \sum_i N_{(i,j),(p,q)}(\mathbf{x_p}) \mathbf{a}_i^t dt \tag{29}$$

$$\mathbf{x}_p^{t+1} = \mathbf{x}_p^t + \sum_i N_{(i,j),(p,q)}(\mathbf{x_p}) \frac{(m\mathbf{v})_i^{t+1}}{m_i^t} dt \tag{30}$$

The nodal velocity at the end of the time step $\mathbf{v}_i^L$ is calculated using MLS shape function and B-spline shape function as follows:

$$(\rho \mathbf{v})_g^L = \sum_p \phi_{gp} \rho_p^t \mathbf{v}_p^{t+1} \tag{31}$$

$$(m\mathbf{v})_i^L = \sum_g N_{(i,j),(p,q)}(\mathbf{x_g}) (\rho \mathbf{v})_g^L \, \omega_g \tag{32}$$

$$\mathbf{v}_i^L = \frac{(m\mathbf{v})_i^L}{m_i^t} \tag{33}$$

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

$$\nabla \mathbf{v}_p^{t+1} = \sum_i \nabla N_{(i,j),(p,q)}(\mathbf{x_p})\mathbf{v}_i^L \tag{34}$$

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

$$\mathbf{F}_p^{t+1} = (\mathbf{I} + \nabla \mathbf{v}_p^{t+1} dt)\mathbf{F}_p^t \tag{35}$$

$$V_p^{t+1} = \det(\mathbf{F}_p^{t+1})\, V_p^o \tag{36}$$

$$\rho_p^{t+1} = \frac{m_p}{V_p^{t+1}} \tag{37}$$

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 resets and new time-step can be computed.

NUMERICAL EXAMPLE: 2D ALIGNED VIBRATION

To evaluate the spatial convergence rate of the proposed algorithm, we consider a 2D aligned vibration problemfrom Sadeghirad et al. (2011) that is constructed using the method of manufactured solutions. The displacement field is assumed as:

$$\mathbf{u}(X,Y,t) = \begin{bmatrix} A\sin\left(2\frac{\pi X}{L}\right)\sin\left(\frac{c\pi t}{L}\right) \\ A\sin\left(2\frac{\pi Y}{L}\right)\sin\left(\frac{c\pi t}{L}+\pi\right) \end{bmatrix} \tag{38}$$

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 ($c = \frac{\sqrt{E}}{\gamma}$). The velocity vector is computed by differentiating the displacement by time as follows:

$$\mathbf{v}(X,Y,t) = \frac{d\mathbf{u}}{dt} = \begin{bmatrix} Ac\pi\sin\left(2\frac{\pi X}{L}\right)\cos\left(\frac{c\pi t}{L}\right) \\ Ac\pi\sin\left(2\frac{\pi Y}{L}\right)\cos\left(\frac{c\pi t}{L}+\pi\right) \end{bmatrix} \tag{39}$$

Then the1st Piola-Kichhoff stress **P**, with respect to the reference configuration and the Cauchy stress$\boldsymbol{\sigma}$, with respect to the deformed configuration for the Neo-Hookean material can be written as:

$$\mathbf{P} = \lambda \ln(J)\mathbf{F}^{-1} + \mu\mathbf{F}^{-1}(\mathbf{F}\mathbf{F}^T - \mathbf{I}) \tag{40}$$

$$\boldsymbol{\sigma} = \frac{\mathbf{P}\mathbf{F}^T}{J} = \frac{\lambda \ln(J)}{J}\mathbf{I} + \frac{\mu}{J}(\mathbf{F}\mathbf{F}^T - \mathbf{I}) \tag{41}$$

where J is the determinant of the deformation gradient **F**, whileμ and λ are the shear modulus and Lame constant, respectively. To manufacture the solution, the body forces are used as the source term to construct the given displacement field as follows:

$$\mathbf{b} = \begin{bmatrix} \frac{\pi u_x}{L^2}\left(4\frac{\mu}{\rho_o} - c^2 - 4\frac{\lambda[\ln(F_{xx}.F_{yy})-1]-\mu}{\rho_o F_{xx}^2}\right) \\ \frac{\pi u_y}{L^2}\left(4\frac{\mu}{\rho_o} - c^2 - 4\frac{\lambda[\ln(F_{xx}.F_{yy})-1]-\mu}{\rho_o F_{yy}^2}\right) \end{bmatrix} \tag{42}$$

In this paper, the numerical parameters include Young modulus E = 10e^{7}Pa, Poisson's ratio υ=0.3, initial density ρ_o= 1000 kg/m3. The duration of the simulation is T=0.2s with the time step dt = 1e^{-5}satisfying the Courant–Friedrichs–Lewy condition. The square domain has a size of 1 x 1m. The spatial convergence rate is tested with the number of the grid cell of 8, 16, 32, 64 in both x and y direction, which corresponds to the grid cell size of 0.125m, 0.0625m, 0.03125m and 0.015625m respectively. In each cell,9 material points are distributed equally in a single cell. Each cell contains 4 Gauss quadrature points for Gauss integration. To generate the large deformation, the displacement amplitude A is 0.1m. The root-mean-square errors (RMS) for numerical simulations are:

$$\mathrm{RMS} = \sqrt{\frac{\sum_{p=1}^{N_p} \left\| \mathbf{f}_{\mathrm{numerical}}(\mathbf{x}_p) - \mathbf{f}_{\mathrm{exact}}(\mathbf{x}_p) \right\|^2}{N_p}} \tag{43}$$

Here, the errors of the displacement, velocity and stress of material points are calculated at the end of the numerical simulation. Figure 2, Figure 3 and Figure 4 show the spatial convergence rate for the quantities of interest respectively for MPM, B-spline MPM, and B-spline MPM combined with moving least square reconstruction technique. The results illustrate that MPM does not converge well with the increasingly refined grid, most probably due to the cell-crossing errors. While the errors made by B-spline MPM are considerably lower than those generated by the original MPM, the method also shows a lack of convergence for finer meshes. The moving least square technique significantly improves the accuracy of the B-spline MPM in the large deformation regime.

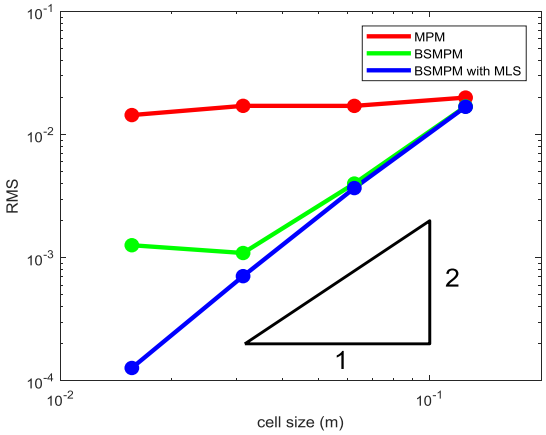

Figure 2 Spatial convergence rate of displacement (A = 0.1m)

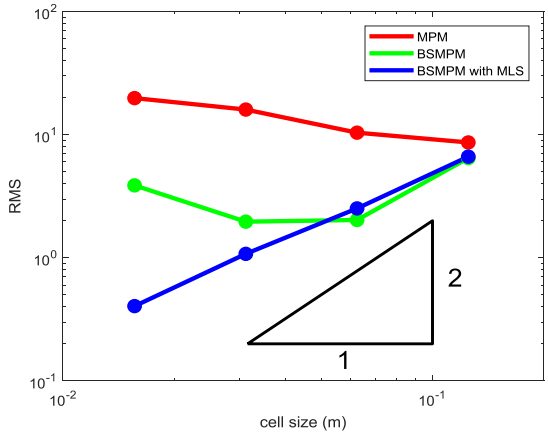

Figure 3 Spatial convergence rate of velocity (A = 0.1m)

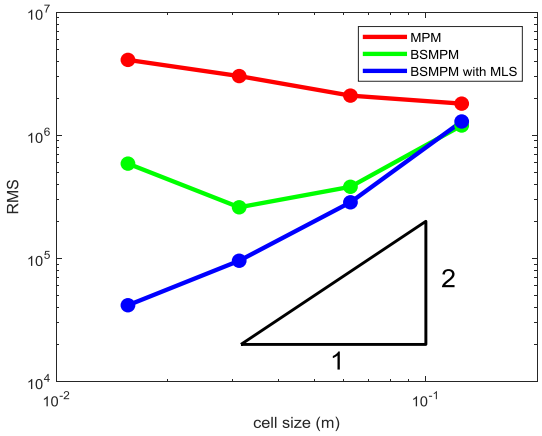

Figure 4 Spatial convergence rate of stress (A = 0.1m)

CONCLUSIONS

In this paper, a moving least squares function reconstruction technique is combined with B-spline MPM. The functions of the density, the momentum, the body forces, and the stress at material points are reconstructed using MLS.

Then, the obtained functions are evaluated at 4 Gauss points per cell to integrate the function using B-spline basis functions. The proposed algorithm can improve the accuracy of B-spline MPM at finer meshes. However, the proposed formulation uses the lumped mass matrix and a low-order time integration. Those can still lead to a lower order of accuracy. Additionally, the function reconstruction is sensitive to the number of material points per cell and the reconstruction may become inaccurate when there are not enough material points to reconstruct the high-order functions in the background grid. These shortcomings of the proposed formulation and the sensitivity of number of material points per cell require further research, so the method can achieve the optimal convergence rate in large deformation problems.

ACKNOWLEDGEMENTS

The work has been funded by the Academy of Finland under the project 'Progressive failure and post-failure modelling of slopes with Generalized Interpolation Material Point Method (GIMP)' under decision number 286628. The authors would like to thank Roel Tielen for his implementation of B-spline MPM.

REFERENCES

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

Boor, de, C. (1971). Subroutine package for calculating with B-splines. *Techn. Rep.* LA-4728-MS, Los Alamos Sci. Lab, Los Alamos, NM

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.

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. (2008). Examination and Analysis of Implementation Choices within the Material Point Method (MPM). *Computer Modellin in Engineering and Sciences*, 107-127.

Sulsky, & Gong. (2016). Improving Material Point Method. In K. Weinberg, & A. Pandolfi, *Innovative Numerical Approaches for Multi-Field and Multi-Scale Problems* (Vols. Lecture Notes in Applied and Computational Mechanics, vol 81). Springer.

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. (2009). A weighted least squares particle.in.cell method for solid mechanics. *International journal for numerical methods in engineering*, 1-6.

Wobbes, E., Moller, M., Galavi, V., & Vuik, C. (2018). Conservative Taylor Least Squares reconstruction with application to material point methods. *International Journal for Numerical Methods in Engineering*.

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