

Unidimensional Axial Vibration Using MPM

Adriano da Silva Rocha^a

^a*Laboratory of Scientific Computation and Visualization, Av. Lourival Melo Mota,
S/N, Maceió, 57072-900, Alagoas, Brazil*

Abstract

The material point method (MPM) is a numerical simulation method that combines features of classical numerical methods to achieve greater applicability and efficiency. This work aimed to apply the MPM for didactic purposes to a one-dimensional system composed of a continuous bar under free axial vibration. For this, initial conditions capable of excite isolated vibration modes were considered, and for the MPM simulation, the USF (update stress first) algorithm was used. After numerical simulation for three vibration modes (1st, 5th, 10th) the center of mass velocity values were compared with analytical results. The results of the first mode of vibration were the same as the analytical solution, while the curves for the following modes showed similar amplitudes, but also an increasing phase difference according to the order of oscillation, being 5% for the fifth mode and 20% to the tenth mode, as found in the literature. Complementary simulations with adjustment in the spatial discretization reduced the phase difference to about 1% for the fifth mode and 5% for the tenth. Thus, it is observed that the application of the method was satisfactory and that its precision is dependent on the parameters used for simulation.

Keywords: Material Point Method, Axial Vibration, USF

1. Introduction

The material point method (MPM) is a numerical simulation method for continuum mechanics problems that was developed to overcome the challenges encountered by other well-established methods such as Lagrangian and Eulerian methods. While Lagrangian methods have advantages like absence of advection between their mesh elements – leading to simpler and

more efficient governing equations – and that they can easily identify material interfaces at the boundaries of the mesh elements, these methods also face considerable disadvantages in problems whose meshes undergo large deformations and become entangled, leading to large numerical errors [1]. Contrastingly, Eulerian methods have the advantage that their mesh does not undergo deformation, avoiding entanglement and making the methods suitable for problems involving large deformations. However, these methods deal with advection of physical variables such as mass and energy between the elements of their mesh, making their governing equations more complex and their numerical solutions more time consuming. In addition, Eulerian methods cannot accurately define material interfaces [1, 2]. MPM manages to combine the advantages of both methods, using a background mesh, which does not undergo permanent deformation, and particles that represent the materials. Thus, the MPM does not deal with advection of physical variables between the elements of its mesh, simplifying its governing equations; its mesh does not suffer entanglement, making it suitable for use in problems with large deformations and its particles successfully determine the material interfaces [2].

The axial vibration of a continuous bar is a didactically interesting mechanical problem for the validation of numerical simulations, since, depending on the boundary conditions, the oscillation occurs in specific patterns, or modes of vibration. As the problem already has a widespread analytical solution, it is possible to compare the solutions obtained by MPM simulation to analytical results, analyzing the simulation performance for different vibrational modes, as done in [3].

2. Metodology

2.1. *The Material Point Method*

The material point method is a numerical simulation method that makes use of a background mesh, which covers the entire problem domain, and particles, also called material points, which represent the continuous body and record all the constitutive parameters such as velocity, mass, tension, deformation, among others. Similar to the finite element method (FEM), MPM uses a mapping function, or shape function, to pass important material body data to mesh element nodes. This transfer of information is important so that the momentum equilibrium equation, discretized in time, can be solved

in the mesh nodes, while the mass conservation equations and constitutive equations are solved in the particles [2].

The MPM has its formulation based on the momentum conservation equation. However, to enable a numerical solution, its weak form is usually used according to Eq. (1):

$$\int_{\Omega} \rho \ddot{\mathbf{u}} \delta \mathbf{u} dV + \int_{\Omega} \boldsymbol{\sigma} \nabla \delta \mathbf{u} dV - \int_{\Omega} \rho \mathbf{b} \delta \mathbf{u} dV - \int_{\Gamma} \bar{\mathbf{t}} \delta \mathbf{u} dA = 0 \quad (1)$$

Where Ω represents the entire domain of the continuum body, ρ is the density, $\ddot{\mathbf{u}}$ is the acceleration, $\delta \mathbf{u}$ is the virtual velocity vector, or test function, $\boldsymbol{\sigma}$ is the tension, \mathbf{b} is the body force per unit of mass acting in the continuum, $\bar{\mathbf{t}}$ is the specific tension acting on the surface Γ .

MPM is a method with discretization in space, through the background mesh and particles and also in time, being solved using iterations or time steps. At each time step, the same algorithm is repeated, until the total simulation time is reached. The algorithms differ in relation to the moment in which the particle stress update is performed in the time step, which may occur before the momentum integration, in the case of the USF (Update Stress First algorithm), or after the moment integration, being called the USL (Update Stress Last algorithm). The difference between the algorithms is enough to generate results with different patterns, especially regarding energy conservation [3]. In this work, only the USF algorithm was used and detailed below following [1].

2.1.1. Algorithm

In the first step, the mass and momentum of the particles are mapped to the nodes of the background mesh through the shape function, according to Eqs. (2) and (3).

$$m_n^t = \sum_{p=1}^{n_p} m_p N_{np}^t \quad (2)$$

$$\mathbf{p}_n^t = \sum_{p=1}^{n_p} m_p \mathbf{v}_p^t N_{np}^t. \quad (3)$$

Where the index t represents the time step, the index n identifies which mesh node the associated property belongs to, in the same way as the index p identifies the particle to which the associated property belongs. Thus, m_n^t means the mass of node n in time step t, likewise m_p means the mass of

particle p ; with the values of n and p varying between 1 and the number of nodes and particles, respectively; while n_p in the sum, represents the number of particles close enough to influence the node which is being analyzed. The term N_{np}^t represents the mapping function, or shape function. More details on the shape function are given in the following subsection. Finally, the term \mathbf{p}_n^t represents the momentum of node n at time step t and the term \mathbf{v}_p^t represents the velocity of particle p at time step t . The second step is to impose boundary conditions on the fixed nodes, changing velocity and momentum of these nodes to zero. The third step in the USF algorithm is to calculate the strain and vorticity increments of the particles based on the nodal velocities and, based on the strain increment, calculate the stress increment and the density of the next time step (which also represents the variation volume over time). For this, the nodal velocities are calculated according to Eq. (4), while the deformation and vorticity increments are calculated according to Eqs. (5) and (6), respectively, and the density according to Eq. (7).

$$v_n^t = \frac{p_n^t}{m_n^t} \quad (4)$$

$$\Delta \epsilon_p^t = \frac{1}{2} \sum_{n=1}^{n_n} \left(\frac{\partial N_{np}^t}{\partial x} \mathbf{v}_n^t + \frac{\partial N_{np}^t}{\partial y} \mathbf{v}_n^t \right) \Delta t \quad (5)$$

$$\Delta \Omega_p^t = \frac{1}{2} \sum_{n=1}^{n_n} \left(\frac{\partial N_{np}^t}{\partial x} \mathbf{v}_n^t - \frac{\partial N_{np}^t}{\partial y} \mathbf{v}_n^t \right) \Delta t \quad (6)$$

Where Δt in equations 5 and 6 represents the time increment (the difference between each time step) and n_n represents the number of nodes influenced by the particle which is being analyzed..

$$\rho_p^{t+1} = \rho_p^t / (1 + \Delta \epsilon_{ip}^t) \quad (7)$$

The stress increment calculation is defined through the constitutive model used. In this paper a hypoelastic model was used, so that the stress increment could be calculated through Eq. (8) and the particle stress state can be updated through Eq. (9).

$$\Delta \boldsymbol{\sigma}_p^t = E \Delta \boldsymbol{\epsilon}_p^t \quad (8)$$

$$\boldsymbol{\sigma}_p^{t+1} = \boldsymbol{\sigma}_p^t + \Delta \boldsymbol{\sigma}_p^t \quad (9)$$

In the next step, the internal, external and total nodal forces are calculated for each node, following Eqs. (10), (11) and (12), respectively. In this step, it is also necessary to impose boundary conditions at fixed nodes, which means null total force at those nodes.

$$\mathbf{f}_{int_n}^t = - \sum_{p=1}^{n_p} \nabla N_{np}^t \boldsymbol{\sigma}_p \frac{m_p}{\rho_p} \quad (10)$$

$$\mathbf{f}_{ext_n}^t = \sum_{p=1}^{n_p} m_p N_{np}^t \mathbf{b}_p^t + \sum_{p=1}^{n_p} N_{np}^t \bar{\mathbf{t}}_p^t h^{-1} \frac{m_p}{\rho_p} \quad (11)$$

$$\mathbf{f}_n^t = \mathbf{f}_{int_n}^t + \mathbf{f}_{ext_n}^t \quad (12)$$

Where $\bar{\mathbf{t}}$ is the specific traction at time t and h is the thickness of the layer under the surface undergoing the traction. Once the nodal forces are known, it is possible to integrate the nodal momentum in time according to Eq. (13) and get to know the next time step nodal momentum.

$$\mathbf{p}_n^{t+1} = \mathbf{p}_n^t + \mathbf{f}_n^t \Delta t \quad (13)$$

Using the next time step nodal momentum and the present time step nodal acceleration and velocity it is possible to calculate the next time step particle velocity and position according to Eqs. (14) and (15) respectively.

$$\mathbf{v}_p^{t+1} = \mathbf{v}_n^t + \sum_{n=1}^{n_n} \frac{\mathbf{f}_n^t N_{np}^t}{m_n^t} \Delta t \quad (14)$$

$$\mathbf{x}_p^{t+1} = \mathbf{x}_n^t + \sum_{n=1}^{n_n} \frac{\mathbf{p}_n^{t+1} N_{np}^t}{m_n^t} \Delta t \quad (15)$$

Once particles velocity and position are updated, the time step ends and the information assigned to the nodes is discarded. Then, in case total simulation time has been reached the simulation is finished, case not, a new time step starts, restarting this algorithm.

2.1.2. Mapping Function

According to [4], the mapping function is an interpolation to transfer the properties of particles to the nodes that surround them. Interpolation works in such a way that the closer the particle is to a node, the greater

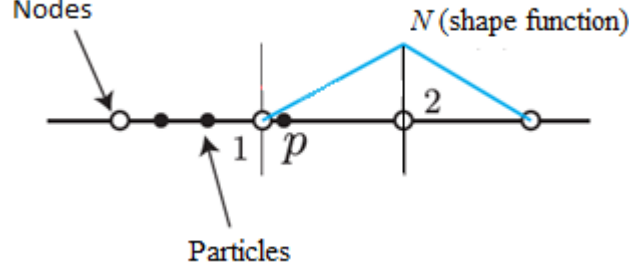


Figure 1: Hat function on 1-D. Edited from [4].

its influence on it and the smaller its participation in the composition of the properties of the other nodes. One of the most used shape functions in MPM is a composite and Y-symmetric function, which has a hat shape [4]. The hat function is composed of linear functions that have a maximum value equal to one at the position of the analysis node and decreases in the direction of the neighboring nodes, where they have a minimum value of zero. In this way, the hat function makes it explicit that particles with a distance greater than or equal to the length of a mesh element do not influence the reference node, making it possible to know how many nodes will be influenced (at most) by each particle based on the dimensions of the problem: in one-dimensional problems, two nodes are influenced, in two-dimensional ones, 4 nodes and in three-dimensional ones, 8 nodes). As this work addresses a one-dimensional problem, the hat function used is composed of only two linear functions, as in Figure 1.

2.1.3. Axial Vibration of a Continuum Bar

According to [3], the axial vibration of a bar occurs in modes of vibration, which one with different frequency and amplitude of oscillation. Generally, several vibration modes are excited at the same time, however, with specific initial conditions, it is possible to excite a single mode of vibration. The vibration modes are shown in Eq. (16).

$$\phi_n(x) = \sin \beta_n x \quad (16)$$

Where β_n are the eigenvalues defined according to Eq. (17), where n varies from 1 to infinity and L corresponds to the length of the bar measured

in mesh elements.

$$\beta_n = \frac{2n-1}{2} \frac{\pi}{L} \quad (17)$$

The oscillation frequencies, in turn, are defined based on the eigenvalues β_n according to Eq. (18).

$$\omega_n = \beta_n c \quad (18)$$

Where c represents the speed of sound in the medium, which can be calculated through the square root of the ratio between the modulus of elasticity of the medium and its density.

Using initial conditions multiples of the systems natural frequency of vibration, it is possible to separately excite the chosen mode of vibration chosen as performed in [3] where the conditions used determined that, in the initial instant, the displacement is null in the whole bar and the velocity is defined by Eq. (19), considering a one-dimensional system.

$$\mathbf{v}(x, 0) = \mathbf{v}_0 \sin \beta_n x \quad (19)$$

Where \mathbf{v}_0 is the amplitude of the initial velocity. The problem has an analytical solution for the velocity in the entire domain of the bar, which can be used to develop an expression for the center of mass velocity depending on time Eq. (20).

$$\mathbf{v}_{cm} = \frac{\mathbf{v}_0}{\beta_n L} \cos(\omega_n t) \quad (20)$$

The problem proposed by [3] was then reproduced, where a one-dimensional bar fixed at $x = 0$ and free at $x = L$ vibrates under the aforementioned initial conditions and simulations were performed for three vibration modes ($n = 1, 5$ and 10). The MPM simulation results for the velocity at the center of mass were then compared to the analytical results from Eq. (20), as performed in [3], using the same input data in the simulation, namely: length of the one-dimensional bar equals to 25, with length of mesh elements (Δx) equals to 1, quantity of particles equals to 50, being two per element, modulus of elasticity (E) equals to 100, density (ρ) equals to 1, initial velocity amplitude (v_0) equals to 0.1 and a time increment defined by Eq. (21). The values can be used for any system of measurement.

$$\Delta t = 0.1 \left(\frac{\Delta x}{c} \right) \quad (21)$$

3. Results

Figure 2 shows the center of mass velocities according to the analytical result and the one simulated by MPM. The two curves overlap, showing great correspondence of the simulation, as in the work of [3]. However, when evaluating the results for the fifth mode of vibration, it is noticed that there is a loss of accuracy. Figure 3 shows the results for $n = 5$, where it is possible to observe that the curves, although having considerably similar amplitudes, are out of phase. This phase difference is due to the difference between the analytical and simulation frequencies of approximately 5% of the analytical value. The same behavior was observed in the work of [3]. Changes in the number of particles per element did not result in better convergence (with tests being performed for up to 6 particles per element, triple the initial configuration), however, adjustments in the spatial discretization were able to diminish the difference between the frequencies and approximate the results, so that the difference was reduced to approximately 1% by decreasing the length of the element used in the initial configurations by half. Following the same principle, the results for the tenth mode of vibration, shown in Figure 4, also showed curves with similar amplitudes, but with an even greater phase offset between them. The difference between the analytical and MPM frequencies reached something around 20%, being this result again consistent with what was found in [3]. Again, new simulations varying the number of particles (increasing up to 3 times the initial amount) and the temporal discretization did not result in significant differences, however, adjusting the spatial discretization to half the initial value of the mesh element length made it possible to reduce the difference between frequencies to 5%.

4. Conclusions

In this paper carried out an application, for didactic purposes, of the MPM method through the USF algorithm to a one-dimensional system composed of a continuum bar in free axial vibration, as proposed by [3]. The results obtained were similar to those found in the literature, with the simulated center of mass velocity being equal to the analytical one in the first mode of vibration and with the phase differences of the remaining modes also corresponding to those reported on [3], indicating that the method was correctly used. For the fifth and tenth modes of vibration, alternative simulations were performed varying the number of particles used, temporal discretization and

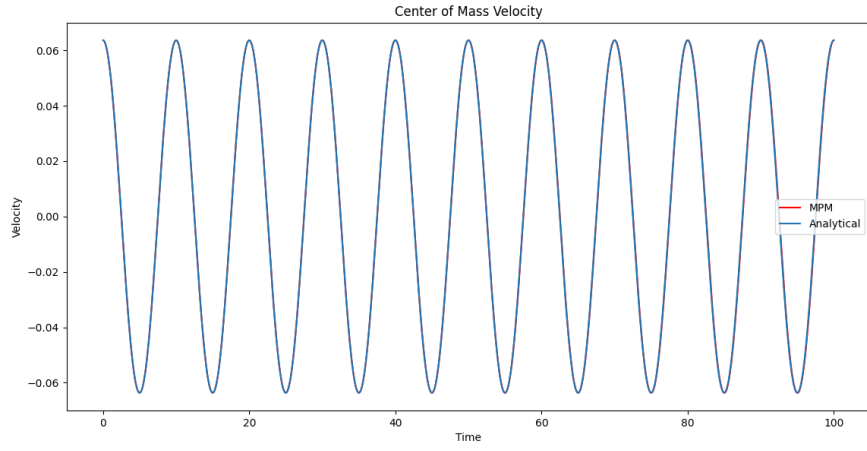


Figure 2: Mode of vibration 1 – Velocity of the center of mass.

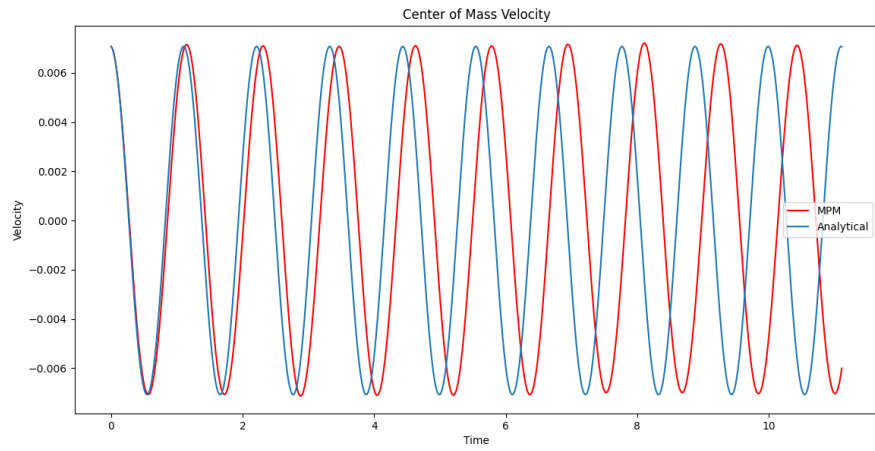


Figure 3: Mode of vibration 5 – Velocity of the center of mass.

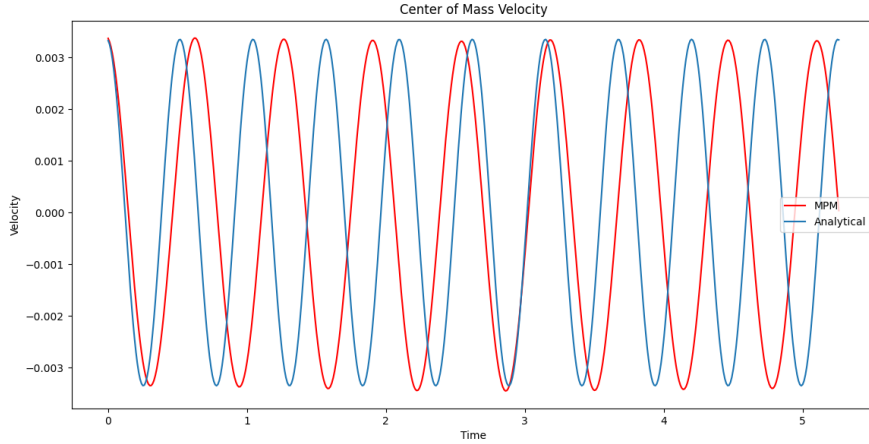


Figure 4: Mode of vibration 10 – Velocity of the center of mass.

spatial discretization, the latter being able to reduce the difference between the simulated and analytical frequencies (from 5% to 1% in the fifth vibrational pattern; and from 20% to 5% in the tenth vibrational mode). From this it is possible to conclude that, despite the results in phase dislocation for $n = [5, 10]$, there is no inadequacy of the numerical method to represent higher order vibration modes, with only the inadequacy of the simulation parameters used, being possible to obtain closer results by adjusting these parameters according to the characteristics of the problem.

References

- [1] X. Zhang, Z. Chen, L. Y, The material point method: A continuum-based particle method for extreme loading cases, Elsevier 1 (2017).
- [2] J. Fern, A. Rohe, K. Soga, E. Alonso, The material point method for geotechnical engineering: A practical guide, CRC Press (2019).
- [3] S. G. Bardenhagen, Energy conservation error in the material point method for solid mechanics, Journal of Computational Physics 180 (2002) 383–403.
- [4] V. P. Nguyen, Material point method: Basics and applications, Technical report. Australia: Monash University (2014).