Material-Point Method Analysis of Bending in Elastic Beams

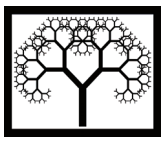
Andersen, Søren Mikkel; Andersen, Lars Vabbersgaard

Published in:
Proceedings of the Eleventh International Conference on Civil, Structural and Environmetal Engineering Computing

Publication date:
2007

Document Version
Publisher's PDF, also known as Version of record

Paper 133



Proceedings of the Eleventh International Conference
on Civil, Structural and Environmental
Engineering Computing,
B.H.V. Topping (Editor),
Civil-Comp Press, Stirlingshire, Scotland

Material-Point Method Analysis of Bending in Elastic Beams

S. Andersen and L. Andersen
Department of Civil Engineering
Aalborg University, Denmark

Abstract

The aim of this paper is to test different types of spatial interpolation for the material-point method. The interpolations include quadratic elements and cubic splines. A brief introduction to the material-point method is given. Simple liner-elastic problems are tested, including the classical cantilevered beam problem. As shown in the paper, the use of negative shape functions is not consistent with the material-point method in its current form, necessitating other types of interpolation such as cubic splines in order to obtain smoother representations of field quantities. It is shown that the smoother field representation using the cubic splines yields a physically more realistic behaviour for impact problems than the traditional linear interpolation.

Keywords: MPM, spatial interpolation, cubic splines, beam bending, impact.

1 Introduction

Over the last decades the material-point method (MPM) has emerged as a serious tool for numerical modelling of continuum problems involving large deformations and contact between multiple bodies. The method in its current form has been developed by Sulsky et al. [1, 2]. The method originates from the particle-in-cell method, developed by Harlow [3]. Some of the early applications of the material-point method include impact problems [4] and simulations of membranes [5, 6].

In the MPM, two descriptions are used. Material points following the deformation of the material are utilised to track the position and the state variables while a computational background grid is employed to solve the governing equations. As the material points carry all the information about the material, total freedom exists for choosing a mesh. Hence, the material-point method avoids some drawbacks associated with the traditional methods adopting a purely Lagrangian or Eulerian description.

Recently, research has focused on the accuracy and theoretical foundation of the method. In [7] the energy conservation in the material-point method is analysed. In the work of Guilkey and Weiss [8], implicit time integration is presented, and an alternative implicit scheme is proposed by Sulsky and Kaul [9]. Another way to obtain a smoother field representation is given by Bardenhagen [10]. Here the field quantities are smoothed by combining shape functions associated with the computational mesh and characteristic functions defined for each of the material points. In the work by Love and Sulsky [11], a formulation of the material-point method is proposed using hyperelastic materials and nonlinear elastodynamics that is shown to preserve conservative field properties.

An issue that deserves attention is the choice of computational mesh and spatial discretization order, which is analysed in this paper. The traditional material-point method approach has been to use a regular mesh consisting of rectangular or triangular first-order elements. As known from the finite-element method, the use of low-order elements can cause locking in problems where material bending is dominant. The use of second-order elements (or computation cells) and cubic splines in the material-point method is tested against first-order elements for simple elastic bending problems and problems involving impact. However, the use of second-order finite elements implies the use of shape functions that have the possibility to become negative. As shown in the appendix, negative mass associated with grid nodes is not consistent with the MPM in its current form.

The scope of this paper is as follows. Firstly, an outline of the governing equations is given in Section 2. Secondly, the material-point method is presented in Section 3, with particular focus on the four different spatial discretizations. Finally, numerical examples are given in Section 4.

2 Governing equations

A linear-elastic continuum is considered. A steel beam can with reasonable accuracy be described as such a material, provided that the deformations remain small. In this paper, only two-dimensional problems are analysed and the initial computational domain is denoted $\Omega_0 \in \mathbf{R}^2$. At the time t, the domain is denoted $\Omega(t)$. The boundary $\partial\Omega$ is divided into two disjoint sets: $\partial\Omega_u$ with prescribed displacements and $\partial\Omega_\tau$ with known traction.

The accuracy of a numerical model for an elastic continuum depends on its ability to obey the three governing equations, i.e. balance of momentum, mass conservation, and conservation of mechanical energy. Firstly, conservation of mass implies that

$$\frac{d\rho}{dt} + \rho\nabla \cdot \mathbf{v} = 0, \tag{1}$$

where $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ is the spatial velocity and $\rho = \rho(\mathbf{x}, t)$ is the current density. Further, ∇ is the gradient operator and $\nabla \cdot \mathbf{a}$ is the divergence of the vector field $\mathbf{a}$. Secondly,

conservation of momentum involves that

$$\rho \frac{d\mathbf{v}}{dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b}, \tag{2}$$

where $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{x}, t)$ is the Cauchy stress tensor and $\mathbf{b} = \mathbf{b}(\mathbf{x}, t)$ is the specific body force. Finally, mechanical energy conservation is ensured by the equation

$$\rho \frac{dE}{dt} = \boldsymbol{\sigma} : \frac{d\boldsymbol{\varepsilon}}{dt} + \rho \mathbf{v} \cdot \mathbf{b}, \tag{3}$$

where E is the internal energy per unit mass in the current configuration, $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\mathbf{x}, t)$ is the strain and $d\boldsymbol{\varepsilon}/dt$ the corresponding strain rate. Here $\mathbf{A} : \mathbf{B} = a_{ij} b_{ij}$ in Cartesian coordinates. The total derivative d/dt is given as

$$\frac{d\mathbf{c}}{dt} = \left.\frac{\partial \mathbf{c}}{\partial t}\right|_{\mathbf{x}} + \mathbf{v} \cdot \nabla \mathbf{c}, \tag{4}$$

where $\mathbf{c}$ is the considered field quantity and $\mathbf{v}$ is the velocity of the material relatively to the coordinate system. When material points that follow the deformation of the material are utilised, the relative velocity is zero and the convective term does not appear. Furthermore, the balance of mass is implicitly assured.

In order to complete the description of the continuum, a constitutive law relating the strain rates and the stress rates of the material is needed. The strain rate can be determined from the equation

$$\frac{d\boldsymbol{\varepsilon}}{dt} = \frac{1}{2} \left(\nabla \mathbf{v} + (\nabla \mathbf{v})^T \right), \tag{5}$$

where superscript T denotes the transpose. The stress depends on the type of material under consideration. A general linear-elastic constitutive model can be expressed as

$$\frac{d\boldsymbol{\sigma}}{dt} = \mathbf{D} : \frac{d\boldsymbol{\varepsilon}}{dt}, \tag{6}$$

where $\mathbf{D}$ is the fourth-order elasticity tensor.

3 The material-point method

3.1 Discretization

To formulate the material-point method (MPM) form of the governing equations, Eqs. (1) to (3) must be given in a discrete form. The domain Ω_0 is divided into N_p sub-domains Ω_p and all the mass is ascribed to material points located in the centroid of each domain. This provides the density field

$$\rho(\mathbf{x}, t) = \sum_{p=1}^{N_p} m_p \delta(\mathbf{x} - \mathbf{x}_p), \tag{7}$$

where m_p is the mass associated with the material point at the spatial coordinates $\mathbf{x}_p$. The values of m_p are based on the initial configuration and they do not change during the computation; hence, the total mass is conserved.

In order to obtain the material gradients, a computational grid is formed. Thus, the domain Ω is divided into N_e finite elements. In the numerical examples, different kinds of discretization are applied, including first-order four-node rectangular elements, second-order eight- and nine-node rectangular elements and cubic splines. This defines a number of computational nodes, N_n, with standard nodal basis functions $\Phi_i(\mathbf{x})$ associated with node i. By letting the nodal basis functions describe the spatial variation, the velocity and the acceleration fields are represented as

$$\mathbf{v}(\mathbf{x},t) = \sum_{i=1}^{N_n} \Phi_i(\mathbf{x})\mathbf{v}_i(t), \quad \frac{d\mathbf{v}}{dt} = \sum_{i=1}^{N_n} \Phi_i(\mathbf{x})\frac{d\mathbf{v}_i}{dt}, \tag{8}$$

respectively, where $\mathbf{v}_i(t)$ are the nodal values of the velocities at the time t. The equations are solved in an updated Lagrangian frame, ensuring that no time derivatives of the shape functions occur.

The weak form of the balance of momentum is obtained by multiplication of Eq. (2) by an arbitrary test function $\mathbf{w}(\mathbf{x},t)$ and integrating over the domain Ω. Standard nodal basis functions are also used to describe the test function $\mathbf{w}$, i.e.

$$\mathbf{w}(t) = \sum_{i=1}^{N_n} \Phi_i(\mathbf{x})\mathbf{w}_i(t), \tag{9}$$

where $\mathbf{w}_i(t)$ are the nodal values of the test function at the time t. Since $\mathbf{w}$ is arbitrary, except where displacements are prescribed, the balance of momentum reduces to

$$\sum_{j=1}^{N_n} m_{ij}\frac{d\mathbf{v}_j}{dt} = \mathbf{f}_i^{int} + \mathbf{f}_i^{ext}. \tag{10}$$

Here $m_{ij} = m_{ij}(t)$ are the components of a consistent mass matrix formed at the time t, whereas the internal and external forces are defined as

$$\mathbf{f}_i^{int} = -\sum_{p=1}^{N_p} \frac{m_p}{\rho_p}\boldsymbol{\sigma}_p \cdot \mathbf{G}_{ip}, \quad \mathbf{f}_i^{ext} = \boldsymbol{\tau}_i + \mathbf{b}_i, \tag{11}$$

respectively, where $\rho_p = \rho_p(t)$ and $\boldsymbol{\sigma}_p = \boldsymbol{\sigma}_p(t)$ are the mass density and Cauchy stresses at the material point p and the time t, respectively. Further,

$$\mathbf{G}_{ip} = \nabla\Phi_i(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_p} \tag{12}$$

is the gradient of the nodal basis functions and the specific body force is given as

$$\mathbf{b}_i = \sum_{p=1}^{N_p} m_p\Phi_i(\mathbf{x}_p)\mathbf{b}_p(t), \tag{13}$$

where $\mathbf{b}_p(t) = \mathbf{b}(\mathbf{x}_p, t)$ is the external acceleration associated with material point p. The applied traction is found by integrating over the boundary of the domain:

$$\boldsymbol{\tau}_i = \int_{\partial\Omega_\tau} \Phi_i(\mathbf{x})\boldsymbol{\tau}(\mathbf{x}, t)d\mathbf{x}. \tag{14}$$

In most of the numerical examples presented, lumped nodal masses are utilised in Eq. (10). The lumped mass associated with node i is expressed as

$$m_i(t) = \sum_{p=1}^{N_p} m_p \Phi_i(\mathbf{x}_p(t)). \tag{15}$$

Thereby, using lumped masses, the balance of momentum can be defined as

$$m_i \frac{d\mathbf{v}_i}{dt} = \mathbf{f}_i^{int} + \mathbf{f}_i^{ext}. \tag{16}$$

3.2 The MPM-algorithm

The general algorithm of the material point method can be given as follows. A problem is formulated with a set of material points that are given all the relevant material properties. Firstly including any initial conditions, the boundary conditions are imposed on the underlying grid and initial conditions are imposed at the material points. Then the time integration algorithm is started. For all time steps,

1. a computational mesh is generated,
2. the information is transferred from the material points to the grid nodes,
3. the equation of momentum is solved at the grid nodes,
4. information is transferred back to the material points,
5. the state variables are updated.

Different implementations are possible. The algorithm used in this paper is given as:

1. Initialisation of state variables at the material points at the time $t^k = 0$.
2. At each time step: Generation of a computational background mesh.
3. Nodal velocities are calculated by solving the system

$$\sum_{j=1}^{N_n} m_{ij}^k \mathbf{v}_j^k = \sum_{p=1}^{N_p} m_p \mathbf{v}_p^k \Phi_i(\mathbf{x}_p), \tag{17}$$

where $\sum_{j=1}^{N_n} m_{ij}^k \mathbf{v}_j^k = \mathbf{p}_i^k$ is identified as the nodal momentum.

4. The strain increments $\Delta\boldsymbol{\varepsilon}_p$ at material point p and the time t^k are found by
$$\Delta\boldsymbol{\varepsilon}_p^k = \frac{\Delta t^k}{2}\sum_{i=1}^{N_n}\left(\mathbf{G}_{ip}^k\mathbf{v}_i^k + (\mathbf{G}_{ip}^k\mathbf{v}_i^k)^T\right). \tag{18}$$
5. The strains are updated at the material points, i.e. $\boldsymbol{\varepsilon}_p^{k+1} = \boldsymbol{\varepsilon}_p^k + \Delta\boldsymbol{\varepsilon}_p^k$.
6. The stresses at the material points are found according to a chosen strain-driven constitutive model and based on the updated strain $\boldsymbol{\varepsilon}_p^{k+1}$.
7. The vectors of internal and external nodal forces are found by Eq. (11).
8. Updated velocities at the material points are found using internal forces:
$$\mathbf{v}_p^{k+1} = \mathbf{v}_p^k + \Delta t^k\sum_{i=1}^{N_n}\sum_{j=1}^{N_n} m_{ij}^{-1}\Phi_j(\mathbf{x}_p^k)\mathbf{f}_j^{int,k}. \tag{19}$$
9. The new position of the material points are found by
$$\mathbf{x}_p^{k+1} = \mathbf{x}_p^k + \Delta t^k\sum_{i=1}^{N_n}\sum_{j=1}^{N_n} m_{ij}^{-1}\Phi_j(\mathbf{x}_p^k)\mathbf{p}_j^k. \tag{20}$$
10. The material density is updated at the material points as
$$\rho_p^{k+1} = \rho_p^k \mathrm{e}^{-\nabla\cdot\mathbf{v}\Delta t}. \tag{21}$$
11. Finally, set $t^{k+1} = t^k + \Delta t^k$ and $k = k+1$ and repeat from item 1.

Typically, the time step is constant, i.e. $\Delta t^k = \Delta t =$ constant. The interpolation from the material points to the computational mesh defined by Eq. (17) is a weighted-least-squares approach [2]. Several choices have been made in order to obtain the above algorithm. For instance, for momentum-based updates, Eq. (19) has been made in order to reduce the numerical dissipation and increase the stability of the method.

Still a number of critical choices are to be made, regarding

- the size of the computational cells,
- the use of an adaptive grid or a uniform grid,
- the number of material points per cell,
- the application of implicit or explicit time integration,
- the use of a forward leap-frog algorithm or higher-order time integration,
- the use of consistent or lumped mass,
- the type of background mesh to use.

The first three items regarding the spatial discretization are problem specific and will not be addressed here in more detail. The two next issues relate to the accuracy of the method for a given computational cost. Here, an explicit formulation with forward leap-frog integration is used. The two last issues regarding the type of background mesh and the use of consistent or lumped masses are analysed in this paper.

3.3 Setting of the problem

As mentioned, the field variables are represented by the nodal basis functions. Thereby also the stress increments will vary according to the shape functions used. When linear shape-functions are applied, the gradient of the shape functions in either direction are constant within an element. Hence, when more material points are located within an element, stress-increments for all these material points are identical. With regard to beam problems in structural mechanics, the stress is known to vary according to Navier's equation:

$$\sigma_x(x, y) = \frac{M(x)}{I_x(x)} y, \tag{22}$$

where $\sigma_x(x)$ is the normal stress, $M(x)$ is the moment about the neutral axis, x is the coordinate along the neutral axis, y is the perpendicular distance to the neutral axis and I_x is the second moment of inertia about the neutral axis. The classical problem of a cantilevered beam with the length L and exposed to a single force, F, is illustrated in Figure 1. In this particular case, the moment is given by $M(x) = (x - L)F$. This suggests that using quadratic shape functions, where the stress increments are varying linearly within each element, is better suited for the solution of beam problems. Also it is known from finite-element theory that the use of first-order elements can cause locking for this kind of problem. In the following, the elements used for the numerical experiments are presented.

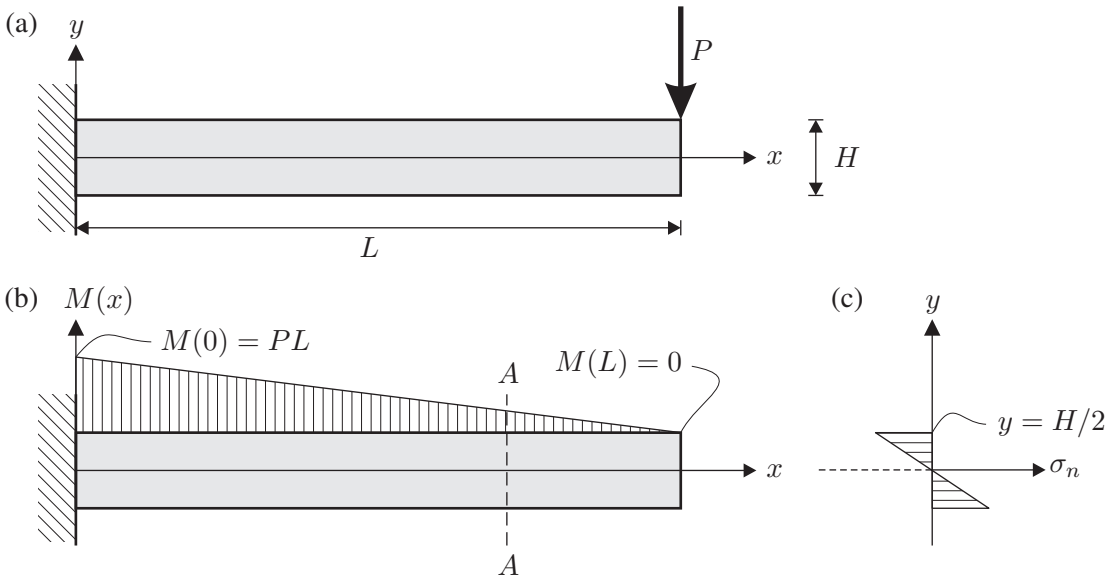


Figure 1: A classical problem in structural mechanics: (a) Geometry and boundary conditions, (b) distribution of the bending moment, $M(x)$, and (c) distribution of the normal stress, σ_n, at cross-section $A-A$.

3.4 Different interpolations used

The four types of elements utilised in the background mesh are shown in Figure 2. Standard normalised coordinates are used. For the linear elements as well as the nine-node quadratic elements and the eight-node serendipity element, ξ and η vary between -1 and 1, i.e. $\xi \in [-1, 1]$ and $\eta \in [-1, 1]$. Lagrangian interpolation functions are applied, defined by:

- Linear element with four nodes:

$$N_1(\xi,\eta) = \tfrac{1}{4}(\xi-1)(\eta-1), \quad N_2(\xi,\eta) = -\tfrac{1}{4}(\xi+1)(\eta-1),$$
$$N_3(\xi,\eta) = \tfrac{1}{4}(\xi+1)(\eta+1), \quad N_4(\xi,\eta) = -\tfrac{1}{4}(\xi-1)(\eta+1).$$

- Quadratic element with nine nodes:

$$N_1(\xi,\eta) = \tfrac{1}{4}(\xi^2-\xi)(\eta^2-\eta), \quad N_2(\xi,\eta) = \tfrac{1}{2}(1-\xi^2)(\eta^2-\eta),$$
$$N_3(\xi,\eta) = \tfrac{1}{4}(\xi^2+\xi)(\eta^2-\eta), \quad N_4(\xi,\eta) = \tfrac{1}{2}(\xi^2+\xi)(1-\eta^2),$$
$$N_5(\xi,\eta) = \tfrac{1}{4}(\xi^2+\xi)(\eta^2+\eta), \quad N_6(\xi,\eta) = \tfrac{1}{2}(1-\xi^2)(\eta^2+\eta),$$
$$N_7(\xi,\eta) = \tfrac{1}{4}(\xi^2-\xi)(\eta^2+\eta), \quad N_8(\xi,\eta) = \tfrac{1}{2}(\xi^2-\xi)(1-\eta^2),$$
$$N_9(\xi,\eta) = (1-\xi^2)(1-\eta^2).$$

- Serendipity element with eight nodes:

$$N_1(\xi,\eta) = \tfrac{1}{4}(1-\xi)(1-\eta)(-\xi-\eta-1), \quad N_2(\xi,\eta) = \tfrac{1}{2}(1-\xi^2)(1-\eta),$$
$$N_3(\xi,\eta) = \tfrac{1}{4}(1+\xi)(1-\eta)(\xi-\eta-1), \quad N_4(\xi,\eta) = \tfrac{1}{2}(1+\xi)(1-\eta^2),$$
$$N_5(\xi,\eta) = \tfrac{1}{4}(1+\xi)(1+\eta)(\xi+\eta-1), \quad N_6(\xi,\eta) = \tfrac{1}{2}(1-\xi^2)(1+\eta),$$
$$N_7(\xi,\eta) = \tfrac{1}{4}(1-\xi)(1+\eta)(-\xi+\eta-1), \quad N_8(\xi,\eta) = \tfrac{1}{2}(1-\xi)(1-\eta^2).$$

The last type of interpolation employs cubic splines that for the one-dimensional case are given by

$$S_3(\xi) = \begin{cases} 0 & \text{for} \quad |\xi| > 2, \\ \tfrac{1}{6}(2-|\xi|)^3 & \text{for} \quad 1 < |\xi| < 2, \\ \tfrac{2}{3} - |\xi|^2 + \tfrac{1}{2}|\xi|^3 & \text{for} \quad 0 < |\xi| < 1. \end{cases} \tag{23}$$

This cubic-spline interpolation is used for each individual material point and maps the material-point values to nodes a maximum distance of two grid nodes away in either direction. In two dimensions, this provides the 16-point stencil shown in Figure 2. Thereby the local coordinates are restrained by $\xi \in [0, 1]$ and $\eta \in [0, 1]$, and the shape functions become:

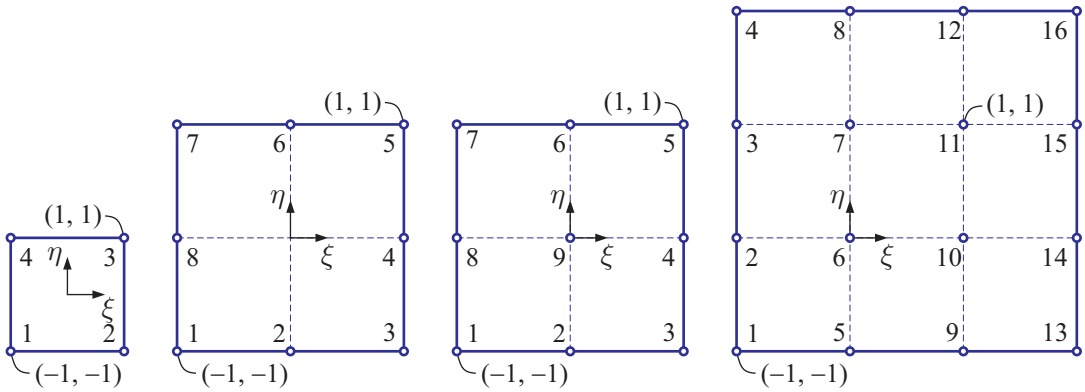


Figure 2: The different elements used: (a) Linear interpolation, (b) serendipity elements, (c) quadratic interpolation and (d) cubic splines.

$$
\begin{aligned}
&N_1(\xi,\eta) = \tfrac{1}{36}(1-\xi)^3(1-\eta)^3, \quad N_2(\xi,\eta) = \tfrac{1}{6}(1-\xi)^3(\tfrac{2}{3}-\eta^2+\tfrac{1}{2}\eta^3),\\
&N_3(\xi,\eta) = \tfrac{1}{6}(1-\xi)^3(\tfrac{1}{6}+\tfrac{1}{2}\eta+\tfrac{1}{2}\eta^2-\tfrac{1}{2}\eta^3), \quad N_4(\xi,\eta) = \tfrac{1}{36}(1-\xi)^3\eta^3,\\
&N_5(\xi,\eta) = (\tfrac{2}{3}-\xi^2+\tfrac{1}{2}\xi^3)\tfrac{1}{6}(1-\eta)^3, \quad N_6(\xi,\eta) = (\tfrac{2}{3}-\xi^2+\tfrac{1}{2}\xi^3)(\tfrac{2}{3}-\eta^2+\tfrac{1}{2}\eta^3),\\
&N_7(\xi,\eta) = (\tfrac{2}{3}-\xi^2+\tfrac{1}{2}\xi^3)(\tfrac{1}{6}+\tfrac{1}{2}\eta+\tfrac{1}{2}\eta^2-\tfrac{1}{2}\eta^3),\\
&N_8(\xi,\eta) = (\tfrac{2}{3}-\xi^2+\tfrac{1}{2}\xi^3)\tfrac{1}{6}\eta^3, \quad N_9(\xi,\eta) = (\tfrac{1}{6}+\tfrac{1}{2}\xi+\tfrac{1}{2}\xi^2-\tfrac{1}{2}\xi^3)\tfrac{1}{6}(1-\eta)^3,\\
&N_{10}(\xi,\eta) = (\tfrac{1}{6}+\tfrac{1}{2}\xi+\tfrac{1}{2}\xi^2-\tfrac{1}{2}\xi^3)(\tfrac{2}{3}-\eta^2+\tfrac{1}{2}\eta^3),\\
&N_{11}(\xi,\eta) = (\tfrac{1}{6}+\tfrac{1}{2}\xi+\tfrac{1}{2}\xi^2-\tfrac{1}{2}\xi^3)(\tfrac{1}{6}+\tfrac{1}{2}\eta+\tfrac{1}{2}\eta^2-\tfrac{1}{2}\eta^3),\\
&N_{12}(\xi,\eta) = (\tfrac{1}{6}+\tfrac{1}{2}\xi+\tfrac{1}{2}\xi^2-\tfrac{1}{2}\xi^3)\tfrac{1}{6}\eta^3,\\
&N_{13}(\xi,\eta) = \tfrac{1}{36}\xi^3(1-\eta)^3, \quad N_{14}(\xi,\eta) = \tfrac{1}{6}\xi^3(\tfrac{2}{3}-\eta^2+\tfrac{1}{2}\eta^3),\\
&N_{15}(\xi,\eta) = \tfrac{1}{6}\xi^3(\tfrac{1}{6}+\tfrac{1}{2}\eta+\tfrac{1}{2}\eta^2-\tfrac{1}{2}\eta^3), \quad N_{16}(\xi,\eta) = \tfrac{1}{36}\xi^3\eta^3.
\end{aligned}
$$

Since each material point interpolates to all grid nodes up to two cells away, it is not clear how to interpolate near the boundary. For the three other types of interpolation, a rigid boundary is enforced by setting the velocity of the grid nodes to zero at the boundary. Here a simple approach is made for the cubic splines. Rigid boundaries are imposed by having an extra column of grid nodes where the velocities are set to zero.

4 Numerical examples

In order to test the accuracy of the different types of interpolation, the free vibrations of a linear-elastic bar are analysed. The geometry of the problem is shown in Figure 3. In the present analysis, the length of the bar is $L = 5$ and the height is $h = 1$. Young's modulus is $E = 10^4$, while Poisson's ratio is $\nu = 0.3$ and the initial mass density is $\rho_0 = 10^3$. A discretization with 180 material points and 45 grid nodes is applied,

using 3×3 uniformly distributed material points in each computation cell. The total time of the simulation is $t_{tot} = 8$ and the fixed time step $\Delta t = 0.003$ is utilised. The bar is given a cosine-shaped initial velocity field

$$v(x) = A \cos\left(\frac{\pi x}{L}\right), \tag{24}$$

A small velocity amplitude of $A = 0.01$ is applied in order for the grid masses to remain positive when quadratic nine-node elements are used. With the other three types of elements, large amplitudes can be prescribed without such problems.

The velocity gradients stemming from the initial velocity field yield strains, and therefore stresses, thus leading to an interchange from kinematic into potential energy. Since no material dissipation is present in the model, undamped free vibrations should occur with no loss of mechanical energy. Therefore, the quality of the different types of interpolation may be tested by their ability to conserve mechanical energy.

The energy of the system is most naturally measured by the energy of the material points, as they carry all the information about the problem at all stages of the computational cycle. As only mechanical energy is considered, the total energy of the system is given by

$$E_{tot} = E_{kin} + E_{pot} \tag{25}$$

where E_{kin} denotes the kinetic energy and E_{pot} denotes the potential energy. The kinetic energy is defined as

$$E_{kin} = \frac{1}{2} \sum_{p=1}^{N_p} m_p \mathbf{v}_p \cdot \mathbf{v}_p \tag{26}$$

and, provided that no gravitational fields or other potential fields exist, the potential energy is solely the strain energy of the material points, i.e.

$$E_{pot} = \frac{1}{2} \sum_{p=1}^{N_p} \frac{m_p}{\rho_p} \boldsymbol{\sigma}_p : \boldsymbol{\varepsilon}_p \tag{27}$$

Here it is noted that the strain energy is mathematically well defined, as the calculation of stress increments according to Eq. (6) yields a hypo-elastic model. Hence, only the incremental form of the constitutive model is valid and the relation $\boldsymbol{\sigma} = \mathbf{D} : \boldsymbol{\varepsilon}$ is generally not valid. However for small deformations it is presumed that the expression for

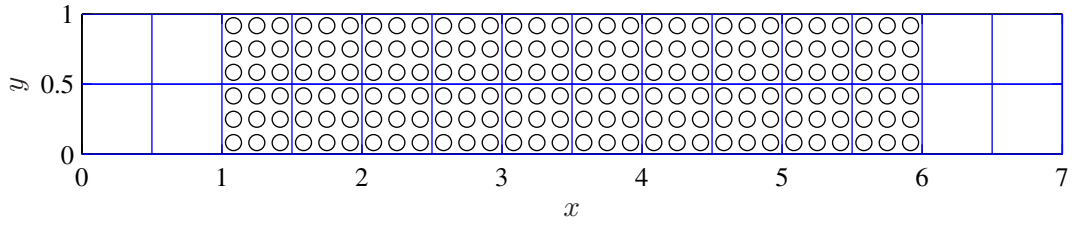


Figure 3: The discretization for the vibrating-bar problem

the potential energy is sufficiently accurate to provide information about the accuracy of the different elements.

When lumped masses are used, the kinetic energy of the nodes is defined as

$$E_{kin}^{nodes} = \frac{1}{2}\sum_{i=1}^{N_n} m_i \mathbf{v}_i \cdot \mathbf{v}_i. \tag{28}$$

When consistent mass is used, the nodal kinetic energy is given by

$$E_{kin}^{nodes} = \frac{1}{2}\sum_{j=1}^{N_n} m_{ij} \mathbf{v}_j \cdot \mathbf{v}_i. \tag{29}$$

For conservative systems, where the mechanical energy is to remain constant, a total error is defined as the change in total energy:

$$\mathcal{E}_{tot} = E_{tot} - E_{tot}^{init}. \tag{30}$$

Alternatively, the total error is defined as

$$\mathcal{E}_{tot} = \Delta E_{kin} + \Delta E_{pot}, \tag{31}$$

where ΔE_{kin} and ΔE_{pot} express the changes in kinetic and potential energy, respectively, between the initial and the current configuration of the system. In accordance with the definitions by Bardenhagen [7], this error is divided into an error due to interpolation of kinetic energy from the grid to the material points, $\mathcal{E}_{int}$, and an algorithmic error associated with everything else, $\mathcal{E}_{alg}$, i.e.

$$\mathcal{E}_{tot} = -\mathcal{E}_{int} - \mathcal{E}_{alg}, \tag{32}$$

where

$$\mathcal{E}_{int} = \Delta E_{kin}^{nodes} - \Delta E_{kin}, \quad \mathcal{E}_{alg} = -\Delta E_{kin}^{nodes} - \Delta E_{pot}.$$

In order to test the accuracy of the different elements used, the quantities E_{kin}, E_{kin}^{nodes}, E_{pot}, E_{tot}, $\mathcal{E}_{int}$ and $\mathcal{E}_{alg}$ are calculated at each time step of the simulation. The results are shown in Figures 4 to 7 using lumped masses.

It is seen that linear elements, the 9-node quadratic elements and the cubic splines all yield smooth vibrations, while the 8-node serendipity elements perform very poorly. This is likely due to the negative grid masses associated with corner nodes. Also, it is observed that the total energy is almost conserved for all the simulations. Hence, the interpolation error and the algorithm error cancel. This is observed very clearly for the eight-node elements, where the errors are large, but total energy is nevertheless conserved. Comparing the magnitude of the interpolation error, it is smallest for the nine-node interpolation, followed by the linear elements. Somewhat surprisingly, the interpolation error is larger using cubic splines than using linear elements. When consistent masses are used, the interpolation error becomes zero. In Figure 8 the interpolation error is compared for the linear and the quadratic 9-node elements and it is clearly observed that the quadratic interpolation reduces the interpolation error.

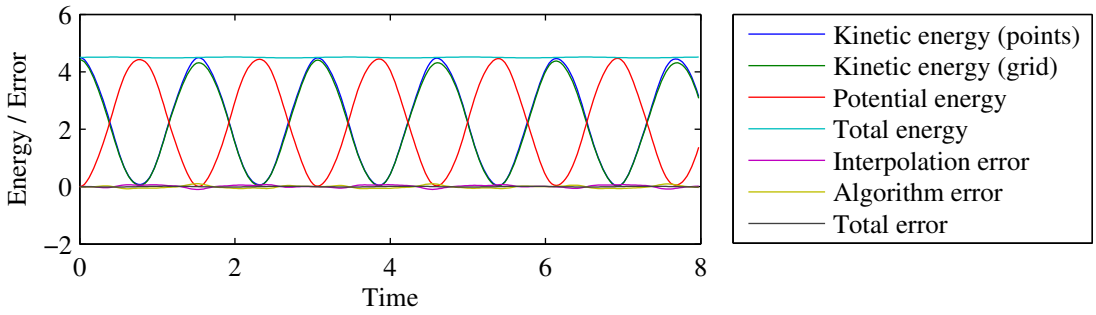


Figure 4: Energy measures for the vibrating-bar problem using 4-node elements.

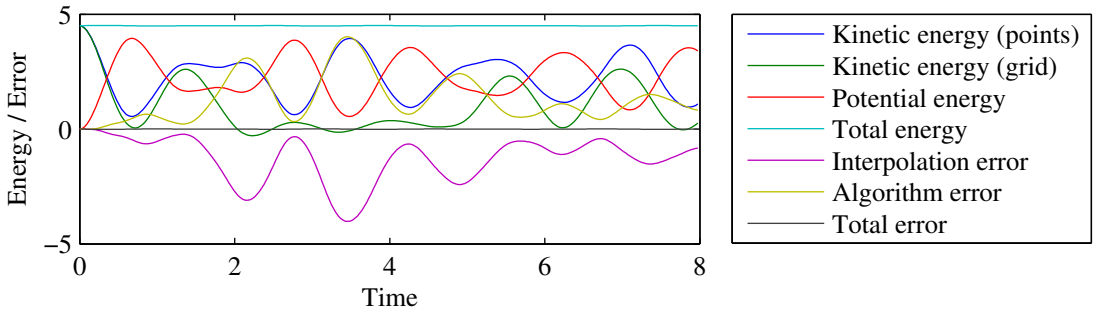


Figure 5: Energy measures for the vibrating-bar problem using 8-node elements.

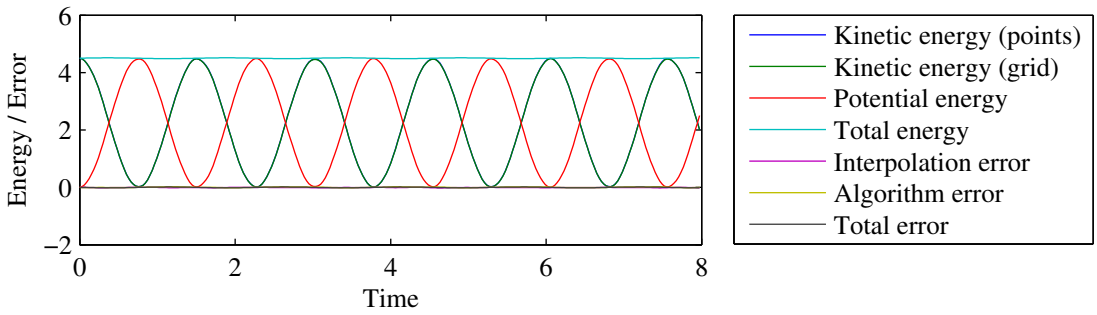


Figure 6: Energy measures for the vibrating-bar problem using 9-node elements.

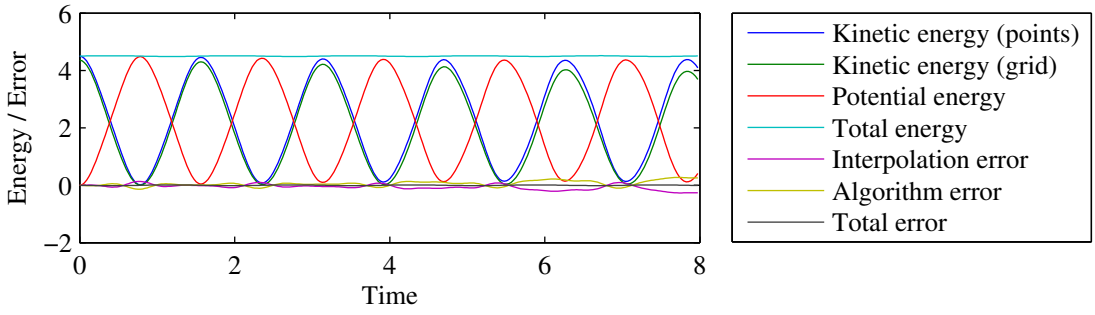


Figure 7: Energy measures for the vibrating-bar problem using cubic-splines.

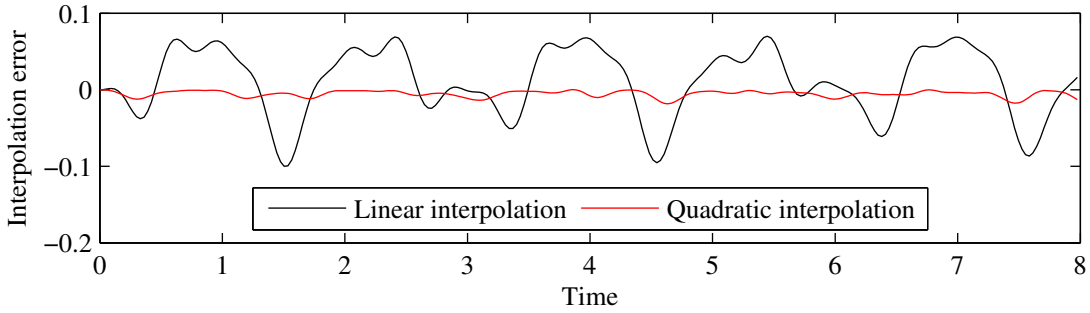


Figure 8: Comparison of the interpolation errors.

4.1 Bending of a cantilevered beam

Next, the problem presented in Subsection 3.3 is analysed for a beam with the length $L = 8$ and the height $H = 1$. In [12] an analytical solution for the deflections is given for linear-elastic materials. The horizontal deflection is given by

$$u(x,y) = -\frac{Px^2y}{2EI} - \frac{\nu Py^3}{6EI} + \frac{Py^3}{6GI} + (\frac{PL^2}{2EI} - \frac{Pc^2}{2GI})y \tag{33}$$

and the vertical deflection is given by

$$v(x,y) = \frac{\nu Pxy^2}{2EI} + \frac{Px^3}{6EI} - \frac{PL^2x}{2EI} + \frac{PL^3}{3EI}. \tag{34}$$

It is basically a quasi-static problem where equilibrium is presumed for the bar during the deformation. This leaves the question on how to simulate it using the dynamic framework of the material-point method as presented above. Many solutions to this probably exist. Here it is chosen to gradually apply a force as a material acceleration $\mathbf{b}(\mathbf{x}_P, t)$. The force is linearly increased starting from zero during the simulation.

The material point in the upper right-hand corner is given a material acceleration corresponding to the downward force F applied as

$$b_{p,y} = \frac{-F}{m_p}\frac{m_p}{m_{tot}}, \tag{35}$$

where $b_{p,y}$ denotes the y-component of acceleration of the material point, m_p is the mass associated with the material point, at which the acceleration is applied, and m_{tot} is the total mass of the beam. The undeformed model is shown in Figure 9 and the material parameters for the beam are: $E = 3 \cdot 10^5$, $\nu = 0$ and $\rho_0 = 10^3$. The number of material points is 288, whereas 133 grid nodes are used. In the initial configuration, 3×3 uniformly distributed material points are applied within each computation cell. The total simulation time is $t_{tot} = 20$ with the time step $\Delta t = 0.003$.

Only a force of $F = 100$, corresponding to very small deformations, is applied. This is done for two reasons: Firstly, in order for the analytic solution to be valid, small deflections are presumed; secondly, small deflections are necessary to avoid a change in sign of the nodal mass for the quadratic 9-node elements.

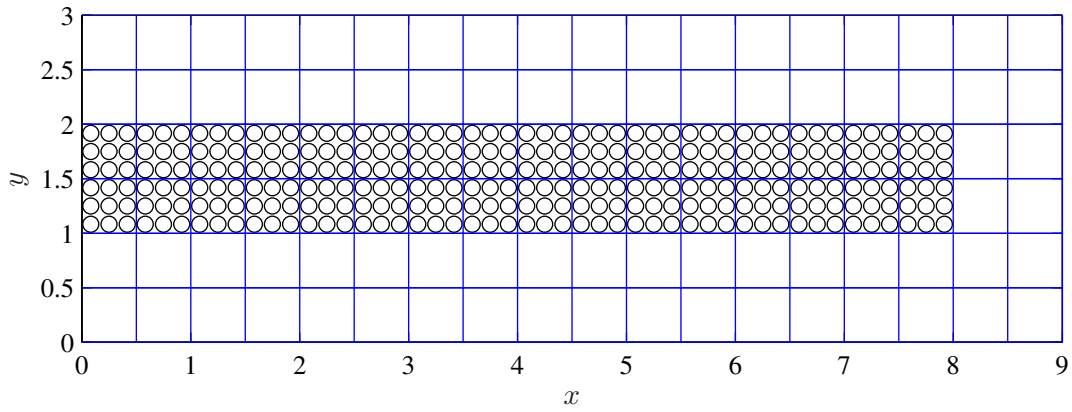


Figure 9: The discretization of the cantilevered beam.

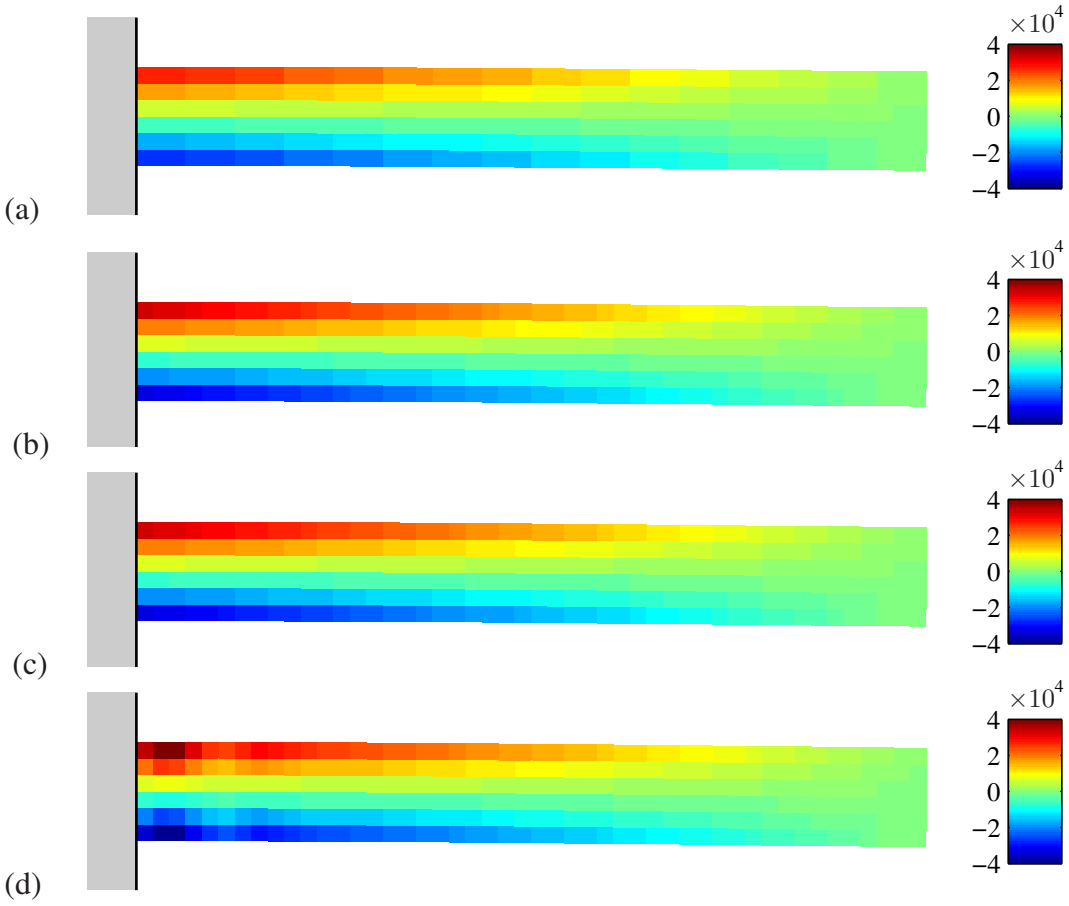


Figure 10: Horizontal normal stresses in the cantilevered beam at the end of the simulation and with different kinds of interpolation: (a) linear interpolation; (b) serendipity elements; (c) quadratic interpolation; (d) cubic splines.

The normal stresses at the material points at the end of the simulation with the different element types are shown in Figure 10. It is seen that the correct variation of the stress, linearly in both directions, is obtained for the eight- and nine-node elements.

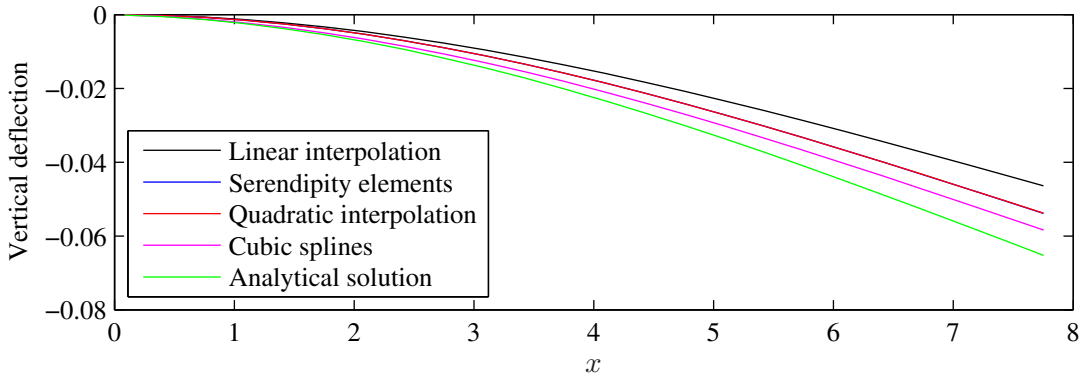


Figure 11: Vertical deflection of the beam neutral axis at the end of the different MPM simulations compared with the analytical solution.

The stresses are of the same magnitude for the two types of elements. For the linear elements, as determined by the gradients of the shape functions, the horizontal stresses within an element are constant while the correct linear variation is observed in the x-direction, i.e. along the beam. For the simulation using cubic splines it is seen that the combination of the cubic splines in the interior and linear elements at the border introduces an nonphysical stress variation.

In order to display the deflection of the neutral beam axis, the average deflection of the two material points near the neutral beam axis is calculated and plotted against the x-coordinate. The result is shown in Figure 11. The analytical solution is also plotted. As seen there is a considerable difference in the magnitude of the deflections. The smaller deflections are observed for the linear interpolation while the simulation using the cubic splines yields the largest deformations. The deflections found using the eight- and nine-node elements are almost identical. The cubic splines yield a result that is closest to the analytical solution, while the linear and the quadratic interpolations yield deflections significantly lower than the analytical solutions.

In order to show the variation of the deflections, another simulation is performed using cubic splines. The force is increased by an order of magnitude to $F = 800$. The final positions of the material points are plotted along with the analytical deflection calculated from Eq. (33) and (34). The result is shown in Figure 12. Clearly, the simulation correctly captures the basic features of the analytic solution.

4.2 A box impacting on a rigidly supported beam

A rigidly supported beam is hit by a falling box. The setting of the problem is shown in Figure 13. The beam has the material properties $E = 10^5$, $\nu = 0.3$ and $\rho_0 = 10^3$, whereas the box has the properties $E = 10^6$, $\nu = 0.3$, $\rho_0 = 10^3$. As illustrated in Figure 13, 864 material points with 3×3 material points per cell are applied, and there are 441 grid nodes in the model. The total time of the simulation is $t_{tot} = 3$, using a time step of $\Delta t = 0.005$, and the initial velocity of the box is $\mathbf{v}_{box}^{init} = (0; -2)$, i.e. the box moves downwards.

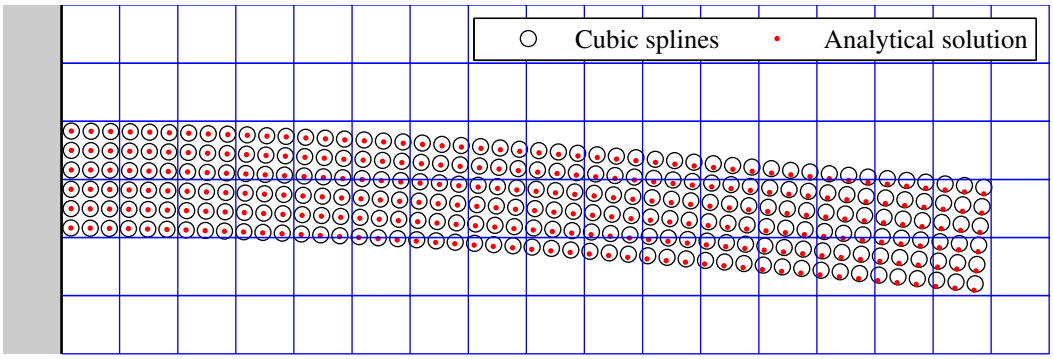


Figure 12: The deflection of the beam obtained using cubic splines compared to the analytic solution ($F = 800$).

A change of sign of the grid masses occurs when the nine-node elements with quadratic interpolation are employed. Hence, this kind of element is not able to simulate the problem. For the remaining element types, Figures 14, 15 and 16 show the energy and the computational errors. The shared velocity field for the two bodies leads to a funny-looking final configuration, where the bar has got an upward deflection as shown in Figure 17. The energy measures do not give a very clear picture of what is going on. Therefore, the average y-position of the material points that make up the box is plotted as a function of time in Figure 17.

The collision modelled using the cubic splines gives a much smoother collision than the linear elements or the eight-node serendipity elements. The upward velocity is considerably lower, indicating that the initial kinetic energy of the box has been transformed to kinetic energy in the beam and strain energy during the collision. Also, the figure indicates that the larger amount of kinetic energy in the box is maintained in the simulations with the eight-node elements.

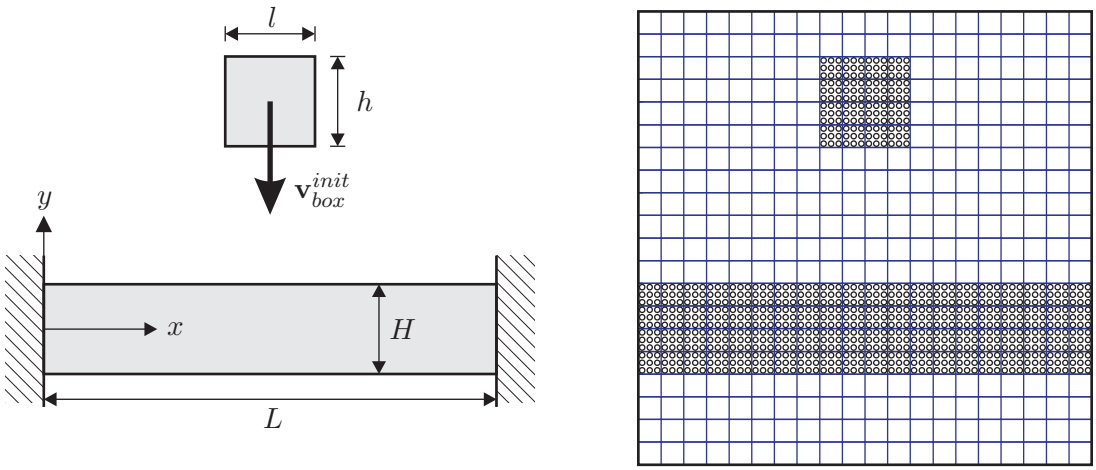


Figure 13: A box falling on a beam: Definition of geometry and boundary conditions (left) and initial discretization (right).

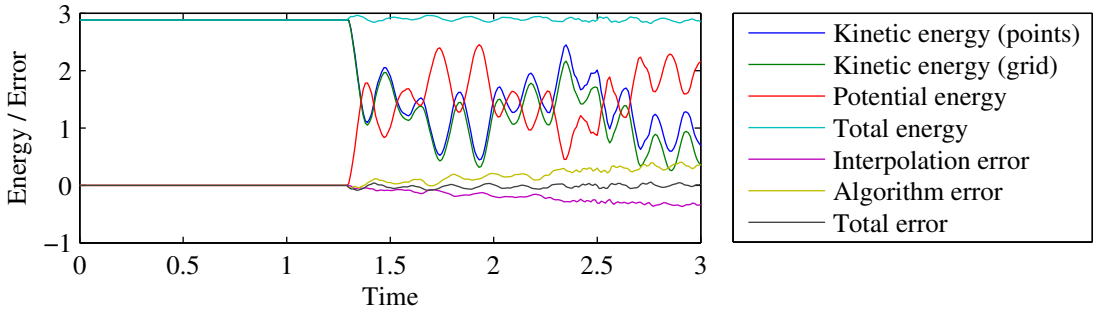


Figure 14: Energy measures for the impact problem using four-node elements.

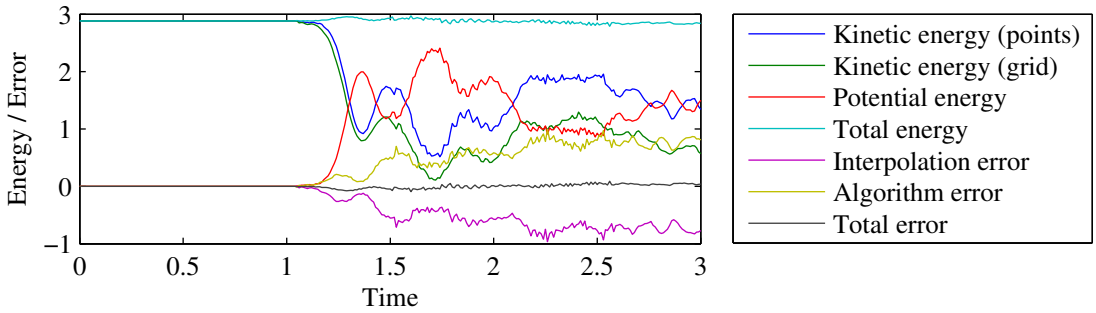


Figure 15: Energy measures for the impact problem using eight-node elements.

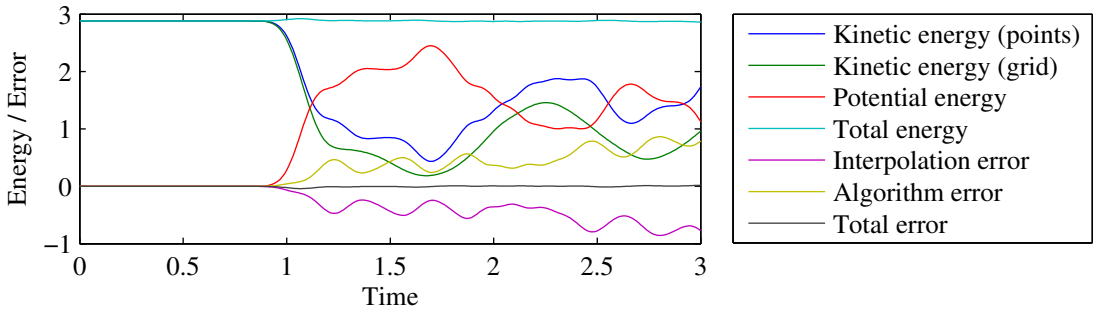


Figure 16: Energy measures for the impact problem using cubic splines.

5 Conclusion

Different types of interpolation for the material-point method have been tested. As shown in the first numerical example, the vibration of a linear-elastic bar, quadratic interpolation provides the most accurate results for the same number of material points and the same global number of grid nodes. However, when large deformations occur, the sign of the mass interpolated to the grid nodes changes for some of the nodes and the use of quadratic elements breaks down.

The main advantage of the material-point method is the ability to handle large deformations, so therefore the use of quadratic elements is very limited. This leads to the study of interpolation using cubic splines. As shown for the vibrating-bar problem, the order of accuracy using cubic splines is only comparable to the linear interpolation.

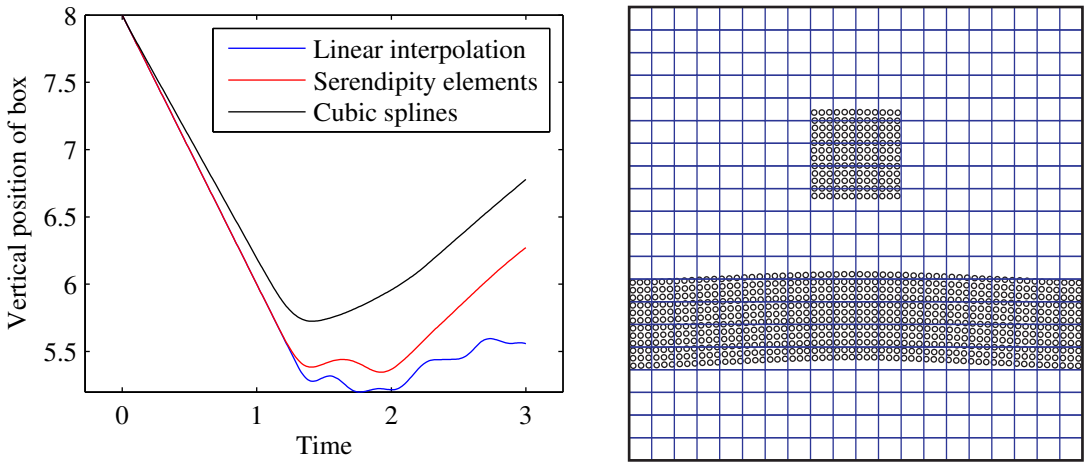


Figure 17: Vertical position of the box during the simulation (left) and the deformation in the final configuration using cubic splines (right).

The main advantage of using the cubic splines is the smoother field representation. The advantage is demonstrated by a third numerical simulation, a falling box impacting with a beam. When linear interpolation is used, the box oscillates on the beam upon collision before it finally bounces off. Using cubic splines, the box hits the bar and bounces off in one smooth movement as physically expected. Therefore, although computationally more expensive, the cubic splines are better suited for problems in which the modelled field quantities vary in a complex way.

Perhaps the main conclusion is that, as the quadratic interpolation is unstable and the cubic splines do not observe a higher order of accuracy for this implementation of the MPM, the means to obtain higher accuracy for the MPM lies elsewhere. Things to explore could be other ways of calculating internal forces by using another integration than the Riemann sum over material points.

References

[1] H.L. Schreyer, D. Sulsky and Z. Chen. “A particle method for history-dependent materials”, Computer Methods in Applied Mechanics and Engineering, 118, 179–196, 1994.

[2] H.L. Schreyer, D. Sulsky and S.J. Zhou. “Application of a particle-in-cell method to solid mechanics”, Computer Physics Communications, 87, 236–252, 1995.

[3] F.H. Harlow. “The particle–in–cell computing method for fluid dynamics”, Methods for Conputational Physics, 3, 319–343, 1964.

[4] H.L. Schreyer and D. Sulsky. “Axisymmetric form of the material point method with applications to upsetting and taylor impact problems”, Computer Methods in Applied Mechanics and Engineering, 139, 409–429, 1996.

[5] A.R. York. “Development of modifications to the material point method for the simulation of thin membranes, compressible fluids, and their interfaces”. Mas-

ter's thesis, Sandia National Laboratories, July 1997.
[6] H.L. Schreyer, A.R. York and D. Sulsky. "The material point method for simulation of thin membranes", International Journal for Numerical Methods in Engineering, 44(10), 1429–1456, 1999.
[7] S.G. Bardenhagen. "Energy conservation error in the material point method for solid mechanics", Journal of Computational Physics, 180(1), 383–403, 2002.
[8] J.A. Weiss and J.E. Guilkey. "Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method", International Journal for Numerical Methods in Engineering, 57(9), 1323–1338, 2003.
[9] H.L. Schreyer and D. Sulsky. "MPM simulation of dynamic material failure with a decohesion constitutive model", European Journal of Mechanics A/Solids, 23(3), 423–445, 2004.
[10] E.M. Kober and S.G. Bardenhagen. "The generalized interpolation material point method", CMES, 5(6), 477–495, 2004.
[11] D. Sulsky D and E. Love. "An unconditionally stable, energy-momemtum consistent implementation of the material point method", Computer Methods in Applied Mechanics and Engineering, 195(33-36), 3903–3925, 2006.
[12] J.N. Goodier and S.P. Timoshenko. "Theory of elasticity", McGraw-Hill, 3rd edition, 1970.

Appendix: Material-point method with negative mass

In the traditional version of the material-point method (MPM) using linear interpolation, the shape functions are always positive. This is not the case for the nine-node quadratic element and the eight-node serendipity element. Here it is analysed, how the method behaves, when negative grid masses are encountered. It is presumed that lumped masses are used and that all the material points have the same mass. Without any loss of generality, only a rigid-body motion is considered.

Step 1: A negative grid mass m_i^k is encountered at node i and time step k.

Step 2: The grid velocities are found according to the definition

$$\mathbf{v}_i^k = \frac{1}{m_i} \sum_{p=1}^{N_p} m_p \mathbf{v}_p^k \Phi_i(\mathbf{x}_p^k), \quad i = 1, 2, \ldots, N_n. \tag{36}$$

This yields the correct sign of the velocity relatively to the velocity of the material points as the negative mass is cancelled by the negative shape function value.

Step 3: The strain increments at the material points are calculated as

$$\Delta\boldsymbol{\varepsilon}_p^k = \frac{\Delta t^k}{2} \sum_{i=1}^{N_n} \left(\mathbf{G}_{ip}^k \mathbf{v}_i^k + (\mathbf{G}_{ip}^k \mathbf{v}_i^k)^T \right), \quad p = 1, 2, \ldots, N_p. \tag{37}$$

If the gradients of the shape functions are correct, this is expected to yield the correct result. The update of stains and stress does not cause any problems in this context.

Step 4: Calculation of the internal forces in the background grid by means of

$$\mathbf{f}_i^{int,k} = -\sum_{p=1}^{N_p} \frac{m_p}{\rho_p^k} \boldsymbol{\sigma}_p^k \cdot \mathbf{G}_{ip}^k, \quad i = 1, 2, \ldots, N_n. \tag{38}$$

Again, this is to be correct if the gradients of the shape functions are correct.

Step 5: Update the velocity of the material points as

$$\mathbf{v}_p^{k+1} = \mathbf{v}_p^k + \Delta t^k \sum_{i=1}^{N_n} \sum_{j=1}^{N_n} m_{ij}^{-1} \mathbf{f}_j^{int,k} \Phi_j(\mathbf{x}_p^k), \quad p = 1, 2, \ldots, N_p. \tag{39}$$

Here the negative sign of the mass and the negative sign of the shape-function also cancel each other, leading to the a consistent velocity update.

Step 6: Update the position of the material points,

$$\mathbf{x}_p^{k+1} = \mathbf{x}_p^k + \frac{\Delta t^k}{m_p} \sum_{i=1}^{N_n} \mathbf{p}_i^{k+1} \Phi_i(\mathbf{x}_p^k), \quad p = 1, 2, \ldots, N_p, \tag{40}$$

where the updated momentum is given by $\mathbf{p}_i^{k+1} = \mathbf{v}_i^{k+1} m_i^k + \mathbf{f}_i^{res,k} \Delta t^k$. Here it is observed that the momentum is directed opposite to the material momentum. In the update of the material point positions, three quantities are directed opposite to the material quantities: $\mathbf{p}_i^{k+1}$, m_i^k and $\Phi_i(\mathbf{x}_p^k)$. Therefore, the material-point update is inconsistent when negative mass is encountered.

In the testing of the eight- and nine-node elements, two characteristic features are observed. The nine-node quadratic element yields very good results when the grid masses do not change sign, but breaks down immediately when the masses change sign. For the eight-node element, the shape functions associated with the four corner nodes are always negative, while the shape functions associated with the four side nodes are always positive. The results obtained with the eight-node element are very inaccurate, but it does not get unstable, as does the nine-node element. These two observations are probably due to the 'self restoration' inherent in the material-point method. The incorrect material-point update is to a certain extent compensated for in the next time step, when the new internal forces are calculated. If the sign of the nodal mass at a grid node changes between two time steps, this restoration becomes a destabilisation instead. Consequently, the simulations become unstable, as observed for the nine-node quadratic element.