# A Dynamic Explicit Total Lagrangian MPM for Large Deformation in 1D

**CVEN 7511 Final Project Report**

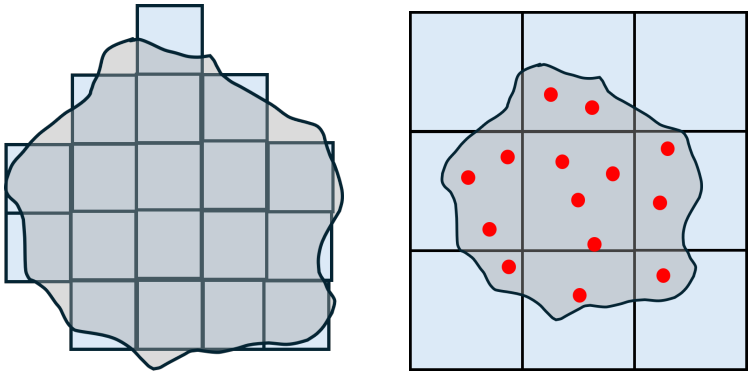
Jacob Nuttall
*University of Colorado Boulder, Materials Science and Engineering*
(Dated: March 11, 2025)

**CONTENTS**

## I. INTRODUCTION

Many solid mechanics problems, such as those involving damage and fracture, cannot easily be simulated using the finite element method (FEM). This is because such problems can lead to entangled meshes or discontinuous meshes, which lead to numerical instability and failure when solving such problems using a mesh-based method such as FEM. Instead, these problems may be formulated using a mesh-free method such as the material point method (MPM).

The MPM, an evolution of methods such as particle-in-cell and fluid-implicit particle method, avoids such problems [1]. This is because, in the MPM, rather than tracking the movement of a mesh representing a solid material of interest, instead the movement of Lagrangian material points with respect to an Eulerian background grid are used to simulate the motion and deformation. Such material points are closely related to the integration points (also material points) used in the FEM, as just like in the FEM they are used in the MPM to track material properties and internal state variables such as Cauchy stress and deformation gradient. The major difference between MPM and FEM is that, in the MPM, material points are allowed to move freely in a domain and the background grid is used solely to solve the equations of motion, whereas in the FEM the positions of the material points are fixed with respect to a particular mesh element and the mesh or grid is allowed to move and be deformed. This difference in problem initialization and structure between the FEM and the MPM is visualized in Fig. 1.



(a) FEM mesh initialization

(b) MPM grid and material point initialization

FIG. 1. A visual comparison between the problem initialization for the FEM in I for an object in 2D. Specifically, initializing the MPM requires defining particle locations for the material points (red) and a background grid whereas the FEM defines a discrete mesh over the material. Notice that, in contrast to the FEM, the domain of the background grid in the MPM extends beyond the extent of the material and that the material is represented specifically by the material points, not the grid. This property of the MPM illustrates how it is 'meshless'.

There exist many variations on implementations of the material point method, such as explicit vs. dynamic formulations, Eulerian vs. Lagrangian approaches, and in how the basis functions used to interpolate material properties from the material points are formed on the background grid. The objective of this project is to understand how the material point method is implemented with explicit dynamics in 1D with a total Lagrangian approach as in [2] using standard quadratic finite element basis functions for the background grid. Specifically, this project will derive an algorithm for a total Lagrangian material point method (TLMPM) for a one-dimensional problem involving an axially-loaded bar using a Neo-Hookean hyperelastic model with no lateral displacement, so that the cross-sectional area remains constant. The resulting MPM algorithm will be verified by comparing results against a total lagrangian FEM (TLFEM) code for three cases: First, results from the TLMPM are compared against results from the TLFEM for the quasi-static compression of a columnm. Then, the MPM results are compared against an analytical solution for the dynamic problem of an axially step-loaded bar at small strain [3]. Lastly, the MPM is verified by checking that the resulting solution exhibits proper energy behavior.

## II. THEORY AND DERIVATION OF THE TLMPM IN 1D

### A. The Strong Formulation for an axially-loaded bar

Deriving an effective theory for a TLMPM in 1D for the case of an isotropic axially-loaded bar begins in much the same manner as a TLFEM. We begin the derivation here by considering an isotropic Neo-Hookean bar of length

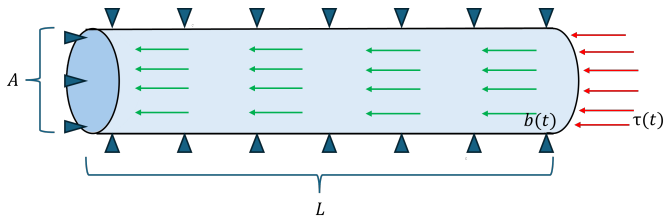



FIG. 2. Diagram of the axially-loaded bar of length $L$, cross-sectional area $A$, and density $\rho$ constrained to move in only one dimension with applied traction and body force $\tau(t)$ and $b(t)$ at time $t$, respectively.

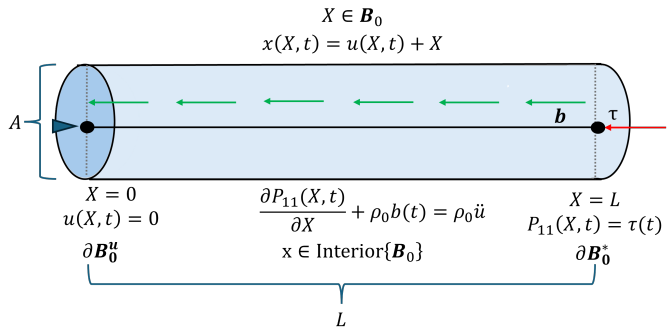



FIG. 3. Graphical representation of the strong form of the axially-loaded bar in 1D. The balance equation of linear momentum $\partial S_{11}/\partial X + \rho_0 b(t) = \rho_0 a$ is solved along the interior of $\boldsymbol{B}_0$, i.e., Interior$\{\boldsymbol{B}_0\} = (0, L)$, to find the displacement field $u(X,t) = x(X,t) - X$ such that the boundary conditions $u(X = 0, t) = 0$ and $P_{11}(X = L, t) = \tau(t)$ are satisfied on $\partial \boldsymbol{B}_0$ for all time $t$.

$L$ and cross-sectional area $A$ with density $\rho$ having elastic modulus $E$ and Poisson's ratio $\nu$, from which the Lamé parameters $\mu$ and $\lambda$ are obtained as

$$\mu = \frac{E}{2(1+\nu)}, \qquad \lambda = \frac{2\mu\nu}{1-2\nu}. \tag{1}$$

The wavespeed of the bar is given by

$$c = \sqrt{E/\rho} \tag{2}$$

One end of the bar has is fixed in place at zero displacement from its initial position and the other end of the bar has an applied traction $\tau(t)$. The bar is also subject to a body force $b(t)$ (such as gravity). The sides of the bar are also constrained to such that there is zero lateral displacement so the motion lies entirely in 1D. Such a model could, for example, approximate the motion of a spring under large deformation. This situation is diagrammed in Fig. 2.

The objective of this problem is to find the displacement $u(X,t)$ for each point $X \in (0, L) = \boldsymbol{B}_0$ in the reference (Lagrangian) configuration and time $t \in [0, T]$. The traction is applied on $\partial \boldsymbol{B}_0^* = \{L\}$ and displacement is prescribed on $\partial \boldsymbol{B}_0^u = \{0\}$, where the boundary for $\boldsymbol{B}_0$ is written as $\partial \boldsymbol{B}_0 = \{0, L\} = \partial \boldsymbol{B}_0^u \cup \partial \boldsymbol{B}_0^*$ . This situation for the problem of an axially-loaded bar in 1D may then be stated as in the following strong formulation with the balance of linear momentum, initial conditions of zero displacement and velocity, and boundary conditions (BCs) described [4]:

$$\text{(Strong Form)} \begin{cases} \text{Find displacement } u(X,t) : \boldsymbol{B}_0 \times [0,T] \mapsto \mathbb{R} \text{ for a Neo-Hookean hyperelastic bar such that the} \\ \text{following balance of linear momentum, BCs, and standard initial conditions are satisfied:} \\ \frac{\partial P_{11}(X,t)}{\partial X} + \rho_0 b(t) = \rho_0 \ddot{x}(X,t) = \rho_0 a(X,t), & X \in \text{Interior}\{\boldsymbol{B}_0\} & \text{(Balance of linear momentum)} \\ u(X,t) = g(X,t), & X \in \partial \boldsymbol{B}_0^u & \text{(Displacement BC on } \partial \boldsymbol{B}_0^u) \\ P_{11}(X,t) = \tau(t), & X \in \partial \boldsymbol{B}_0^* & \text{(Traction BC on } \partial \boldsymbol{B}_0^*) \\ u(X, t=0) = 0, & X \in \boldsymbol{B}_0 & \text{(Initial Displacement)} \\ v(X, t=0) = \dot{x}(X, t=0) = v_0(x), & X \in \boldsymbol{B}_0 & \text{(Initial Velocity)} \end{cases} \tag{3}$$

Fig. 3 provides a visual representation of the strong form for this problem. In this formulation, $P_{11}$ is the first Piola-Kirchoff stress tensor in 1D and $a$ is the acceleration. Note that the Interior$\{\boldsymbol{B}_0\} = (0, L)$ is the interior of the

reference configuration that does not include the boundary $\partial \boldsymbol{B}_0$. Computing $P_{11}(X,t)$ is done by obtaining $S_{11}(X,t)$ from the constitutive assumption of a Neo-Hookean hyperelasticity model [4, 5] with strain energy energy density

$$\rho_0 \phi(\boldsymbol{C}) = \frac{1}{2}\lambda(\ln J)^2 - \mu \ln J + \frac{1}{2}\mu(\mathrm{Tr}\,\boldsymbol{C} - 3), \tag{4}$$

where $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$ is the right Cauchy-Green tensor, from which it can be shown that

$$S_{11} = \frac{\partial(\rho_0\phi)}{\partial C_{11}} = \mu + (\lambda \ln(J) - \mu)C_{11}^{-1} \tag{5}$$

so that $P_{11} = F_{11}S_{11}$, In Eq. (5), $J$ is the determinant of the deformation gradient, $\boldsymbol{F}$, where for problems with motion constrained along only one dimension,

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}(\boldsymbol{X},t)}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial u}{\partial X} - 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

so that the only non-trivial component of $\boldsymbol{F}$ is $F_{11} = \frac{\partial u}{\partial X} - 1$, which leads to $J = \det \boldsymbol{F} = F_{11}$. $C_{11}^{-1}$ is the inverse of the right Cauchy-Green tensor in 1D given by $C_{11}^{-1} = F_{11}^{-2}$.

### B. Weak Formulation

The weak form is obtained by introducing a weighting function $w(X) \in H^1$, where $H^1$ is the first Sobolev space and $w(X) = 0$ where there is a prescribed displacement, i.e., $u(X=0,t) = g_0(t)$, and a trial displacement function $u(X,t) \in H^1$ that satisfies the same boundary conditions as in Eq. 5. In particular, define vector spaces

$$\boldsymbol{\mathcal{U}} = \{u(X,t) \mid u(X,t) : [0,L] \times [0,T] \mapsto \mathbb{R}, \tag{7}$$

$$u \in H^1, u(X=0,t) = g_0(t), u(X,t=0) = 0, \dot{u}(X,t=0) = v_0\big\}, \tag{8}$$

as the space of all possible trial displacement functions and

$$\boldsymbol{\mathcal{W}} = \big\{w(X) \mid w(X) : [0,L] \mapsto \mathbb{R}, w \in H^1, w(0) = 0\big\} \tag{9}$$

as the space of all weighting functions.

Then, multiply the equations in Eq. 3 by $w(X)$ and integrate over the domain $X \in \boldsymbol{B}_0$ to get a variational equation of the balance of linear momentum $G\{w(X), u(X)\}$ where

$$G = G\{w(X), u(X)\} = \int_{\boldsymbol{B}_0} dX \left[ w(X)\frac{\partial P_{11}}{\partial X} + w(X)\rho_0 b(t) - w(X)\rho_0 a(X,t) \right] = 0 \tag{10}$$

By applying the chain rule obtain $w(X)\frac{\partial P_{11}(X,t)}{\partial X} = \frac{\partial [w(X)P_{11}(X,t)]}{\partial X} - \frac{\partial w(X)}{\partial X}P_{11}(X,t)$, obtain

$$G = \int_{\boldsymbol{B}_0} dX \left[ \frac{\partial [w(X)P_{11}(X,t)]}{\partial X} - \frac{\partial w(X)}{\partial X}P_{11}(X,t) + w(X)\rho_0 b(t) - w(X)\rho_0 a(X,t) \right]. \tag{11}$$

Note that

$$\int_{\boldsymbol{B}_0} dX \left[ \frac{\partial [w(X)P_{11}]}{\partial X} \right] = w(X)P_{11}(X,t)\Big|_{\partial \boldsymbol{B}_0} = w(L)P_{11,t}(L) - w(0)P_{11}(0,t) = w(L)\tau(t), \tag{12}$$

since $w(0) = 0$ and $P_{11}(X = L, t) = \tau$, the applied traction. The above can be written making use of the dirac delta $\delta(X)$ as

$$\int_{\boldsymbol{B}_0} dX[w(X)\tau(t)\delta(X-L)]. \tag{13}$$

Then

$$G = \int_{\boldsymbol{B}_0} dX \left[ -\frac{\partial w(X)}{\partial X}P_{11}(X,t) + w(X)\rho_0 b - w(X)\rho_0 a(X,t) + w(X)\tau(t)\delta(X-L) \right], \tag{14}$$

which may be written more compactly by defining

$$
\begin{aligned}
G_1^{\text{INT}} &= \int_{\boldsymbol{B}_0} dX \left[ -\frac{\partial w(X)}{\partial X} P_{11}(X,t) \right], \\
G_1^{\text{EXT}} &= \int_{\boldsymbol{B}_0} dX \left[ w(X) \rho_0 b(t) \right], \\
G_2^{\text{INT}} &= \int_{\boldsymbol{B}_0} dX \left[ -w(X) \rho_0 a(X,t) \right], \\
G_2^{\text{EXT}} &= \int_{\boldsymbol{B}_0} dX \left[ w(X) \tau(t) \delta(X-L) \right],
\end{aligned} \tag{15}
$$

so that

$$
G = G_1^{\text{INT}} + G_1^{\text{EXT}} + G_2^{\text{INT}} + G_2^{\text{EXT}} = 0. \tag{16}
$$

With this result the weak formulation of the problem of the axially-loaded bar may be stated as

$$
\text{(Weak Form)} \begin{cases} \text{Find a trial displacement function } u(X,t) \in \boldsymbol{\mathcal{U}}, \\ \quad \text{for a Neo-Hookean hyperelastic bar such that} \\ \qquad G = 0 \\ \quad \text{holds for all weighting functions } w(X) \in \boldsymbol{\mathcal{V}}. \end{cases} \tag{17}
$$

### C. Discretization

Up to now, the derivation of the TLMPM has been the same as that for the TLFEM: from the strong form obtain a weak form. The next step is approximating the weak form in Eq. (17) by discretizing the problem domain. In the TLFEM, this is done by introducing elements which integrate over $G$ piecewise using shape functions $\boldsymbol{N}$ and then applying the assembly operator over the elements and a nonlinear solver like Newton-Raphson to determine the final displacement solution. The TLMPM is similar in that the same shape functions $\boldsymbol{N}$ may be employed as are typically used in TLFEM to interpolate grid quantities to material points and extrapolate material properties from the material points to the background grid. However, because the background grid is Eulerian, it remains fixed in place throughout solving for Eq. (17).

### D. The Background grid for TLMPM

#### 1. *Bubnov-Galerkin Form*

We obtain the background grid TLMPM as part of discretizing the weak form by first considering the discretization for TLFEM. In particular, note the Bubnov-Galerkin form used to approximate Eq. (17) in the FEM:

$$
\text{(Galerkin Form)} \begin{cases} \text{Find } u^h(X,t) \in \boldsymbol{\mathcal{U}}^h, \\ \quad \text{for a Neo-Hookean hyperelastic bar such that} \\ \qquad G^h = 0 \\ \quad \text{holds for all weighting functions } w^h(X) \in \boldsymbol{\mathcal{V}}^h. \end{cases} \tag{18}
$$

where, similar to above,

$$
\begin{aligned}
\boldsymbol{\mathcal{U}}^{\boldsymbol{h}} = \big\{ & u^h(X,t) \mid u^h(X,t) : \boldsymbol{B}_0^h \times [0,T] \mapsto \mathbb{R}, \\
& u^h \in H^1, u^h(X=0,t) = g_0(t), u^h(X,t=0) = 0, \dot{u}(X,t=0) = v_0 \big\}, \qquad (19) \\
\boldsymbol{\mathcal{W}}^h = \big\{ & w(X) \mid w(X) : \boldsymbol{B}_0^h \mapsto \mathbb{R}, w^h \in H^1, w^h(0) = 0 \big\}, \qquad (20)
\end{aligned}
$$

with $h$ defined as the characteristic length of the mesh, $\boldsymbol{B}_0^h$. Note that, for 1D, $\boldsymbol{B}_0^h = \boldsymbol{B}_0$ though this is not true in general for problems in 2D or 3D.

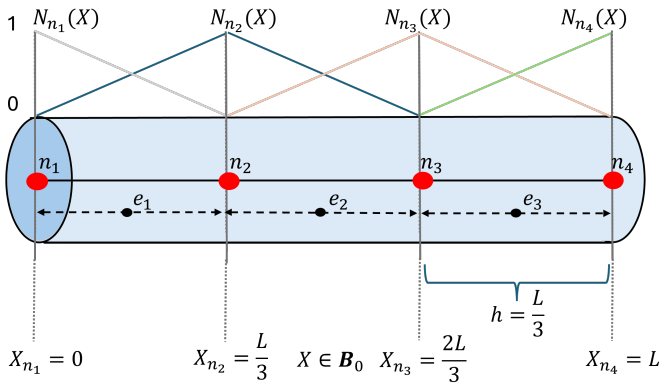



FIG. 4. A visual of the domain mesh discretization in 1D with $n_{el} = 3$ elements $e_k$ for an isotropic bar. The characteristic length of the mesh is $h = L/3$ and, in the case of linear shape functions $N_{n_a}(X)$, the mesh has global nodes $n_a$ denoted by red circles at locations $X_{n_a}$, with $a = 1, 2, .., n_{gn}$ where $n_{gn} = 4$ is the number of global nodes. Notice that, with linear shape functions, each element $e_k$ has two 'local' nodes: $n_k$ and $n_{k+1}$. Also, for each shape function, $N_{n_a}(X_{n_a}) = 1$.

### 2. *The Mesh for the TLFEM*

In the case of the 1D axially-loaded bar, the mesh $B_0^h$ is a piecewise discretization of the reference domain $B_0 = [0, L]$. The mesh is defined with $n_{el}$ elements $e_k$, where $k = 1, 2, ..., n_{el}$, where the domain of each element $\boldsymbol{B}_{0,e_k}^h$ is such that $\boldsymbol{B}_0^h = \cup_{k=1}^{n_{el}} \boldsymbol{B}_{0,e_k}^h$ and the characteristic size is $h = L/n_{el}$. Typically, $\boldsymbol{B}_{0,e_k}^h = \left[\frac{kh}{L}, \frac{(k+1)h}{L}\right]$. For example, if the mesh has three elements, then $h = \frac{L}{3}$ and the reference domain is partitioned so that $B_0^h = [0, \frac{L}{3}] \cup [\frac{L}{3}, \frac{2L}{3}] \cup [\frac{2L}{3}, L]$.

The mesh also has $n_{gn}$ global nodes $n_a$ with collocated positions $X_{n_a}^h \in \boldsymbol{B}_0$ with $a = 1, 2, ..., n_{gn}$, where the number of nodes is determined by the order of shape functions $\boldsymbol{N} = [N_{n_a}]$. Shape functions are defined with the property that $N_{n_a}(X) = 1$ if $X = X_{n_a}$ and 0 if $X = X_{n_b}$ for all $b \neq a$. For example, assuming a mesh with $n_{el}$ elements and linear shape functions, the number of global nodes is $n_{gn} = n_{el} + 1$, the locations of the nodes are $X_{n_a}^h = ah/L$, and the shape functions are evaluated as

$$N_{n_a}(X) = \frac{h - |X - X_{n_a}^h|}{h} \begin{cases} 1, & X \in \boldsymbol{B}_h \text{ and } |X - X_{n_a}^h| < h, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Note that with linear shape functions (polynomials of degree one), each element $e_k$ is bordered by two global nodes, $n_k$ and $n_{k+1}$, referred to as the local nodes of element $e_k$.

The gradient of linear shape functions $N_k$ in 1D are computed as

$$B_{n_a}(X) = \frac{dN_{n_a}(X)}{dX} = \frac{1}{h} \begin{cases} 1, & X \in \boldsymbol{B}_0^h \text{ and } X < X_{n_a}^h, |X - X_{n_a}^h| < h \\ -1 & X \in \boldsymbol{B}_0^h \text{ and } X > X_{n_a}^h, |X - X_{n_a}^h| < h \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

With these results for the shape functions $\boldsymbol{N}$, the displacement $u^h(X, t)$ is computed as

$$u^h(X, t) = \sum_{a=1}^{n_{gn}} N_{n_a}(X) d_{n_a}(t) = \boldsymbol{N}(X) \cdot \boldsymbol{d}_n(t), \tag{23}$$

where $\boldsymbol{d}_n(t) = \{d_{n_a}\}$ are the nodal displacements at time $t$ corresponding to locations $\boldsymbol{X}_n = \{X_{n_a}\}$, respectively. Notice that the effect of the shape functions $\boldsymbol{N}$ is to interpolate nodal displacements onto the domain $\boldsymbol{B}_0^h$. In the Bubnov-Galerkin method, the weighting function is also computed using the same shape functions so that

$$w^h(X) = \sum_{a=1}^{n_{gn}} N_{n_a}(X) w_{n_a} = \boldsymbol{N}(X) \cdot \boldsymbol{w}_n, \tag{24}$$

where $\boldsymbol{w}_n = \{w_{n_a}\}$ are the nodal weights.

The power of the TLFEM comes from the fact that by applying the discretizations as described in Eqs. (23) and (24), the problem of solving $G^h$ in Eq. 16 can be recast by integrating over element domains using Gauss-Legendre quadrature points [6, 7] and assembling the results over each element to solve for the displacement field $u^h$. This is done by first calculating the shape functions $\boldsymbol{N}$ in terms of the natural coordinate of each element, $\xi$, where

$$\xi = \begin{cases} -1, & \text{at leftmost element boundary} \\ 0, & \text{at the center} \\ 1, & \text{at the right most element boundary} \end{cases}, \tag{25}$$

so that

$$u_e^h(\xi, t) = \sum_{k=1}^{n_{en}} N_{e_k}(\xi) d_k^e = \boldsymbol{N}_e(\xi) \cdot \boldsymbol{d}_e(t) \tag{26}$$

and similarly for $w_e^h(\xi)$. Then, using this, define element quantities $G^e = G_1^{\text{INT},e} + G_2^{\text{INT},e} + G_3^{\text{INT},e} + G^{\text{EXT},e}$ by integrating over element domains using Gauss-Legendre quadrature, and obtain the nodal displacements $\boldsymbol{d}_n$ by solving

$$G^h = \mathop{\mathbb{A}}_{e=1}^{n_{el}} [G^e] = 0, \tag{27}$$

where $\mathbb{A}$ is the assembly operator as described in Hughes [7]. Time marching over $t \in [0, T]$ is typically done employing a Newmark time integration scheme. Also, because modeling large deformation is a nonlinear problem, a nonlinear solver like Newton-Raphson is typically employed as in Borst et. al. [6].

#### *3. Higher order shape functions*

Higher order shape functions may be defined in terms of the natural coordinate $\xi$ for an element $e$ with $n_{en}$ local nodes where

$$\boldsymbol{N}_k^e(\xi) = \prod_{j=1}^{n_{en}} \frac{\xi - \xi_j}{\xi_k - \xi_j}, \tag{28}$$

with $n_{en}$ abscissa $\xi_k$, $k = 0, 1, ..., n_{en} - 1$, defined on interval $[-1, 1]$. The order $d$ of the shape functions is given by one minus the number of element nodes $n_{en}$; i.e, $d = n_{en} - 1$. One choice of abscissa $\xi_k$ evenly spaces them such that that $\xi_k = \frac{2k}{n_{en}-1} - 1$. The $n_{gn}$ global nodes are obtained by mapping the abscissa $\xi_k$ for element $e_k$ onto the configuration domain $\boldsymbol{B}_0$ for each element $e$. In general, for 1D problems, the number of global nodes is related to the number of local nodes and number of elements by $n_{gn} = n_{el}(n_{en} - 1) + 1$. Other choices of shape functions also exist, with B-Splines a common choice in the MPM [1].

#### *4. The Background Grid for the TLMPM*

We use the same method (i.e., making use of the Bubnov-Galerkin form) for discretizing the configuration domain $\boldsymbol{\Omega}$ for the TLMPM as used in the TLFEM. There are, however, two major distinctions. First, once discretized, the mesh or background grid is not deformed in the TLMPM unlike in the TLFDEM. In the total Lagrangian perspective, this may not seem an important distinction because the computation is done in the reference configuration, but TLMPM still has several advantages over the TLFEM because it is a particle-based method. Specifically, in the TLMPM the discretization domain need not be the same as the extent of the axially loaded bar; i.e., it can be that $\boldsymbol{\Omega} \neq \boldsymbol{B}_0$, though with $\boldsymbol{B}_0 \subset \boldsymbol{\Omega}$. This distinction is important in problems in 2D or 3D where the advantages of the MPM over the FEM are more clear, such as when a finite element mesh does not perfectly emulate the geometry of the material. Furthermore, as a MPM, the TLMPM can handle more readily solve more interesting problems such as those involving damage or contact BCs due to its particle-based nature.

### E. Initialization of Material Points in the MPM

In the MPM, a material point $p_l \in \boldsymbol{p}$, with $l = 1, 2, ..., n_{mp}$ where $n_{mp}$ is the number of material points, is defined as a location $X_{p_l} \in \boldsymbol{\Omega}$ at which material quantities like mass, velocity, deformation gradient, and stress are calculated.

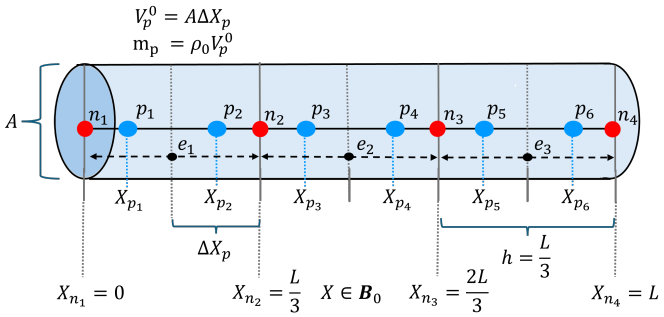



FIG. 5. A visual example for the initialization of the MPM. The background grid is initialized with three grid cells and linear shape functions so that each grid cell (i.e., element in the FEM) has two local nodes, denoted with red circles. Because there are three grid cells, the characteristic length of the background grid (size of each cell) is again $h = L/3$. Material points, denoted with blue circles, (related to the integration points of the FEM) are initialized in this project with two particles per cell such that they are evenly spaced throughout the material reference domain $\boldsymbol{B}_0$. Each material point $p_l$ represents a volume $V_{p_l} = A\Delta X_{p_l}$ of the material with material quantities concentrated at the the point in the center of the volume (mass, velocity, deformation gradient, stress, etc.). The mass of the material represented by the material point is given by $m_p = \rho_0 V_{p_l}^0$.

Material points are closely related to the Gauss-Legendre quadrature or integration points as in the normal FEM, and may be initialized as such. However, due to the difference in how motion is propagated in time in an MPM as compared to the FEM, particularly for MPM's with computations done outside of the reference configuration the material, points at a certain time $t$ will not generally be collocated at the optimal locations (i.e., Gauss-Legendre integration points). This means that, overall, the MPM has reduced solution accuracy and convergence rate for increasing the number of elements and material points as compared with FEM. The Total Lagrangian MPM can avoid this issue entirely as all integrations are done in the initial (reference) configuration. It is typical to initialize material points using a specified number of particles per grid cell $n_{ppc}$. The total number of material points for this problem may be calculated as $n_{mp} = n_{el} n_{ppc}$. A common number of particles per cell is to set $n_{ppc} = 2$. For simplicity, in this project the material points $p_l$ are initialized with $n_{ppc} = 2$ such that the particles are evenly spaced in the reference configuration $X_{p_l}$ and represent equal portions of the bar's volume, described in more detail below.

In the MPM, the material density $\rho_0$ is written as the sum of the mass for a discrete set of material points [1]. In particular, for a finite number of $n_{mp}$ material points, define reference positions $\boldsymbol{X}_p = \{X_{p_l}\} \subset \boldsymbol{\Omega}_0$. Then, in deriving the MPM, we replace the constant reference density in the weak form statement of Eq. 17 of $G = 0$ with the following density field (note that the units of $\delta(X)$ are 1/length):

$$\rho_0'(X) = \sum_{l=1}^{n_{mp}} \frac{m_{p_l}}{A} \delta(X - X_{p_l}), \tag{29}$$

where $m_p$ corresponds to the mass of a material point $p$, subject to the constraint that the total mass of the bar, $M$, is given by

$$\begin{aligned} M = \rho_0 V &= \int_{\boldsymbol{B}_0} \rho_0 A dX = \int_{\boldsymbol{B}_0} \rho_0'(X) A dX \\ &= \int_{\boldsymbol{B}_0} dX A \left[ \sum_{l=1}^{n_{mp}} \frac{m_{p_l}}{A} \delta(X - X_{p_l}) \right] \\ &= \sum_{l=1}^{n_{mp}} m_{p_l} \int_{\boldsymbol{B}_0} dX \delta(X - X_{p_l}) = \sum_{p=1}^{n_{mp}} m_{p_l}. \end{aligned} \tag{30}$$

In this formulation, each material point is implicitly assigned a reference volume $V_p^0$ where $m_p = \rho_0 V_p$, with $\rho_0$ again the density of the material in the reference configuration. In particular, $V_p^0 = A\Delta X_p$, where $\Delta X_p$ is the length of an interval on $\boldsymbol{B}_0$ representing a portion of the bar in 1D. Fig. 5 shows an example of how the MPM is initialized with material points and background grid for the case of equally-spaced particles with three grid cells (i.e., elements) and two particles per cell.

### F. Computation of $G^h$ in the TLMPM

In this section, we explicitly compute $G^h$ in the TLMPM. From the definition of $G$ in Eq. (15) and (16), using the definition of the Bubnov-Galerkin form from Eq. (18), and the material point density field defined in Eq. (29), we obtain

$$\begin{aligned}
G_1^{\mathrm{INT},h} &= \int_{\boldsymbol{B}_0^h} dX \left[ -\frac{\partial w^h(X)}{\partial X} P_{11}(X,t) \right], \\
G_1^{\mathrm{EXT},h} &= \int_{\boldsymbol{B}_0^h} dX \left[ w^h(X) b(t) \rho_0'(X) \right], \\
G_2^{\mathrm{INT},h} &= \int_{\boldsymbol{B}_0^h} dX \left[ -w^h(X) \rho_0'(X) a^h(X,t) \right], \\
G_2^{\mathrm{EXT},h} &= \int_{\boldsymbol{B}_0^h} dX \left[ w^h(X) \tau(t) \delta(X-L) \right], \\
G^h &= G_1^{\mathrm{INT},h} + G_2^{\mathrm{INT},h} + G_3^{\mathrm{INT},h} + G^{\mathrm{EXT},h} = 0.
\end{aligned}$$

#### 1. *Particle-to-Grid*

From the definitions of $x^h(X)$ and $w^h(X)$ in Eqs. (23) and (24) we have

$$\begin{aligned}
w^h(X) &= \boldsymbol{w}_n \cdot \boldsymbol{N}(X), \\
u^h(X) &= \boldsymbol{N}(X) \cdot \boldsymbol{d}_n(t),
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial w^h(X)}{\partial X} &= \boldsymbol{w}_n \cdot \boldsymbol{B}(X), \\
\ddot{a}^h(X,t) &= \ddot{x}^h(X,t) = \boldsymbol{N}(X) \cdot \ddot{\boldsymbol{d}}_n(t).
\end{aligned} \tag{31}$$

Because the First-Piola Kirchoff stress $P_{11}$ tensor is computed at the locations $X_{p_l}$ at time $t$ of the material points $\boldsymbol{p}$, we approximate $G_1^{\mathrm{INT}}$ as

$$\begin{aligned}
G_1^{\mathrm{EXT},h} &= \int_{\boldsymbol{B}_0^h} dX \left[ -\frac{\partial w^h(X)}{\partial X} P_{11}(X,t) \right] \\
&\approx \sum_{l=1}^{n_{mp}} \left[ -\frac{w^h(X_{p_l})}{\partial X} P_{11}(X_{p_l},t) \Delta X_{p_l} \right] \\
&= \boldsymbol{w}_n \cdot \sum_{l=1}^{n_{mp}} \left[ -\boldsymbol{B}(X_{p_l}) P_{11}(X_{p_l},t) \Delta X_{p_l} \right].
\end{aligned} \tag{32}$$

Also, since the traction is applied at the boundary $\partial \boldsymbol{B}_0^* = \{L\}$, we apply the force at the rightmost edge of the volume of the last material point, where the material points numbered in increasing order for increasing $X$ (i.e., left-to-right) like is shown in Fig. 5. The boundary at which to apply the traction is determined from the last material point as

$$X^* = X_{p_{n_{mp}}} + \frac{\Delta X_{p_{n_{mp}}}}{2}, \tag{33}$$

so that we may write

$$\begin{aligned}
G_2^{\mathrm{EXT},h} &= \int_{\boldsymbol{B}_0^h} dX \left[ w^h(X) \tau(t) \delta(X-L) \right] \\
&= \int_{\boldsymbol{B}_0^h} dX \left[ w^h(X) \tau(t) \delta(X-X^*) \right] \\
&= w^h(X^*) \tau(t) = \boldsymbol{w}_n \cdot \boldsymbol{N}(X^*) \tau(t).
\end{aligned} \tag{34}$$

For the other two integrals we obtain

$$\begin{aligned} G_2^{\mathrm{INT},h} &= \int_{\boldsymbol{B}_0^h} dX \left[ w^h(X) b(t) \rho_0'(X) \right] \\ &= \int_{\boldsymbol{B}_0^h} dX \left[ w^h(X) b(t) \sum_{p=1}^{n_{mp}} \frac{m_{p_l}}{A} \delta(X - X_{p_l}) \right] \\ &= \sum_{p=1}^{n_{mp}} \left\{ \int_{\boldsymbol{B}_0^h} dX \left[ w^h(X) b(t) \frac{m_p}{A} \delta(X - X_{p_l}) \right] \right\} \\ &= \sum_{p=1}^{n_{mp}} \left[ w^h(X_{p_l}) b(t) \frac{m_{p_l}}{A} \right] = \boldsymbol{w}_n \cdot \sum_{p=1}^{n_{mp}} \left[ \boldsymbol{N}(X_{p_l}) b(t) \frac{m_{p_l}}{A} \right], \end{aligned} \tag{35}$$

and, likewise, if we define the *consistent mass matrix* (see [1]) as

$$\boldsymbol{M} = \sum_{p=1}^{n_{mp}} m_{p_l} \boldsymbol{N}(X_{p_l}) \otimes \boldsymbol{N}(X_{p_l}), \tag{36}$$

where $\otimes$ represents the tensor product, then

$$\begin{aligned} G_2^{\mathrm{INT},h} &= \int_{\boldsymbol{B}_0^h} dX \left[ -w^h(X) \rho_0'(X) a^h(X,t) \right] \\ &= \sum_{p=1}^{n_{mp}} \left[ -w^h(X_{p_l}) \frac{m_{p_l}}{A} a^h(X,t) \right] \\ &= \sum_{p=1}^{n_{mp}} \left[ -(\boldsymbol{w}_n \cdot \boldsymbol{N}(X_{p_l})) \frac{m_{p_l}}{A} (\boldsymbol{N}(X_{p_l}) \cdot \ddot{\boldsymbol{d}}_n(t)) \right] \\ &= \boldsymbol{w}_n \cdot \sum_{p=1}^{n_{mp}} \left[ -\frac{m_{p_l}}{A} \boldsymbol{N}(X_{p_l}) \otimes \boldsymbol{N}(X_{p_l}) \right] \cdot \ddot{\boldsymbol{d}}_n(t), \end{aligned} \tag{37}$$

$$= -\boldsymbol{w}_n \cdot \frac{\boldsymbol{M}}{A} \cdot \ddot{\boldsymbol{d}}_n(t) = -\boldsymbol{w}_n \cdot \frac{\boldsymbol{M}}{A} \cdot \boldsymbol{a}_n(t), \tag{38}$$

where $\boldsymbol{a}_n(t)$ are the nodal accelerations obtained from the nodal displacements, $\boldsymbol{a}_n(t) = \ddot{\boldsymbol{d}}_n(t)$.

Now, define quantities

$$-\boldsymbol{\mathcal{F}}_2^{\mathrm{INT},h}(t) = \boldsymbol{\mathcal{F}}^{\mathrm{NET},h}(t) = \boldsymbol{\mathcal{F}}_1^{\mathrm{INT},h}(t) + \boldsymbol{\mathcal{F}}_1^{\mathrm{EXT},h}(t) + \boldsymbol{F}_2^{\mathrm{EXT},h}(t) \tag{39}$$

where

$$\begin{aligned} \boldsymbol{\mathcal{F}}_1^{\mathrm{INT},h}(t) &= \sum_{l=1}^{n_{mp}} -\boldsymbol{B}(X_{p_l}) P_{11}(X_{p_l}, t) V_{p_l}^0, \\ \boldsymbol{\mathcal{F}}_1^{\mathrm{EXT},h}(t) &= \sum_{l=1}^{n_{mp}} \boldsymbol{N}(X_{p_l}) m_{p_l} b(t), \\ \boldsymbol{\mathcal{F}}_2^{\mathrm{INT},h}(t) &= -\boldsymbol{M} \cdot \boldsymbol{a}_n(t), \\ \boldsymbol{\mathcal{F}}_2^{\mathrm{EXT},h}(t) &= \boldsymbol{N}(X^*) A \tau(t), \end{aligned} \tag{40}$$

such that

$$AG^h = \boldsymbol{w}_n \cdot (\boldsymbol{\mathcal{F}}^{\mathrm{NET},h}(t) + \boldsymbol{\mathcal{F}}_2^{\mathrm{INT},h}(t)) = 0 \tag{41}$$

With these results, we can rewrite Eq. 31 to obtain a differential equation for nodal displacements in the familiar form of a force-balance equation *on the background grid*, satisfying the same boundary conditions and initial conditions on $\boldsymbol{d}_n(t)$ as in Eq. (3):

$$\boldsymbol{M} \cdot \boldsymbol{a}_n(t) = \boldsymbol{\mathcal{F}}^{\mathrm{NET},h}(t) = \boldsymbol{\mathcal{F}}^{\mathrm{INT},h}(t) + \boldsymbol{\mathcal{F}}^{\mathrm{EXT},h}(t), \tag{42}$$

where

$$\boldsymbol{\mathcal{F}}^{\mathrm{EXT},h} = \boldsymbol{\mathcal{F}}_1^{\mathrm{EXT},h} + \boldsymbol{\mathcal{F}}_2^{\mathrm{EXT},h}, \tag{43}$$

$$\boldsymbol{\mathcal{F}}^{\mathrm{INT},h} = \boldsymbol{\mathcal{F}}_1^{\mathrm{INT},h}. \tag{44}$$

Written in this manner, notice that the TLMPM (and MPM in general) seems to become simply a matter of solving the force-balance equations to get nodal displacements $\boldsymbol{d}_n$, and that the weighting function $\boldsymbol{w}^h(X)$ has entirely disappeared. However, Eq. (42) does not explicitly show how the quantities at the material points are updated. Furthermore, the deformation gradient needs to also be computed for each material point $p_l$ in order to compute the corresponding stress $P_{11}(X_{p_l}, t)$. What this means is that solving Eq. (42) is the first step of the MPM known as *particles-to-grid* [1, 2], where the material point quantities are extrapolated to the grid nodes using the finite element shape functions in Eqs. (23) and (24) in order to solve the force-balance equation on the background grid. The following section discusses how material point quantities are updated and computed using the force balance equation from the background grid in the step known as *grid-to-particles*.

### *2. Grid-to-Particles*

As stated, the objective of the *grid-to-particles* step in the MPM is to interpolate quantities computed on the grid back to the particles in order to update the material quantities corresponding to the material points $\boldsymbol{p}$.

First, the deformation gradient at material point $p_l$, $F_{11}(X_{p_l})$, used in the calculation of $\boldsymbol{\mathcal{F}}_1^{\mathrm{INT},h}$ from Eq. (40) is computed using the nodal displacements $\boldsymbol{d}_n(t)$ as

$$F_{11}(X_{p_l}, t) = \frac{\partial u^h(X_{p_l}, t)}{\partial X} - 1 = \boldsymbol{B}(X_{p_l}) \cdot \boldsymbol{d}_n(t) - 1. \tag{45}$$

or with

$$\dot{F}_{11}(X_{p_l}, t) = \frac{\partial F_{11}(X_{p_l}, t)}{\partial t} = \boldsymbol{B}(X_{pl}) \cdot \boldsymbol{v}_n(t) \tag{46}$$

subject to the initial condition $F_{11}(X_{p_l}, t) = 1$.

In order to update the current position $x_{p_l}(t)$ and velocity $v_{p_l}(t)$ of each particle in the MPM, as well as obtain the nodal velocities $\boldsymbol{v}_n(t)$ to calculate the deformation gradient, the momentum of each particle is extrapolated to the grid to obtain nodal momenta $\boldsymbol{\phi}_n$ with

$$\boldsymbol{\phi}_n(t) = \sum_{l=1}^{n_{mp}} \boldsymbol{N}(X_{p_l}) m_{p_l} v_{p_l}(t), \tag{47}$$

where prescribed boundary conditions are applied so that $\Phi_{n_a}(t) = 0$ if $n_a$ is a node on the boundary $\partial \boldsymbol{B}_0^u$. The change in nodal momenta at time $t$ is calculated from the force-balance equation in Eq. (42) by taking the material time derivative of the nodal momenta, noting that $\boldsymbol{N} \otimes \boldsymbol{N} \cdot \boldsymbol{a} = \boldsymbol{N}(\boldsymbol{N} \cdot \boldsymbol{a})$ to get

$$\frac{D}{Dt} \boldsymbol{\Phi}_n(t) = \boldsymbol{M} \cdot \boldsymbol{a}_n(t) \tag{48}$$

$$\implies \sum_{l=1}^{n_{mp}} \boldsymbol{N}(X_{p_l}) m_{p_l} \dot{v}_{p_l}(t) = \sum_{p=1}^{n_{mp}} m_{p_l} \boldsymbol{N}(X_{p_l}) \otimes \boldsymbol{N}(X_{p_l}) \cdot \boldsymbol{a}_n(t) = \sum_{p=1}^{n_{mp}} m_{p_l} \boldsymbol{N}(X_{p_l}) (\boldsymbol{N}(X_{p_l}) \cdot \boldsymbol{a}_n(t)), \tag{49}$$

so that $v_{p_l}(t)$ may be solved for as

$$\dot{v}_{p_l}(t) = \boldsymbol{N}(X_{p_l}) \cdot \boldsymbol{a}_n(t) \tag{50}$$

where the displacement $u_{p_l}(t)$ is given by

$$u_{p_l}(t) = x_{p_l}(t) - X_{p_l} \tag{51}$$

subject to the initial conditions for velocity and displacement as in Eq. (3). By defining nodal mass $\boldsymbol{m}_n$ as

$$\boldsymbol{m}_n = \sum_{l=1}^{n_{mp}} \boldsymbol{N}(X_{p_l}) m_{p_l}, \tag{52}$$

may be obtained as

$$\boldsymbol{v}_n(t) = \boldsymbol{\phi}_n(t) \oslash \boldsymbol{m}_n, \tag{53}$$

where $\oslash$ denotes the Hadamard (elementwise) division operator; that is $v_{n_a}(t)$ is computed as

$$v_{n_a}(t) = \frac{\phi_{n_a}(t)}{m_{n_a}}. \tag{54}$$

With this, the position update for the particle is also obtained as

$$\dot{x}_{p_l}(t) = \boldsymbol{N}(X_{p_l}) \cdot \boldsymbol{v}_n(t) \tag{55}$$

### G. Problem Statement for the TLMPM

We are now ready to define the problem statement for the TLMPM. To do so, first define the background grid $\boldsymbol{\Gamma}$ by the tuple

$$\boldsymbol{\Gamma}^h = (\boldsymbol{e}, \boldsymbol{n}, \boldsymbol{\Omega}^h, \boldsymbol{X}_n, \boldsymbol{N}, \boldsymbol{B}) \tag{56}$$

where $\boldsymbol{e}$ are the element numbers or IDs, $\boldsymbol{n}$ are the global node IDs, $\boldsymbol{X}_n$ are the locations of the global nodes in the configuration domain $\boldsymbol{\Omega}^h$, and $\boldsymbol{N} \subset H^1$ are the shape functions, and $\boldsymbol{B}$ are the gradients of the shape functions.

Also (re)define material points by the tuple

$$\boldsymbol{\mathcal{P}} = (\boldsymbol{p}, \boldsymbol{X}_p, \boldsymbol{x}_p, \boldsymbol{v}_p, \boldsymbol{u}_p, \boldsymbol{V}_p^0, \boldsymbol{m}_p, \boldsymbol{F}_p, \boldsymbol{P}_p), \tag{57}$$

where $\boldsymbol{p}$ are the material point IDs, $\boldsymbol{X}_p$ are the corresponding locations in the reference configuration, $\boldsymbol{x}_p$ are the locations of the material points in the current configuration, $\boldsymbol{v}_p$ are the material point velocities, $\boldsymbol{u}_p$ the displacements, $\boldsymbol{V}_p^0$ the volumes, $\boldsymbol{m}_p$ the masses, $\boldsymbol{F}_p$ is the deformation gradient, and $\boldsymbol{P}_p$ the First-Piola Kirchoff stress tensors for each material point. With these definitions, the problem statement of the TLMPM may be formulated as follows:

$$\text{(TLMPM)} \quad \begin{cases} \text{Find the displacement } \boldsymbol{u}_p(t) \in \boldsymbol{\mathcal{U}}_p^h \text{ of material points } \boldsymbol{\mathcal{P}} \\ \text{with respect to a background grid } \boldsymbol{\Gamma}^h \text{ for a specified continuum model} \\ \text{satisfying the following coupled differential equations:} \\ \quad \boldsymbol{M} \cdot \boldsymbol{a}_n(t) = \boldsymbol{\mathcal{F}}^{\text{NET},h}(t) = \boldsymbol{\mathcal{F}}^{\text{INT},h}(t) + \boldsymbol{\mathcal{F}}^{\text{EXT},h}(t) & \text{(Force-Balance Eqn.)} \\ \quad \dot{\boldsymbol{F}}_p(\boldsymbol{X}_p, t) = \boldsymbol{B}(\boldsymbol{X}_p) \cdot \boldsymbol{v}_n(t) \text{ with } \boldsymbol{F}(\boldsymbol{X}p, t=0) = \boldsymbol{1}, & \text{(Def. Gradient)} \\ \quad x_{n_a}(t) = g_{n_a}(t) \text{ if } X_{n_a} \in \partial \boldsymbol{B}_0^u, & \text{(Prescribed Displacement BCs)} \\ \quad \dot{\boldsymbol{v}}_p(t) = \boldsymbol{N}(\boldsymbol{X}_p) \cdot \boldsymbol{a}_n(t), & \text{(Velocity)} \\ \quad \dot{\boldsymbol{x}}_p(t) = \boldsymbol{N}(\boldsymbol{X}_{p_l}) \cdot \boldsymbol{v}_n(t), & \text{(Current Position)} \\ \quad \boldsymbol{u}_p(t) = \boldsymbol{x}_p(t) - \boldsymbol{X}_p, & \text{(Displacement)} \\ \quad \boldsymbol{u}_p(t=0) = \boldsymbol{0}, & \text{(Initial Displacement)} \\ \quad \boldsymbol{v}_p(t=0) = \boldsymbol{v}_p^0, & \text{(Initial Velocity)} \end{cases} \tag{58}$$

where the set of all displacement configurations $\boldsymbol{\mathcal{U}}_p$ for material points $\boldsymbol{\mathcal{P}}$ is defined as

$$\boldsymbol{\mathcal{U}}_p^h = \{\boldsymbol{u}_p(t) \mid \boldsymbol{u}_p(t) : [0, T] \mapsto \mathbb{R}^{n_{mp}}\}. \tag{59}$$

Notice that the problem formulation for the TLMPM is strikingly similar to the original strong form in Eq. (3), but with the major difference being that displacements $\boldsymbol{u}_p(t)$ no longer form a continuum over the configuration space $\boldsymbol{B}_0$ because $n_{mp} = |\boldsymbol{\mathcal{P}}|$ is finite. Furthermore, as formulated, Eq. (58) may be also be used to describe the motion of a material for problems in 2D or 3D, not just problems in 1D. Also notice that prescribed force boundary conditions enter the solution in $\boldsymbol{\mathcal{F}}^{EXT,h}(t)$.

## III. ALGORITHM

In this section I describe the general TLMPM algorithm implementing Eq. 58 as well as additional useful algorithms for the computation of auxiliary quantities like nodal quantities, strain tensors, and energy (strain and kinetic).

### A. A general TLMPM Algorithm

Implementing the TLMPM as described in Eq. (58) is done with a time-marching algorithm where at each time step a four-fold process occurs:

1. **(Particles-to-grid):** Material quantities are mapped to the grid for material points in order to solve the Force-Balance equation from Eq. (42).
2. **(Momentum mapping):** The particle momenta are mapped to the grid nodes in order to solve for the updated momenta using the resultant of the sum of forces. Dirchlet boundary conditions are imposed for the grid nodes with prescribed displacement. In this step I use the double-momentum mapping Newmark explicit time-integration scheme with parameter $0 < \alpha < 1$ described in Nguyen 2023 [1] to obtain updated particle velocities. I set $\alpha = 0.5$ in the Newmark scheme to obtain an explicit central-difference scheme in updating particle velocities.
3. **(Grid-to-particles)** Material quantities are updated by mapping grid quantities back to the material points.
4. **(Grid Reset and step time forward)**: Grid quantities are reset, the time is incremented with $t \leftarrow t + dt$, and this four-step cycle repeats until $t = T$. The time step used should be selected using the Courant-Friedrichs-Lewy (CFL) condition: $dt(c/h) = 1$, where $c$ is the wavespeed for the material (see Eq. 2) and $h$ is the characteristic length of the background grid. Numerically, my implementation seems to exhibit the most stability when the size of the timestep is defines as a fraction of the CFL condition, with $dt \ll \frac{1}{c}\left(\frac{h}{n_{ppc}}\right)$ .

As the algorithm developed in this report is an explicit dynamic code, note that all updated quantities depend solely on previously calculated quantities so there is no need for a nonlinear solver like Newton-Raphson. However, in the case of an implicit MPM, then such a nonlinear solver would be required. A detailed description of the algorithm described above to solve Eq. 58 is provided in Alg. 1. Note that Alg. 1 is general, and may be used to implement the TLMPM in 2D and 3D as well [1]. Also, note that the parameter $\boldsymbol{\theta}$ of Alg. 1 is the collection of material parameters. In the 1D axially-loaded bar, the parameters to the method are $\boldsymbol{\theta} = \{E, \nu, L, A, \rho, \tau(t), b(t)\}$ and $n_{el} \rightarrow n_{el}$, $n_{en}$, and $n_{ppc}$, each of which are defined in Sec. II.

### B. Calculation of auxiliary quantities and extrapolation of particle quantities to the grid

Verifying the solution obtained from implementing Eq. 58 with Alg. 1 typically requires the computation of auxiliary quantities like strain, energy and may require the extrapolation of material point quantities to the grid.

#### *1. Extrapolation of material quantities to grid*

Extrapolation of material point quantities to the grid is done in the same manner that nodal velocities were computed on the grid: by performing a weighted sum over particle quantities, where the particle quantities are weighted by the particle's mass (see Eq. 53). In particular, extrapolating a material quantity $\boldsymbol{Q}_p$ to a nodal quantity $\boldsymbol{Q}_n$ can be done as described in Alg. 2.

#### *2. Computation of Lagrangian, Eulerian, and Hencky Strain*

Strain quantities like Eulerian, Lagrangian, and Hencky (logarithmic) strain tensor s may also be calculated in the MPM at each material material point by making use of the computated deformation gradients. Recall the definitions of Eulerian strain $\boldsymbol{e}$ and Lagrangian strain $\boldsymbol{E}$ tensors as

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - 1), \tag{60}$$

$$\boldsymbol{e} = \frac{1}{2}(1 - \boldsymbol{b}^{-1}), \tag{61}$$

where $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$ is the right Cauchy-Green tensor and $\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^T$ is the left Cauchy-Green tensor. The hencky strain is computed by obtaining the spectral decomposition of $\boldsymbol{b}$ as described in Holpzafel [5] where

$$\boldsymbol{b} = \sum_{\alpha=1}^{3} \lambda_\alpha^2 \hat{\boldsymbol{n}}_\alpha \otimes \hat{\boldsymbol{n}}_\alpha. \tag{62}$$

Then, define $\boldsymbol{v} = \sqrt{\boldsymbol{b}}$ so that

$$\boldsymbol{v} = \sum_{\alpha=1}^{3} \lambda_\alpha \hat{\boldsymbol{n}}_\alpha \otimes \hat{\boldsymbol{n}}_\alpha. \tag{63}$$

Then the Hencky (logarithmic) strain is computed as

$$\boldsymbol{e}^h = \ln(\boldsymbol{v}). \tag{64}$$

**Algorithm 1** Algorithm to solve the TLMPM problem formulated in Eq. (58) (also described in [1]). The parameters to this method are $\boldsymbol{\theta}$ representing material parameters, number of elements per direction $\boldsymbol{n}_{en}$, number of element nodes per direction $\boldsymbol{n}_{en}$, and number of particles per cell direction $\boldsymbol{n}_{ppc}$, timestep $dt$, and total time $T$. Note that this algorithm as described can be used in problems in 1D, 2D, or 3D. Also note that the timestep $dt$ should be selected so that $dt \ll \frac{h}{c}$ as per the CFL condition.

**procedure** TOTAL LAGRANGIAN MPM ($\boldsymbol{\theta}, \boldsymbol{n}_{el}, \boldsymbol{n}_{en}, \boldsymbol{n}_{ppc}, dt, T, \alpha$)
  **Initialization**
    Define background grid $\boldsymbol{\Gamma} \leftarrow (\boldsymbol{e}, \boldsymbol{n}, \boldsymbol{\Omega}^h, \boldsymbol{X}_n, \boldsymbol{N}^n, \boldsymbol{B}^n)$
    Initialize materal points $\boldsymbol{\mathcal{P}} \leftarrow (\boldsymbol{p}, \boldsymbol{X}_p, \boldsymbol{x}_p, \boldsymbol{v}_p, \boldsymbol{u}_p, \boldsymbol{V}_p^0, \boldsymbol{m}_p, \boldsymbol{F}_{11,p}, \boldsymbol{P}_{11,p})$ on $\boldsymbol{\Gamma}$
    Compute nodal mass $\boldsymbol{m}_n \leftarrow \sum_{l=1}^{n_{mp}} \boldsymbol{N}(X_{p_l}) m_{p_l}$
    Set time $t_s \leftarrow 0$
  **end**
  **while** $t_s \leq T$ **do**
    Compute $t_{s+1} \leftarrow t_s + dt$
    **Reset Grid Quantities:** Set $\boldsymbol{\phi}_n, \boldsymbol{F}^{\text{INT}}, \boldsymbol{F}^{\text{EXT}} \leftarrow \boldsymbol{0}$
    **Mapping Particles-To-Grid (P2G)**
      Compute nodal momenta $\boldsymbol{\Phi}_n(t_s) \leftarrow \sum_{l=1}^{n_{mp}} \boldsymbol{N}(X_{p_l}) m_{p_l} v_{p_l}(t_s)$
      Compute internal force $\boldsymbol{\mathcal{F}}^{\text{INT}}(t_s) \leftarrow -\sum_{l=1}^{n_{mp}} \boldsymbol{B}(X_{p_l}) \cdot \boldsymbol{P}(X_{p_l}, t_s) V_{p_l}^0$
      Compute external force $\boldsymbol{\mathcal{F}}^{\text{EXT}}(t_s)$
      Compute resultant force $\boldsymbol{\mathcal{F}}^{\text{NET}}(t_s) \leftarrow \boldsymbol{\mathcal{F}}^{\text{INT}}(t_s) + \boldsymbol{\mathcal{F}}^{\text{EXT}}(t_s)$
    **end**
    **Update velocities (Momenta double mapping)**
      Compute 1st grid momenta update from result force $\tilde{\boldsymbol{\Phi}}(t_{s+1}) \leftarrow \boldsymbol{\Phi}_{\boldsymbol{n}} + dt\boldsymbol{\mathcal{F}}^{\text{NET}}$ ▷ First mapping
      Apply prescribed displacement BCs $\Phi_{n_a}(t_s) = \tilde{\Phi}_{n_a}(t_{s+1}) = 0$ if $X_{n_a} \in \partial \boldsymbol{B}_0^u$
      Compute time $t$ nodal velocities $\boldsymbol{v}_n(t_s) = \boldsymbol{\phi}_n(t_s) \oslash \boldsymbol{m}_n$
      Compute time $t_{s+1}$ nodal velocities $\tilde{\boldsymbol{v}}_n(t_{s+1}) = \tilde{\boldsymbol{\phi}}_n(t_s) \oslash \boldsymbol{m}_n$
      Map momenta at time $t$ to particles $\boldsymbol{v}_g(t_s) \leftarrow \boldsymbol{N}(\boldsymbol{X}_p) \cdot \boldsymbol{v}_n(t_k)$
      Update $\tilde{\boldsymbol{v}}_g(t_{s+1}) \leftarrow \boldsymbol{N}(\boldsymbol{X}_p) \cdot \tilde{\boldsymbol{v}}_n(t_{s+1})$
      Update particle velocity $\boldsymbol{v}_p(t_{s+1}) = \alpha\left[\boldsymbol{v}_p(t_s) + \tilde{\boldsymbol{v}}_g(t+dt) - \boldsymbol{v}_g(t_s)\right] + (1-\alpha)\tilde{\boldsymbol{v}}_g(t_{s+1})$ ▷ Newmark integration
      Compute grid momenta $\boldsymbol{\Phi}_n(t_{s+1}) \leftarrow \sum_{l=1}^{n_{mp}} N_{X_{p_l}} m_{p_l} v_{p_l}(t_{s+1})$ ▷ Second mapping to grid velocities
      Fix displacement BCs $\Phi_{n_a}(t_{s+1}) = 0$ if $X_{n_a} \in \partial \boldsymbol{B}_0^u$
      Obtain nodal velocities $\boldsymbol{v}_n(t_{s+1}) \leftarrow \boldsymbol{\Phi}_n(t_{s+1}) \oslash \boldsymbol{m}_n$
    **end**
    **Mapping Grid-to-Particles (G2P)**
      Update material position $\boldsymbol{x}_p(t_{s+1}) \leftarrow \boldsymbol{x}_p(t_s) + dt\boldsymbol{v}_p(t_{s+1})$
      Update material displacement $\boldsymbol{u}_p(t_{s+1}) \leftarrow \boldsymbol{x}_p(t_{s+1}) - X_p$
      Compute deformation rate $\dot{\boldsymbol{F}}_p(t_{s+1}) \leftarrow \boldsymbol{B}(\boldsymbol{X}_p) \cdot \boldsymbol{v}_n(t_{s+1})$
      Update deformation gradient $\boldsymbol{F}_p(t_{s+1}) \leftarrow \boldsymbol{F}_p(t_s) + dt\dot{\boldsymbol{F}}_p(t_{s+1})$
      Compute spatial velocity gradient $\boldsymbol{l}_p(t_{s+1}) \leftarrow \dot{\boldsymbol{F}}\boldsymbol{F}^{-1}(t_{s+1})$
      Compute 2nd PK stress $\boldsymbol{S}_p(t_{s+1}) \leftarrow \boldsymbol{S}(\boldsymbol{F}_p(t_{s+1}))$ ▷ From Consitutive Assumption
      Compute 1st PK stress $\boldsymbol{P}_p(t_{s+1}) \leftarrow \boldsymbol{F}_p(t_{s+1})\boldsymbol{S}_p(t_{s+1})$
    **end**
    $t_s \leftarrow t_{s+1}$ ▷ March time forward one step
  **end while**
**end procedure**

**Algorithm 2** An algorithm describing the extrapolation of a material point quantity $\boldsymbol{Q}_p$ to the grid to obtain $\boldsymbol{Q}_n$.

**function** MAP-TO-GRID($\boldsymbol{Q}_p, \boldsymbol{\mathcal{P}}, \boldsymbol{\Gamma}^h$)
  $\boldsymbol{m}_n \leftarrow \sum_{l=1}^{n_{mp}} \boldsymbol{N}(X_{p_l}) m_{p_l}$
  $\tilde{\boldsymbol{Q}}_n \leftarrow \sum_{l=1}^{n_{mp}} \boldsymbol{N}(X_{p_l}) m_{p_l} Q_{p_l}$
  $\boldsymbol{Q}_n \leftarrow \tilde{\boldsymbol{Q}}_n \oslash \boldsymbol{m}_n$
  **Return** $\boldsymbol{Q}_n$
**end function**

*3. Energy Computation*

The total energy of the material may be computed by summing kinetic and potential contributions, where the potential contribution is obtained via integrating the strain energy in Eq. 4 over the configuration domain of the MPM and the total kinetic energy is calculated as the sum of the kinetic energy of each of the particles. Specifically, if $T$ is the kinetic energy and $U$ the potential energy, then calculate $T$ as

$$T = \sum_{l=1}^{n_{mp}} m_{p_l} v_{p_l}^2. \tag{65}$$

The strain energy is obtained by integrating the strain energy density in the reference configuration over the domain $\boldsymbol{B}_0$. In the case of a Neo-Hookean elasticity model, the strain energy is given in Eq. 4 where

$$\rho_0 \psi(\boldsymbol{C}) = \frac{1}{2}\lambda(\ln J)^2 - \mu \ln(J) + \frac{1}{2}\mu(\operatorname{Tr}\boldsymbol{C} - 3). \tag{66}$$

Note that $J = \det F = \frac{1}{2}\det \boldsymbol{C}$. Then, making use of the MPM reference density field in Eq. 29, we obtain in 1D the strain energy as

$$\begin{aligned} U &= \int_{\boldsymbol{B}_0} \rho_0 \psi(\boldsymbol{C}, t) dV = \int_{\boldsymbol{B}_0} \rho'(X) \psi A dX \\ &= \int_{\boldsymbol{B}_0} \left[ \sum_{l=1}^{n_{mp}} \frac{m_{p_l}}{A} \delta(X - X_{p_l}) \psi(X, \boldsymbol{C}, t) \right] (A dX) \\ &= \sum_{l=1}^{n_{mp}} m_{p_l} \psi(X_{p_l}, \boldsymbol{C}, t) = \sum_{l=1}^{n_{mp}} m_{p_l} \psi(X_{p_l}, F_{11,p_l}, t) = \sum_{l=1}^{n_{mp}} U_{p_l}, \end{aligned} \tag{67}$$

with $U_{p_l}$ defined as

$$U_{p_l} \equiv m_{p_l} \psi(F_{11,p_l}, t) = V_{p_l}^0 \rho_0 \psi(F_{11,p_l}, t) = V_{p_l}^0 \left[ \frac{1}{2}\lambda(\ln F_{11,p_k})^2 - \mu \ln(F_{11,p_l}) + \frac{1}{2}\mu(F_{11,p_l}^2 - 1) \right]. \tag{68}$$

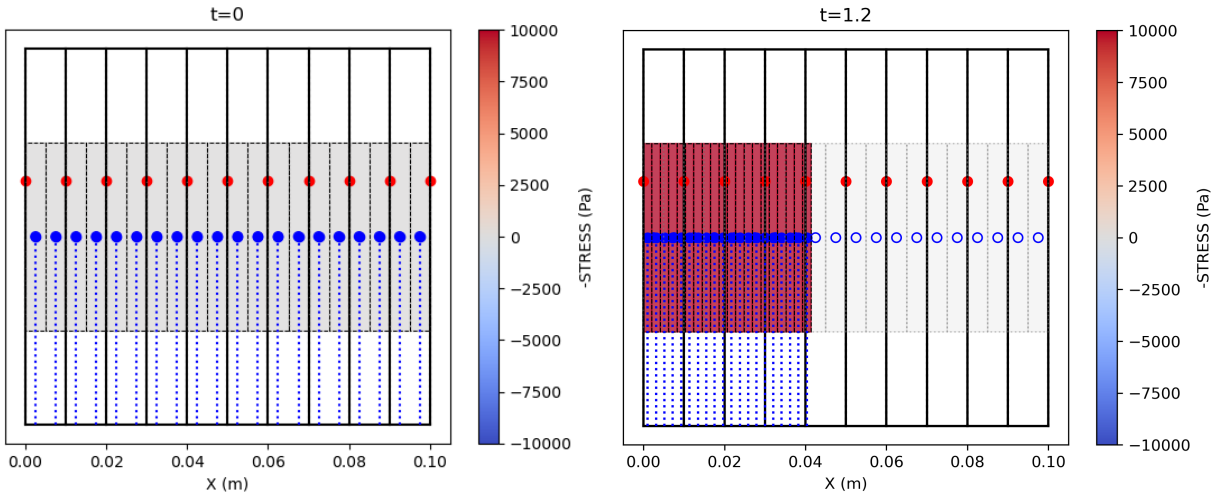
## IV. VERIFICATION

Verification of the TLMPM algorithm developed in III is done in this section for three cases. First, the results from the TLMPM are against results from an FEM code in Sec. IV A for a problem involving the quasi-static large compression of a column. Then, the results from TLMPM are compared to an analytical solution for the case of a step-loaded bar with prescribed displacement on one end in Sec. IV B. Finally, the energy behavior of the TLMPM is checked in Sec. IV C.

### A. Quasi-Static Column Compression

The first verification problem for the MPM code and algorithm developed in this report considered is the quasi-static column compression problem defined with the parameters in Eq. (69). In this problem, a column is compressed with a linearly increasing traction until $t = 1.0$ (pseudo-time) and then held until $t = 1.1$.

The results obtained with the MPM code are compared with the results from an FEM code (like Abaqus/CAE) where both codes are initialized using linear shape functions, 10 elements, and two integration points/particles per




(a) Initial Configuration at $t = 0$

(b) Final Configuration at $t = 1.2$

FIG. 6. Graphic showing the initial and final configurations of the TLMPM for the verification study of the quasi-static large deformation column compression. Material points are represented by blue circles and global nodes represented by red circles, as in Fig. 5. The representative volume of each material point is colored with respect to the principal Cauchy stress component $\sigma_{11}$. Plot (a) shows the initial configuration at $t = 0$ (pseudo-time) and plot (b) shows the final configuration at $t = 1.2$.

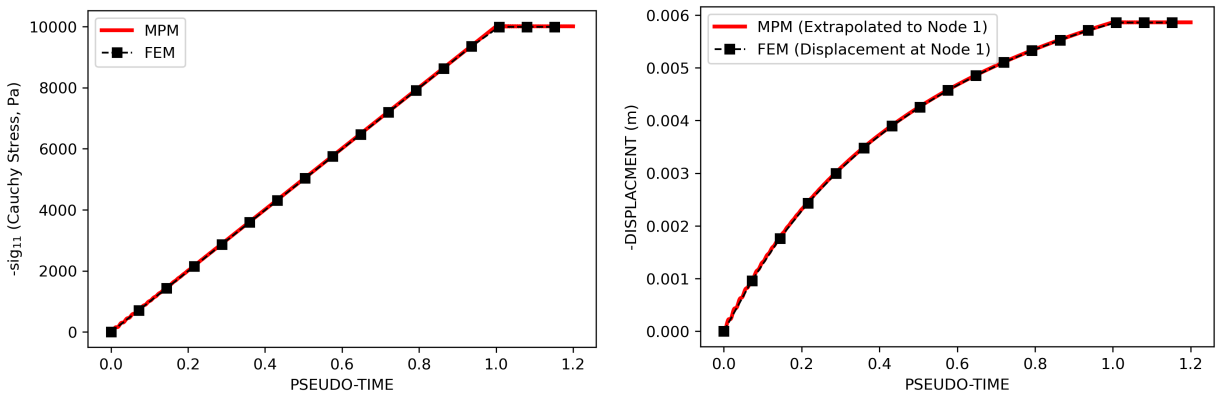
cell. Fig. 6 visualizes the initial and final configurations for the column, with colors representing the magnitude of the Cauchy stress per material point volume. Fig. 7(a) compares the Cauchy stress vs. pseudo-time between the two codes for global node 1 (at reference position $X = L/10 = 0.1m$. Fig. 7(b) shows the displacement at the same node. Fig. 7(c) visualizes the results comparing stress (Second PK stress, Cauchy stress) vs. the strain measures defined in Sec. III. Notice the excellent agreement between the results from the MPM as compared to the results from the FEM code in these figures.

| Quantity | Value | Units |
|---|---|---|
| $E$ | $5 \times 10^3$ | Pa |
| $\nu$ | 0.3 | |
| $\rho$ | 10 | kg/m$^3$ |
| $b(t)$ | $g = -9.81$ | m/s$^2$, gravity |
| $\tau_0$ | $-10^4$ | Pa |
| $\tau(t)$ | $\text{Ramp}(\tau_0, t, T_{\text{ramp}})$ | Pa |
| $\boldsymbol{v}_0$ | 0 | |
| $L$ | 0.1 | m |
| $A$ | $10^{-4}$ | m$^2$ |
| $n_{el}$ | 10 | |
| $n_{en}$ | 2 | |
| $n_{ppc}$ | 2 | |
| $T$ | 1.2 | pseudo-time |
| $T_{\text{ramp}}$ | 1.0 | pseudo-time |
| $dt$ | $0.1(h/c)$ | pseudo-time |
| $\alpha$ | 0.5 | |

(69)

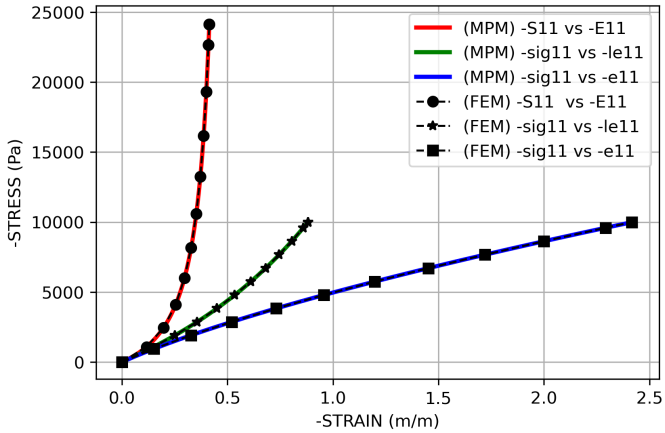
### B. Dynamic Step Loading of a Bar in Uniaxial Stress Compression

The second verification problem considered is the dynamic step loading of a bar at small strain; that is, a bar has a prescribed, constant traction fixed at the free end. Parameters for this problem used in the MPM code are provided in Eq. 71. An analytical solution is found in Eq. (8.71) of Meirovitch 1967 [3], where the displacement at the end of




(a) Stress vs. Pseudo-Time

(b) Displacement vs. Pseudo-Time



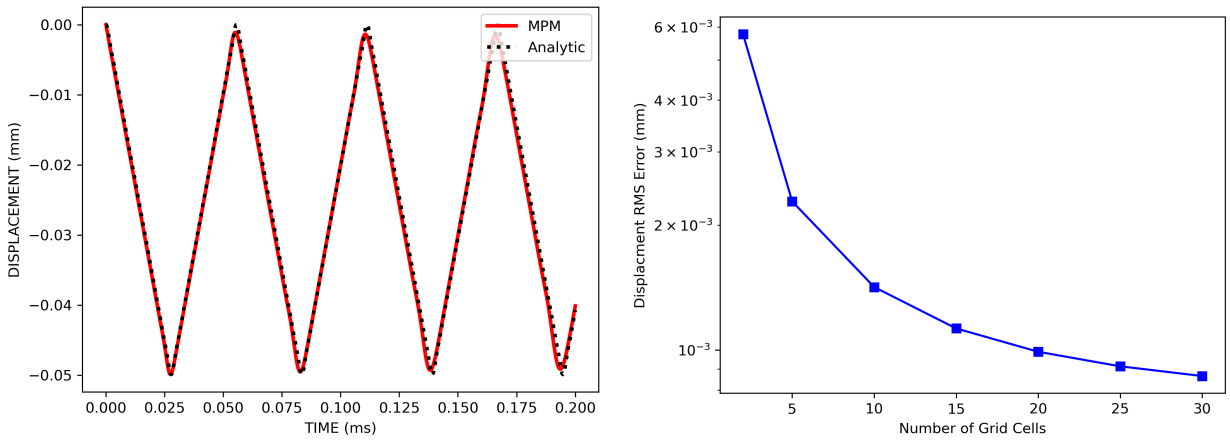

(c) Stress vs. Strain Measures

FIG. 7. Comparisons between the MPM code developed in this project against an implementation of total Lagrangian FEM for the large deformation compression of a column for (a) Stress vs. Pseudo-Time, (b) Displacement vs. Pseudo-Time, (c) Stress vs. Strain measures.

the bar is found as

$$u(x,t) = \frac{8\tau L}{\pi^2 E}\sum_{n=1}^{\infty}\frac{(-1)^{n-1}}{(2n-1)^2}\sin\left[(2n-1)\frac{\pi x}{2L}\right]\left(1-\cos\left[(2n-1)\frac{\pi ct}{2L}\right]\right). \tag{70}$$

The resulting comparison between the MPM code and the analytical solution described above is displayed in Fig. 8(a). A plot of the error convergence between the MPM code and the analytical solution using the root-mean-square error when increasing the number of grid cells is provided in Fig. 8(b). Notice the excellent visual agreement between the displacement results from the MPM code taken from the end of the bar and the analytical solution and convergence in RMS error.




(a) Displacment vs. time

(b) Error convergence

FIG. 8. A comparison between the TLMPM code developed in this project with parameters described in Eq. 71 and the analytical solution in Eq. 70 for the dynamic step loading of a bar with prescribed displacement on one end. Displacement vs. time is shown in (a) and a plot of the root-mean-square error in the displacement for an increasing number of grid cells is provided in (b).

| Quantity | Value | Units |
|---|---|---|
| $E$ | 250 | MPa |
| $\nu$ | 0 | |
| $\rho$ | $1.9355 \times 10^{-3}$ | g/mm$^3$ |
| $b(t)$ | 0 | m/s$^2$, gravity |
| $\tau(t)$ | $-1.27$ | MPa |
| $\boldsymbol{v}_0$ | 0 | |
| $L$ | 5 | mm |
| $A$ | 19.635 | mm$^2$ |
| $n_{el}$ | 32 | |
| $n_{en}$ | 3 | |
| $n_{ppc}$ | 2 | |
| $T$ | 0.2 | ms |
| $dt$ | $0.1(h/c)$ | ms |
| $\alpha$ | 0.5 | |

(71)

### C. Verification of Energy Conservation in the MPM

The last verification case considered is that where the bar is imparted an initial total energy $\mathcal{E}_0$, with no applied forces. In particular, the energy is initially distributed evenly over every particle, so if there are $n_{mp}$ particles, the initial velocity of each particle $v_0 = \sqrt{\frac{2E}{m}}$. The expected behavior for this case is that the total energy at any time $t > 0$ is $E_{tot} \leq T + U$ where $T$ is the kinetic energy and $U$ is the strain energy. Note that it is expected for $\mathcal{E}_{tot}$ to be less than $\mathcal{E}_0$ in general because of errors resulting due to the interpolation step from grid-to-particles and the removal of energy due to the fixed displacement condition, resulting in energy dissipation over time. (A discussion on this topic is found in Nguyen 2023 [1].) The specific parameters used for this problem Eq. 72. The code is run over 0.2 seconds to obtain the energy plot in Fig. 9 using the formulas for kinetic energy as in Eq. 65 and strain energy as in Eq. 67. This plot shows that the TLMPM obeys the expected energy behavior.

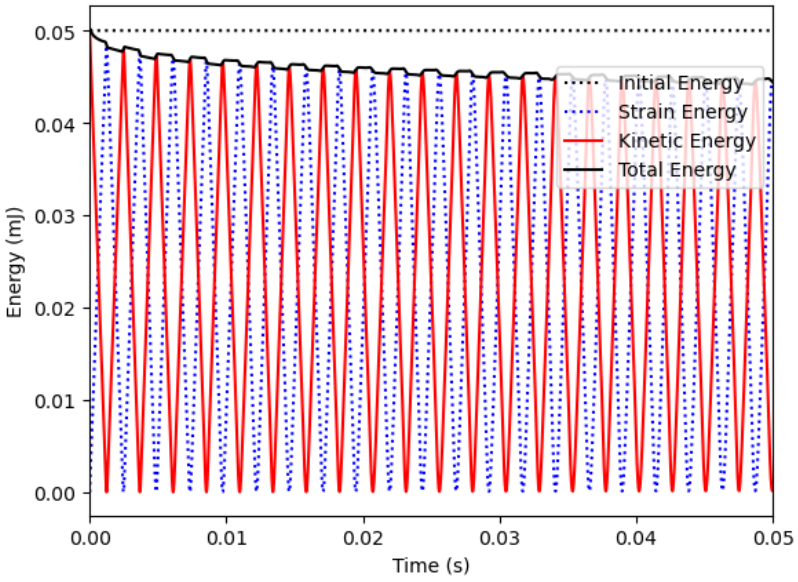



FIG. 9. A plot of the energy against time for the case of a bar with an initial energy of $\mathcal{E}_0 = 0.05$ mJ. Notice that at all times $\mathcal{E}_{tot} \leq \mathcal{E}_0$, showing proper energy behavior as described in Sec. IV C.

| Quantity | Value | Units |
|---|---|---|
| $E$ | $5 \times 10^3$ | Pa |
| $\nu$ | 0.3 | n/a |
| $\rho$ | 10 | g/mm$^3$ |
| $b(t)$ | 0 | m/s$^2$, gravity |
| $\tau(t)$ | 0 | MPa |
| $\mathcal{E}_0$ | 0.05 | mJ |
| $L$ | 0.1 | mm |
| $A$ | $1 \times 10^{-4}$ | mm$^2$ |
| $n_{el}$ | 32 | n/a |
| $n_{en}$ | 3 | n/a |
| $n_{ppc}$ | 3 | n/a |
| $T$ | 0.05 | ms |
| $dt$ | $0.2(h/c)$ | ms |
| $\alpha$ | 0.5 | |

(72)

## V. CONCLUSION

The objective of this report was to implement, in one dimension, a code implementing a total Lagrangian MPM. The theory and algorithm of the TLMPM were developed in Secs. II and III, respectively, for specifically the case of an axially-loaded bar assuming a Neo-Hookean constitutive model. Verification problems in Sec. IV showed excellent agreement between implementation of the MPM as described in Sec. III and results from an analytical solution and the TLFEM. However, it is apparent from checking the energy behavior of the TLMPM as in Fig. 9 that this implementation suffers from significant energy dissipation. Sources of error in the energy arise from the Dirichlet fixed displacement condition removing energy from the problem as well as an accumulation of errors resulting from the mappings to the grid and to the particles.

For future work on this code, I would like to implement an implicit version of the MPM algorithm as well as study the literature to determine if there exists a way mitigate the energy conservation issue, and if not, find out a way to do so. An implicit implementation of the code would also be desirable in order to both speedup the code and improve its stability. This is especially because, due to its explicit dynamic nature, I had to set timesteps as strictly a fraction of the CFL condition so to avoid numerical instability (typically around $10^{-6}$ or $10^{-7}$ time units). As a

result, computations using my code take several minutes to solve for even just 30 finite element cells in the background grid. An implicit code would significantly improve the implementation described here by allowing for much greater numerical stability at much larger time increments.

---


[1] V. P. Nguyen, A. D. Vaucorbeil, and S. Bordas, *The Material Point Method: Theory, Implementations and Applications*, Scientific Computation (Springer International Publishing, Cham, 2023).

[2] A. de Vaucorbeil, V. P. Nguyen, and C. R. Hutchinson, A Total-Lagrangian Material Point Method for solid mechanics problems involving large deformations, Computer Methods in Applied Mechanics and Engineering **360**, 112783 (2020).

[3] L. Meirovitch, *Analytical Methods in Vibrations* (Macmillan, 1967).

[4] R. A. Regueiro, *CVEN 7511 Lecture Notes* (University of Colorado Boulder, 2020).

[5] G. A. Holzapfel, *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*, repr ed. (Wiley, Chichester Weinheim, 2010).

[6] R. de Borst, M. A. Crisfield, J. J. Remmers, and C. V. Verhoosel, *Nonlinear Finite Element Analysis of Solids and Structures*, second edition ed., Wiley Series in Computational Mechanics (Wiley, Chichester, West Sussex, 2012).

[7] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, corr. republication ed. (Dover publ, Mineola (N. Y.), 2000).