XLIII
CILAMCE
21-25, November, 2022
FOZ DO IGUAÇU - PR - BRAZIL

Two-dimensional nonlinear analysis of elastic columns stability using the convected particle domain interpolation material point method

Lucas P. de Souza[1], Marco A. Argenta[1]

[1] *Postgraduate Program in Civil Engineering, Federal University of Paraná*
Politechnique Center of UFPR, 81531-980, Curitiba/Paraná, Brazil
lucasperes.eng@gmail.com, marco.argenta@ufpr.br

Abstract. This paper presents an elastic two-dimensional analysis of columns stability, using an extension of the material point method (MPM), called convected particle domain interpolation (CPDI). As instability phenomena require a large-deformation study, MPM is chosen by its ability to deal with this situation, using a simple and regular cartesian background grid to calculate the spatial gradients and divergences, besides automatically considers explicit dynamics. The CPDI extension is applied in order to facilitate natural boundary conditions treatment and to avoid the cell-crossing error, since the subdomains are explicitly tracked through the analysis and the shape functions have continuous derivatives into these subdomains. A computational routine is implemented using Python 3 language, employing Euler-Gauss explicit time integration and update-stress-last (USL) scheme. A rectangular-section slender column is submitted to 4 different load cases. The results obtained are the load-displacement behavior of the columns and its total mechanical energy time variation, which are validated by comparison with the results produced by classic analytical Euler's theory. It can be verified, after the analyses, that CPDI MPM is able to predict the columns buckling loads and to reproduce the post-critical instability phenomena with a good accuracy. The conservation of energy is achieved in all tests.

Keywords: Buckling, Columns, Convected particle domain interpolation, Material point method, Stability

1 Introduction

Elastic instability, also named buckling, is a highly nonlinear phenomena and occurs when a slender axially loaded column fails due to an abrupt change in the equilibrium state. Because of the high degree of nonlinearity, an accurate numerical analysis of buckling and post-critical behavior requires a method able to solve Continuum Mechanics equilibrium equations taking into account large strains. In these terms, Material Point Method (MPM), firstly published by Sulsky et al. [1], has produced good results when applied to complex problems involving large deformations [2–5], although an study of buckling phenomena has not yet been done. In view of this, this work aims to analyze the elastic stability of a column - buckling loads and post-critical behavior - using a computational algorithm, implemented in Python 3 [6] language and based on an extension of MPM, called convected particle domain interpolation (CPDI MPM), published by Sadeghirad et al. [7, 8], with explicit time integration. Analyses are performed in two dimensions for a clamped-free column submitted to 4 different load cases, considering a linear elastic, isotropic and homogeneous material, plane stress and finite strains.

2 Method

2.1 Governing equations

Derived from the continuum mechanics, equation

$$\int_{\Omega} \mathbf{v} \cdot \rho \frac{d\mathbf{v}}{dt} dV + \int_{\Omega} \boldsymbol{\mathcal{L}}_x : \boldsymbol{\sigma} dV = \int_{\Omega} \rho \mathbf{b} \cdot \mathbf{v} dV + \int_{\Gamma_t} \mathbf{v} \cdot \mathbf{t_n} dA \tag{1}$$

represents, at the time t, the balance of mechanical energy of a continuous body with domain $\Omega(t)$, surface traction boundary $\Gamma_t(t)$, velocity field $\mathbf{v}(\mathbf{x}, t)$, density field $\rho(\mathbf{x}, t)$, spatial velocity gradient field $\boldsymbol{\mathcal{L}}_x = \partial \mathbf{v}(\mathbf{x}, t)/\partial \mathbf{x}$, Cauchy stress field $\boldsymbol{\sigma}(\mathbf{x},t)$, body force per unit mass field $\mathbf{b}(\mathbf{x},t)$ and surface traction field $\mathbf{t}_\mathbf{n}(\mathbf{x},t)$. This equation is the basis for MPM discretization and is used to update particle state using properly time integration.

In addition, considering an homogeneus, linear elastic and isotropic material, the constitutive relation $\dot{\mathbf{S}} = \lambda\mathrm{tr}(\dot{\mathbf{E}})\mathbf{I}+2\mu\dot{\mathbf{E}}$ is applied to update stresses, being $\dot{\mathbf{S}} = d\mathbf{S}/dt$ the time rate of second Piola-Kirchhoff stress tensor and $\dot{\mathbf{E}} = d\mathbf{E}/dt = \mathbf{F}^T\mathbf{D}\mathbf{F}$ the time rate of Green-Lagrange finite strain tensor, where $\mathbf{D}$ is the simetric part of $\boldsymbol{\mathcal{L}}_x$. Values λ and μ are the Lamé constants of the material. The relation between the time rates of Cauchy and second Piola-Kirchhoff stress tensors is given by eq.

$$\dot{\boldsymbol{\sigma}} = -\boldsymbol{\sigma}\mathrm{tr}(\mathbf{D}) + \boldsymbol{\mathcal{L}}_x\boldsymbol{\sigma} + \boldsymbol{\sigma}\boldsymbol{\mathcal{L}}_x^T + \mathbf{F}\dot{\mathbf{S}}\mathbf{F}^T/\det(\mathbf{F}), \tag{2}$$

being $\mathbf{F}$ the deformation gradient tensor. In this study, analyses are performed using plane stress state definitions, where the time rate of out-of-plane strain and the out-of-plane term of strain rate tensor are calculated, respectively, as $\dot{E}_{zZ} = [-\lambda/(\lambda+2\mu)]\left(\dot{E}_{xX}+\dot{E}_{yY}\right)$ and $D_{zz} = \dot{E}_{zZ}/(1+2E_{zZ})$.

2.2 MPM discretization

MPM discretizes a domain Ω into a set of subdomains Ω_p (Lagrangian mesh), which move through an Eulerian grid, as shown in Fig. 1 (a). Each subdomain carries all time history information - mass, velocity, energy, strain - along the time of analysis. Eulerian grid is fixed and allows, at a time step t^n, the calculation of spatial gradients/divergences and the solution of mechanical energy balance. At the beginning of each time step, information is mapped from the subdomains to Eulerian grid nodes, where nodal velocities are updated. Then, these values are mapped back to the Lagrangian mesh to update subdomains state and the Eulerian grid is reset.

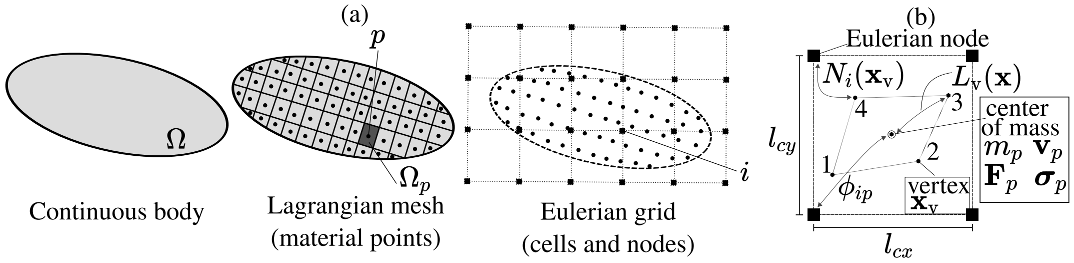

Figure 1. (a) MPM discretization, (b) CPDI MPM mapping scheme.

According to MPM, the spatial interpolation of velocities through the Eulerian grid is written as

$$\mathbf{v}(\mathbf{x}^n) = \sum_i N_i(\mathbf{x}^n)\mathbf{v}_i^n, \tag{3}$$

where i is the index of the Eulerian grid node, $\mathbf{v}_i^n$ are the velocities of these nodes and $N_i(\mathbf{x}^n)$ are the values of nodal interpolation functions evaluated at point with position $\mathbf{x}^n$. Thus, considering eq. (3) and that $\Omega^n = \sum_p \Omega_p^n$, eq. (1) is rewritten, for the i-th node, as

$$\dot{\boldsymbol{p}}_i^n = \mathbf{f}_{b_i}^n + \mathbf{f}_{t_i}^n - \mathbf{f}_{\sigma_i}^n, \tag{4}$$

also known as the momentum equation of MPM. In eq. (4), $\dot{\boldsymbol{p}}_i^n = \sum_p \int_{\Omega_p^n} \rho(\mathbf{x})N_i(\mathbf{x})\dot{\mathbf{v}}(\mathbf{x})dV$ is the time rate of momentum at i-th node, where $\rho(\mathbf{x},t)$ is the mass density field, $\mathbf{f}_{b_i}^n = \sum_p \int_{\Omega_p^n} N_i(\mathbf{x})\rho(\mathbf{x})\mathbf{b}(\mathbf{x})dV$ is the external body force at i-th node, $\mathbf{f}_{t_i}^n = \sum_p \int_{\Gamma_{t_p}^n} N_i(\mathbf{x})\mathbf{t}_\mathbf{n}(\mathbf{x})dA$ is the external surface traction force at i-th node and $\mathbf{f}_{\sigma_i}^n = \sum_p \int_{\Omega_p^n} \boldsymbol{\sigma}(\mathbf{x})\dfrac{\partial N_i(\mathbf{x})}{\partial \mathbf{x}}dV$ is the internal force at i-th node.

2.3 CPDI MPM

As represented in Fig. 1 (b), CPDI MPM, published by Sadeghirad et al. [7, 8], is a variant of MPM in which the masses, velocities, stresses and strains are stored at the center of mass of subdomains and positions are tracked using its vertices. The main difference between CPDI and original MPM [1] is the monitoring of subdomains shape, facilitating the application of boundary conditions and removing cell-crossing error. The interpolation functions used in CPDI MPM, given by

$$N_i^{\mathrm{CPDI}}(\mathbf{x}) = \sum_{\mathrm{v}} N_i(\mathbf{x}_{\mathrm{v}}) L_{\mathrm{v}}(\mathbf{x}), \tag{5}$$

are composed by two portions: the external one $N_i(\mathbf{x}_{\mathrm{v}})$, mapping from the vertices to the Eulerian nodes, and the internal one $L_{\mathrm{v}}(\mathbf{x})$, mapping from the internal part of the subdomain to its vertices. Both functions are bilinear for two-dimensional analysis, being

$$N_i(\mathbf{x}) = \mathcal{R}_i(x)\mathcal{S}_i(y), \tag{6}$$

with

$$\mathcal{R}_i(x) = \begin{cases} 1 + \mathrm{r}_i(x) : -1 \le \mathrm{r}_i(x) < 0 \\ 1 - \mathrm{r}_i(x) : 0 \le \mathrm{r}_i(x) \le 1 \\ 0 : \mathrm{r}_i(x) < -1, \mathrm{r}_i(x) > 1 \end{cases} \quad \text{and} \quad \mathcal{S}_i(y) = \begin{cases} 1 + \mathrm{s}_i(y) : -1 \le \mathrm{s}_i(y) < 0 \\ 1 - \mathrm{s}_i(y) : 0 \le \mathrm{s}_i(y) \le 1 \\ 0 : \mathrm{s}_i(y) < -1, \mathrm{s}_i(y) > 1 \end{cases}, \tag{7}$$

where $\mathrm{r}_i(x) = (x - x_i)/l_{cx}$ and $\mathrm{s}_i(y) = (y - y_i)/l_{cy}$ are relative coordinates between a point and the i-th node and l_{cx} and l_{cy} are the dimensions of Eulerian grid cells along x and y directions, respectively. Internal functions, in turn, are given, for a 4-node quadrilateral subdomain, by eq. (8) FEM bilinear shape functions

$$L_1(\xi,\eta) = \frac{(1-\xi)(1-\eta)}{4} \quad L_2(\xi,\eta) = \frac{(1+\xi)(1-\eta)}{4} \quad L_3(\xi,\eta) = \frac{(1+\xi)(1+\eta)}{4} \quad L_4(\xi,\eta) = \frac{(1-\xi)(1+\eta)}{4}, \tag{8}$$

being (ξ, η) internal local coordinates varying from -1 to 1. Considering the CPDI assumptions and adopting a lumped mass matrix formulation, nodal momentum time rate is rewritten as

$$\dot{\boldsymbol{p}}_i = m_i^n \dot{\mathbf{v}}_i^n = \left(\sum_p \phi_{ip}^{\mathrm{CPDI},n} m_p \right) \dot{\mathbf{v}}_i^n, \tag{9}$$

where m_p is the Ω_p subdomain total mass, $\dot{\mathbf{v}}_i^n = \mathbf{a}_i^n$ is the i-th node acceleration at time t^n and

$$\phi_{ip}^{\mathrm{CPDI},n} = \frac{1}{V_p^n} \int_{\Omega_p^n} N_i^{\mathrm{CPDI}}(\mathbf{x}) dV = \left[\frac{1}{48\left(x_1^n y_2^n - x_1^n y_4^n - x_2^n y_1^n + x_2^n y_3^n - x_3^n y_2^n + x_3^n y_4^n + x_4^n y_1^n - x_4^n y_3^n\right)} \right] \\ \left[\left(1 - \kappa_A^n - \kappa_B^n\right) N_i(\mathbf{x}_1^n) + \left(1 - \kappa_A^n + \kappa_B^n\right) N_i(\mathbf{x}_2^n) + \left(1 + \kappa_A^n + \kappa_B^n\right) N_i(\mathbf{x}_3^n) + \left(1 + \kappa_A^n - \kappa_B^n\right) N_i(\mathbf{x}_4^n) \right] \tag{10}$$

is the interpolation weight of a 4-node quadrilateral subdomain Ω_p to node i, with $\kappa_A^n = \left(x_4^n - x_1^n\right)\left(y_2^n - y_3^n\right) - \left(x_2^n - x_3^n\right)\left(y_4^n - y_1^n\right)$ and $\kappa_B^n = \left(x_3^n - x_4^n\right)\left(y_1^n - y_2^n\right) - \left(x_1^n - x_2^n\right)\left(y_3^n - y_4^n\right)$ being coefficients which are functions of Ω_p^n vertices coordinates and with V_p^n the volume of Ω_p^n. The same $\phi_{ip}^{\mathrm{CPDI},n}$ interpolation weights are used to map external body forces to nodes using eq.

$$\mathbf{f}_{b_i}^n = \sum_p \phi_{ip}^{\mathrm{CPDI},n} m_p \mathbf{b}_p^n, \tag{11}$$

where $\mathbf{b}_p^n$ is the subdomain Ω_p^n average body force per unit mass. In case of external surface forces exclusively applied to Ω_p^n vertices, which is the case of this paper, the eq.

$$\mathbf{f}_{t_i}^n = \sum_p \sum_{\mathrm{v}} N_i(\mathbf{x}_{\mathrm{v}}^n) \mathbf{f}_{\mathrm{v}}^n \tag{12}$$

straight mapping can be done, being $\mathbf{f}_{\mathrm{v}}^n$ the force applied to local vertex v of subdomain Ω_p^n. For the calculation of nodal internal forces

$$\mathbf{f}_{\sigma_i}^n = \sum_p \boldsymbol{\sigma}_p^n \nabla_x \phi_{ip}^{\mathrm{CPDI},n} V_p^n, \tag{13}$$

where $\boldsymbol{\sigma}_p^n$ is the average Cauchy stress tensor of subdomain Ω_p^n, it is necessary to determine the spatial gradients of interpolation weights by means of eq.

$$\nabla_x \phi_{ip}^{\mathrm{CPDI},n} = \left[\frac{1}{4\left(x_1^n y_2^n - x_1^n y_4^n - x_2^n y_1^n + x_2^n y_3^n - x_3^n y_2^n + x_3^n y_4^n + x_4^n y_1^n - x_4^n y_3^n\right)} \right] \\ \left[N_i(\mathbf{x}_1^n) \left\{ \begin{array}{c} y_2^n - y_4^n \\ x_4^n - x_2^n \end{array} \right\} + N_i(\mathbf{x}_2^n) \left\{ \begin{array}{c} y_3^n - y_1^n \\ x_1^n - x_3^n \end{array} \right\} + N_i(\mathbf{x}_3^n) \left\{ \begin{array}{c} y_4^n - y_2^n \\ x_2^n - x_4^n \end{array} \right\} + N_i(\mathbf{x}_4^n) \left\{ \begin{array}{c} y_1^n - y_3^n \\ x_3^n - x_1^n \end{array} \right\} \right] \tag{14}$$

2.4 Algorithm

In order to analyze the stability of elastic columns using a nonlinear two-dimensional approach, a *Python 3* [6] computational script was developed, using CPDI MPM [7, 8] assumptions with forward Euler explicit time integration, plane stress theory and update-stress-last (USL) scheme. A simplified flowchart of implemented calculations is shown in Fig. 2. The process starts with the discretization of entire domain Ω into n_p Ω_p subdomains and, consequently, subdomains initial properties are settled: vertices positions, masses, volumes, velocities, deformation gradients and Cauchy stresses. In parallel, the nodal coordinates $\mathbf{x}_i$ of the Eulerian background grid are defined. Then, initial time $t^1 = 0$ and initial time step $n = 1$ are assigned and a computational loop is started.

Firstly, the time step size Δt^n is calculated and, next, masses, momenta, internal and external forces are mapped from subdomains to Eulerian nodes. Thus, nodal momenta are updated using MPM momentum equation and the respective updated nodal velocities are calculated and Dirichlet boundary conditions are applied to nodes coinciding with structure supports. Hence, increments in nodal velocities are mapped back to update subdomain velocities and nodal updated velocities are used to calculate subdomain spatial velocity gradients $\boldsymbol{\mathcal{L}}_{x_p}^n$. Next, the calculation of strain rate tensors $\mathbf{D}_p^n$ is done, in order to determine the time rate of Green-Lagrange finite strain tensors $\dot{\mathbf{E}}_p^n$. Also, a correction of out-of-plane components $(\dot{E}_{zZ})_p^n$ and $(D_{zz})_p^n = (\mathcal{L}_{x_{zz}})_p^n$ is done to include plane stress theory. After that, the time rate of second Piola-Kirchhoff stress tensors are calculated. These stress rates are converted to the time rates of Cauchy stress tensors, using velocity and deformation gradients. So, Cauchy stresses and deformation gradients are updated with the application of a simple forward Euler time integration scheme. Then, new deformed volumes are calculated using the deformation gradients determinants and finite strain tensors are updated. After that, time step is increased by 1 and the new time value is compared to the desired time of analysis t_{an}. If $t^n < t_{\mathrm{an}}$, a new loop is started, otherwise process is ended.

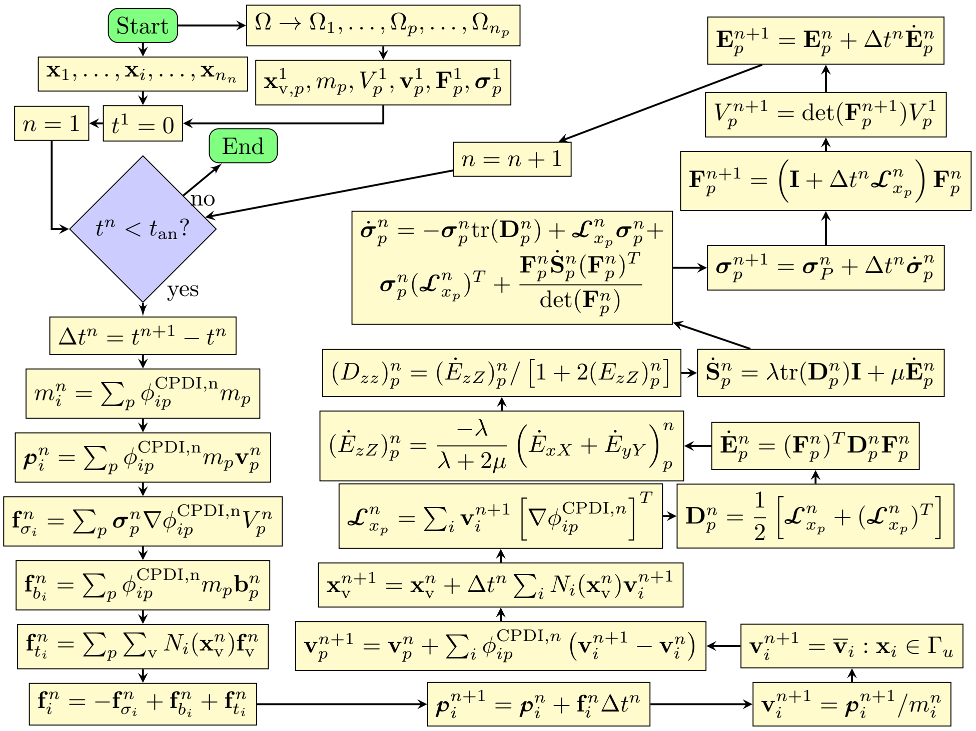
Start
$\Omega \rightarrow \Omega_1, \ldots, \Omega_p, \ldots, \Omega_{n_p}$
$\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_{n_n}$
$\mathbf{x}_{\mathrm{v},p}^1, m_p, V_p^1, \mathbf{v}_p^1, \mathbf{F}_p^1, \boldsymbol{\sigma}_p^1$
$n = 1$
$t^1 = 0$
End
no
$t^n < t_{\mathrm{an}}?$
yes
$\Delta t^n = t^{n+1} - t^n$
$m_i^n = \sum_p \phi_{ip}^{\mathrm{CPDI,n}} m_p$
$\boldsymbol{p}_i^n = \sum_p \phi_{ip}^{\mathrm{CPDI,n}} m_p \mathbf{v}_p^n$
$\mathbf{f}_{\sigma_i}^n = \sum_p \boldsymbol{\sigma}_p^n \nabla \phi_{ip}^{\mathrm{CPDI,n}} V_p^n$
$\mathbf{f}_{b_i}^n = \sum_p \phi_{ip}^{\mathrm{CPDI,n}} m_p \mathbf{b}_p^n$
$\mathbf{f}_{t_i}^n = \sum_p \sum_{\mathrm{v}} N_i(\mathbf{x}_{\mathrm{v}}^n) \mathbf{f}_{\mathrm{v}}^n$
$\mathbf{f}_i^n = -\mathbf{f}_{\sigma_i}^n + \mathbf{f}_{b_i}^n + \mathbf{f}_{t_i}^n$
$\boldsymbol{p}_i^{n+1} = \boldsymbol{p}_i^n + \mathbf{f}_i^n \Delta t^n$
$\mathbf{v}_i^{n+1} = \boldsymbol{p}_i^{n+1} / m_i^n$
$\mathbf{v}_i^{n+1} = \bar{\mathbf{v}}_i : \mathbf{x}_i \in \Gamma_u$
$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \sum_i \phi_{ip}^{\mathrm{CPDI},n} \left(\mathbf{v}_i^{n+1} - \mathbf{v}_i^n\right)$
$\mathbf{x}_{\mathrm{v}}^{n+1} = \mathbf{x}_{\mathrm{v}}^n + \Delta t^n \sum_i N_i(\mathbf{x}_{\mathrm{v}}^n) \mathbf{v}_i^{n+1}$
$\boldsymbol{\mathcal{L}}_{x_p}^n = \sum_i \mathbf{v}_i^{n+1} \left[\nabla \phi_{ip}^{\mathrm{CPDI},n}\right]^T$
$\mathbf{D}_p^n = \frac{1}{2} \left[\boldsymbol{\mathcal{L}}_{x_p}^n + (\boldsymbol{\mathcal{L}}_{x_p}^n)^T\right]$
$\dot{\mathbf{E}}_p^n = (\mathbf{F}_p^n)^T \mathbf{D}_p^n \mathbf{F}_p^n$
$(\dot{E}_{zZ})_p^n = \frac{-\lambda}{\lambda + 2\mu} \left(\dot{E}_{xX} + \dot{E}_{yY}\right)_p^n$
$(D_{zz})_p^n = (\dot{E}_{zZ})_p^n / \left[1 + 2(E_{zZ})_p^n\right]$
$\dot{\mathbf{S}}_p^n = \lambda \mathrm{tr}(\mathbf{D}_p^n) \mathbf{I} + \mu \dot{\mathbf{E}}_p^n$
$\dot{\boldsymbol{\sigma}}_p^n = -\boldsymbol{\sigma}_p^n \mathrm{tr}(\mathbf{D}_p^n) + \boldsymbol{\mathcal{L}}_{x_p}^n \boldsymbol{\sigma}_p^n + \boldsymbol{\sigma}_p^n (\boldsymbol{\mathcal{L}}_{x_p}^n)^T + \frac{\mathbf{F}_p^n \dot{\mathbf{S}}_p^n (\mathbf{F}_p^n)^T}{\det(\mathbf{F}_p^n)}$
$\boldsymbol{\sigma}_p^{n+1} = \boldsymbol{\sigma}_P^n + \Delta t^n \dot{\boldsymbol{\sigma}}_p^n$
$\mathbf{F}_p^{n+1} = \left(\mathbf{I} + \Delta t^n \boldsymbol{\mathcal{L}}_{x_p}^n\right) \mathbf{F}_p^n$
$V_p^{n+1} = \det(\mathbf{F}_p^{n+1}) V_p^1$
$\mathbf{E}_p^{n+1} = \mathbf{E}_p^n + \Delta t^n \dot{\mathbf{E}}_p^n$
$n = n + 1$

Figure 2. Flowchart of implemented CPDI MPM algorithm.

2.5 Analyzed columns and analysis parameters

According to Fig. 3, a clamped-free column is analyzed under 4 different compressive load cases: centered load, centered load with a minimal transverse load (1000 times smaller than the axial load) and eccentric loads with eccentricities equals to $0.25l_y$ and $0.50l_y$. Column geometry and material properties are also summarized in Fig. 3. The column is subdivided in $n_p = 60$ subdomains with initial dimensions $10 \times 10 \times 1$ cm and the analysis domain is discretized in an Eulerian background grid with 35×40 cells with dimensions 10×10 cm, as shown in Fig. 4 (a). Taking into account Dirichlet boundary conditions, nodes that are coincident to support vertices of column are restrained. The load function $P(t) = P_E\left[1 + \sin(\pi t/t_{\text{an}} - \pi/2)\right]$, graphically represented in Fig. 4 (b) and valid for the total time of analysis $t_{\text{an}} = 40$ s, is based on the analytical Euler's buckling critical load of the column $P_E = \pi EI/(2l_x)^2 = 0.7494$ kN. The purpose of this sinusoidal function is to increase axial load as smooth as possible, obtaining a static analysis, without significant vibration.

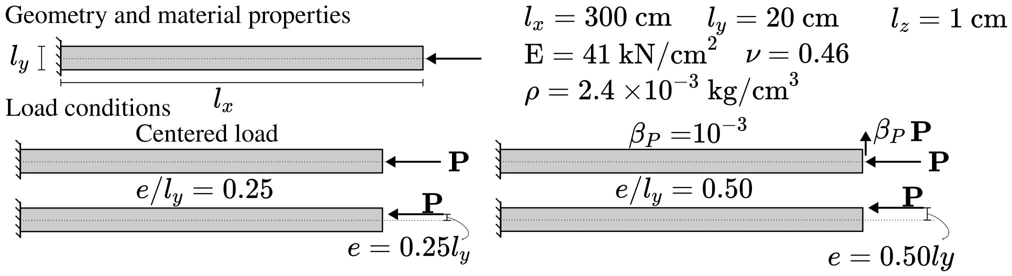

Figure 3. Geometry, material properties and load conditions.

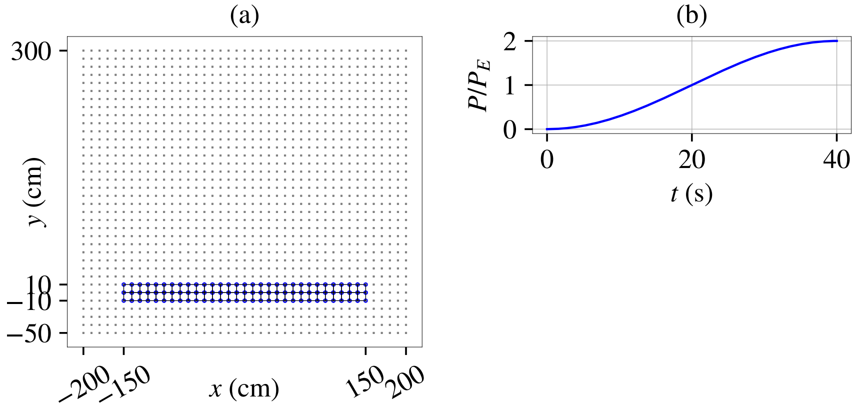

Figure 4. (a) Eulerian grid nodes (grey dots) and Lagrangian mesh (blue dots conected by black lines), (b) load function.

3 Results and Discussions

Figures 5 (a) and (b) present, respectively, the axial and transverse load-displacement behavior, detected at the column free end midpoint, in the analyses of the four load cases. The processing of "centered load" and $\beta_P = 10^{-3}$ cases are interrupted before t_{an} is totally completed, in order to avoid the occurrence of strong vibrations in the results, something that is outside the scope of this work. In Fig. 6, evolution of total potential energy $\Pi(t) = \mathcal{U}(t) + \mathcal{K}(t)$ (sum of kinetic and internal energies) and total work done by external loads $\mathcal{W}(t)$ are plotted along time. Red dots in the charts represent the beginning of instability, when the energetic equilibrium starts an abrupt variation. This point is chosen, in this study, as the maximum time t^n before $\Delta\left(\Pi^n/\Delta t^n\right) = \left(\Pi^{n+1} - \Pi^n\right) / \left(t^{n+1} - t^n\right)$ exceeds 1.00 kN.cm/s. Blue dots, in turn, represent the end of the analysis. Finally, deformed shapes at times symbolized by red and blue dots are plotted in Fig. 7.

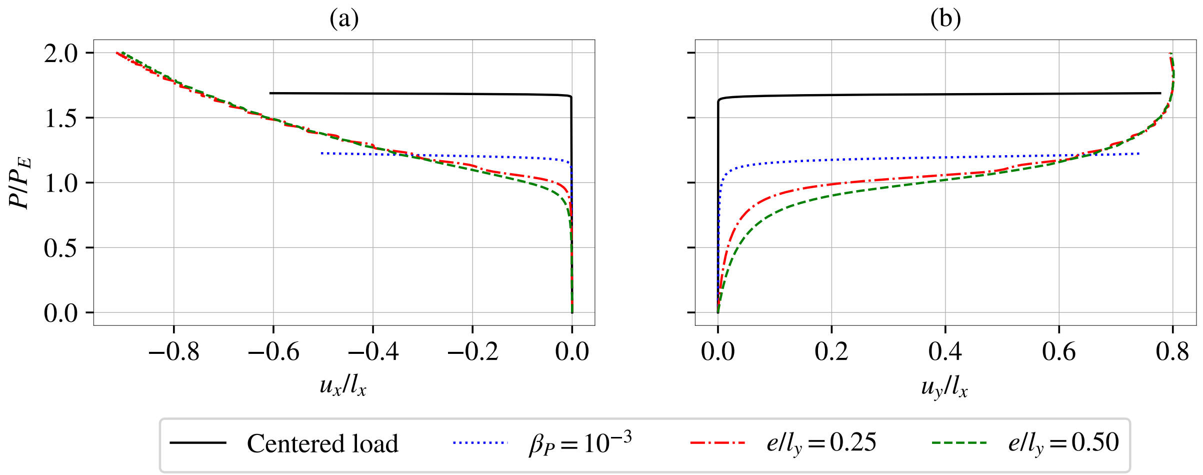

Figure 5. Displacement-load behavior measured at the free end of analyzed columns in x (a) and y (b) directions.

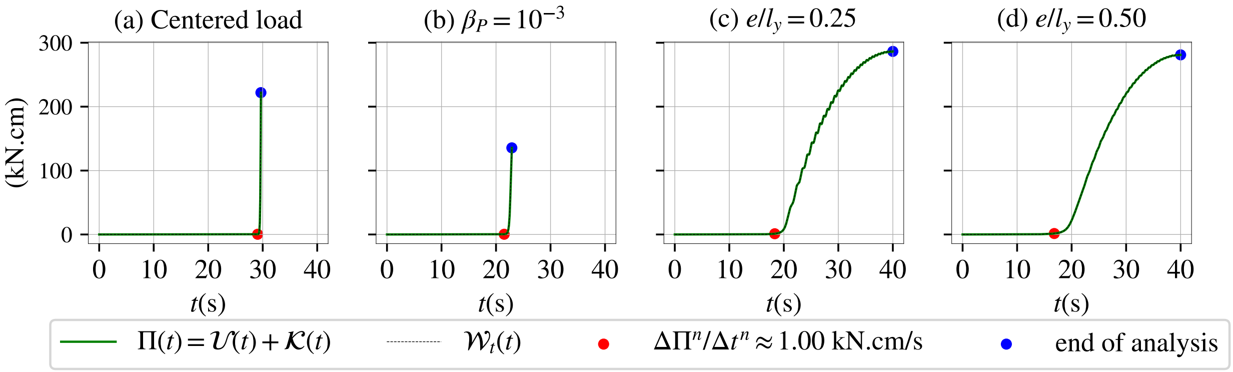

Figure 6. Total potential energy and total work done by external loads.

In general, according to results, the performed stability analyses using CPDI MPM presented a good accuracy when compared to Euler's critical load, with the benefit of considering large deformations and tracking post-critical behavior. Even the column under centered axial load presented instability when $P/P_E \approx 1.65$ (Figs. 6 (a) and 7 (a)), as a consequence of a numerical eccentricity caused by large strains in a slender structure. Additionally, results of load case $\beta_P = 10^{-3}$ - a minimal perturbation - show that column has a loss of stability when $P/P_E \approx 1.12$, as can be seen in Figs. 6 (b) and 7 (c). This value is above 1, indicating that the consideration of nonlinear finite strains and two dimensions increases unidimensional linear critical loads by approximately 12%. As expected, for eccentric load cases, loss of stability occurs under lower axial loads $P/P_E \approx 0.87$ when $e/l_y = 0.25$ - Figs. 6 (c) and 7 (e) - and $P/P_E \approx 0.75$ when $e/l_y = 0.50$ - Figs. 6 (d) and 7 (h). Furthermore, the displacement-load behavior of both cases didactically show that the column leaves an unstable equilibrium state and reaches the same stable post-critical equilibrium, regardless of the eccentricity magnitude. Moreover, Fig. 6 energy plots not only confirm the implemented CPDI MPM algorithm capability of conserving energy in the absence of dissipative forces, but also clearly indicate the beginning of columns instability. Also, it is verified in Fig. 5 that the load application function $P(t)$ is able to represent a static loading in dynamic analyses, avoiding significant vibrations, even without any consideration of damping forces.

4 Conclusions

In this study, a two-dimensional analysis of a clamped-free column stability is performed using a MPM CPDI algorithm, considering plane stress theory and finite strains. Governing equations, derived from the continuum

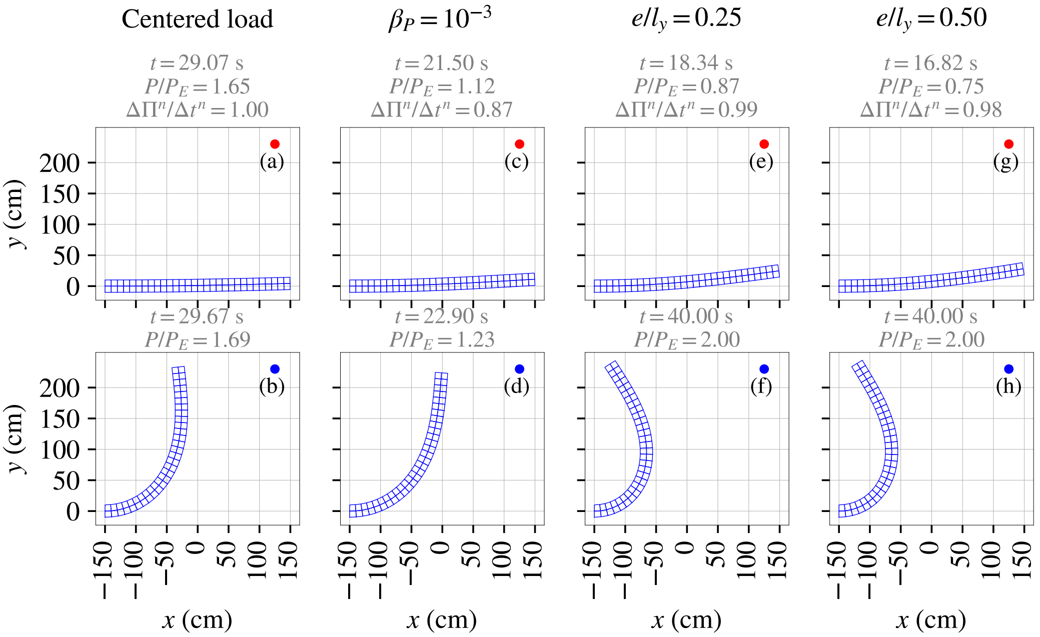

Figure 7. Deformed shapes shortly before buckling (red circle) and in the end of analysis (blue circle).

mechanics, are present, as well as CPDI MPM equations and the computational algorithm flowchart. This algorithm is used to process 4 different load cases and the results produced are displacement-load and time-energy curves, with which the accuracy and ability of implemented algorithm to deal with large displacements and strains are confirmed. In addition, analyses are capable to precisely predict buckling loads and to track all post-critical behavior, keeping numerical convergence and energy conservation.

Acknowledgements. The authors gratefully acknowledge the financial support provided by CAPES (Coordination for the Improvement of Higher Education Personnel) and Araucária Foundation.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] D. Sulsky, Z. Chen, and H. L. Schreyer. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, vol. 118, pp. 179–196, 1994.
[2] S. G. Bardenhagen, J. U. Brackbill, and D. Sulsky. The material-point method for granular materials. *Computer Methods in Applied Mechanics and Engineering*, vol. 187, pp. 529–541, 2000.
[3] J. Fern, A. Rohe, K. Soga, and E. Alonso. *The Material Point Method for Geotechnical Engineering*. CRC Press, Taylor and Francis Group, 2019.
[4] V. P. Nguyen, C. T. Nguyen, T. Rabczuk, and S. Natarajan. On a family of convected particle domain interpolations in the material point method. *Finite Elements in Analysis and Design*, vol. 126, pp. 50–64, 2017.
[5] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. A material point method for snow simulation. *ACM Transactions on Graphics*, vol. 32, pp. 102:1–102–9, 2013.
[6] G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.
[7] A. Sadeghirad, R. M. Brannon, and J. Burghardt. A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. *International Journal for Numerical Methods in Engineering*, vol. 86, pp. 1435–1456. cited By 131, 2011.
[8] A. Sadeghirad, R. M. Brannon, and J. E. Guilkey. Second-order convected particle domain interpolation (cpdi2) with enrichment for weak discontinuities at material interfaces. *International Journal for Numerical Methods in Engineering*, vol. 95, pp. 928–952. cited By 49, 2013.