



**HAL**  
open science

# Critical Comparison of Motion Integration Strategies and Discretization Choices in the Material Point Method

Sacha Duverger, Jérôme Duriez, Pierre Philippe, Stéphane Bonelli

## ► To cite this version:

Sacha Duverger, Jérôme Duriez, Pierre Philippe, Stéphane Bonelli. Critical Comparison of Motion Integration Strategies and Discretization Choices in the Material Point Method. *Archives of Computational Methods in Engineering*, 2025, 32 (3), pp.1369-1397. <10.1007/s11831-024-10170-y>. <hal-04707839>

**HAL Id: hal-04707839**

**<https://hal.inrae.fr/hal-04707839v1>**

Submitted on 24 Sep 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Copyright - All rights reserved

1 Critical comparison of motion integration  
2 strategies and discretization choices in the  
3 Material Point Method

4 Sacha Duverger<sup>1</sup>, Jérôme Duriez<sup>1\*</sup>, Pierre Philippe<sup>1</sup>  
5 and Stéphane Bonelli<sup>1</sup>

6 <sup>1</sup>INRAE, Aix Marseille Univ, RECOVER, Aix-en-Provence,  
7 France.

8 \*Corresponding author(s). E-mail(s): [jerome.duriez@inrae.fr](mailto:jerome.duriez@inrae.fr);  
9 Contributing authors: [sacha.duverger@inrae.fr](mailto:sacha.duverger@inrae.fr);  
10 [pierre.philippe@inrae.fr](mailto:pierre.philippe@inrae.fr); [stephane.bonelli@inrae.fr](mailto:stephane.bonelli@inrae.fr);

11 **Abstract**

12 To simulate large, history-dependent material displacements, the Mate-  
13 rial Point Method (MPM) solves for the kinematics of Lagrangian  
14 material points being embedded with mechanical variables while mov-  
15 ing freely within a fixed mesh. The MPM procedure makes use of  
16 the latter mesh as a computational grid, where the momentum bal-  
17 ance equation with the acceleration field are first projected onto nodes,  
18 before material points can be moved. During that process, a number  
19 of different choices have been adopted in the literature for what con-  
20 cerns the computational definition of time increments of velocity and  
21 position, from the knowledge of nodal acceleration. An overview of  
22 these different motion integration strategies is herein proposed, with a  
23 particular emphasis on their impact onto the MPM conservative proper-  
24 ties. Original results illustrate the discussion, considering either simple  
25 configurations of solid translation and rotation or a more complex col-  
26 lapse of a frictional mass. These analyses furthermore reveal hidden  
27 properties of some motion integration strategies regarding conserva-  
28 tion, namely a direct influence of the time step value during a time  
29 integration being inspired by the Particle In Cell (PIC) ancestor of

the MPM. The spatial, resp. temporal (in comparison with vorticity), discretizations are also shown to affect the angular momentum conservation of the FLIP method, resp. an affine extension of PIC (APIC).

**Keywords:** Material Point Method; PIC; FLIP; APIC; XPIC

## 1 Introduction

Anticipating the deformations of large-scale constructions, e.g. water dams possibly rising up to hundreds of meters, under various mechanical loads is crucial for safety concerns. These studies are typically carried out with numerical methods based on the Finite Element Method (FEM) and corresponding pioneering works [1, 2] for continuous materials. The FEM unfortunately suffers from a number of limitations related with its underlying Lagrangian mesh for solid mechanics. In particular, many scenarios of interest involve large deformations, e.g. possible structure failure, where extreme mesh distortions would prevent the FEM procedure to pursue. A solution to overcome this issue can be to include a conditional remeshing step in the FEM framework [3, 4, 5] but this can turn to be computationally expensive. If present, e.g. for elastoplastic solids, a field of hardening variables has also to be recast into the new mesh which makes things even more intricate. These issues have been reviewed e.g. in [6], with a particular focus onto the Particle Finite Element Method (PFEM) [7] as an appropriate remeshing FEM technique in presence of hardening variables.

In contrast, the historical Particle In Cell method (PIC) [8] has avoided distortion issues by fixing the mesh, turning it into an Eulerian frame within which the integration points of the FEM are free to move. As such, the PIC method came as an hybrid Eulerian-Lagrangian method with a robust nature in the case of large displacements. Then, in [9], the FLuid Implicit Particle

56 method (FLIP) has revisited PIC with the idea of limiting the numerical dif-  
57 fusion and energy dissipation the latter suffers from, adapting in particular  
58 the way integration points velocities are defined from the acceleration at grid  
59 points. These methods eventually led to the slightly more recent Material Point  
60 Method (MPM) [10], possibly in a GIMP [11] or CPDI [12] variants, which aims  
61 to extend the hybrid Eulerian-Lagrangian formulation to history-dependent  
62 materials and has become a popular tool to simulate large displacements in  
63 solid mechanics, see e.g. [13, 14, 15] for general MPM reviews.

64 Following PIC and FLIP, the MPM essentially solves continuum mechan-  
65 ics equations and computes accelerations on fixed mesh grid points in order  
66 to eventually update positions of Lagrangian material points. Subtleties then  
67 arise during the MPM procedure with a frequent transport, i.e. mapping,  
68 of mechanical quantities between these two uncoupled spatial discretizations,  
69 which deteriorates the proper definitions of classical kinematic fields of acceler-  
70 ations, velocities and positions. As it will be further exposed in the manuscript,  
71 it is for instance to note that material point velocities may not always be the  
72 time integral of their accelerations, nor the time derivatives of their positions.  
73 These subtleties lead the MPM to face similar challenges as PIC and FLIP for  
74 what concerns the conservation of energy and momentum. As a matter of fact,  
75 these fundamental objectives are not always fulfilled and prompted a number  
76 of previous studies.

77 As for momentum conservation, the MPM is formulated such that linear  
78 momentum is automatically conserved but strong issues may concern angular  
79 momentum and rotational motion, as discussed, e.g., in [16, 17, 18] and further  
80 recalled in the remainder of the manuscript. It was actually shown in [16, 17]  
81 that the angular momentum is lost both with PIC and FLIP during the two-  
82 ways transport between material points and grid points with classical lumped

4 *Comparison of MPM motion integration strategies and discretization choices*

83 expressions of nodal masses and an Affine-augmented Particle In Cell method  
84 (APIC) was accordingly proposed in [17] aiming to solve the issue. Before  
85 being further improved in [18], APIC was shown in [17] to conserve the energy  
86 better than PIC if not perfectly, while being more stable than FLIP. Since  
87 the remaining energy dissipation sources in APIC [17] basically come from its  
88 share of PIC transport equations, an affine-augmented variation of the FLIP  
89 velocity mapping, AFLIP, was tested in [19] and logically found to conserve  
90 the energy even better than both APIC and FLIP.

91 Similar to APIC and AFLIP which eventually aim at better capturing affine  
92 velocity fields when mapping one field from another, a spatial gradient-based  
93 Taylor expansion has been proposed in [20] and later denoted as Taylor-PIC  
94 (TPIC) in [21].

95 Another recent study [22] highlights that the motion integration procedure  
96 actually corresponds to choosing a velocity field among an infinity of solutions  
97 to the velocity transportation equations. With this point of view, the PIC  
98 strategy appears as the solution which minimizes the velocity by filtering out  
99 all the noise, while the FLIP strategy minimizes the acceleration and does not  
100 filter any noise. Using this insight, a new XPIC( $m$ ) strategy was proposed in  
101 the same study [22], for adopting the velocity field that minimizes the difference  
102 with respect to its previous values filtered at the order  $m$ . The XPIC( $m$ )  
103 strategy can thus be used to adapt the noise filtration for a specific problem,  
104 giving a great adjustable compromise between PIC and FLIP strategies. Based  
105 on the XPIC( $m$ ) strategy, a FMPM (Full mass matrix MPM) formulation was  
106 proposed in [23], considering an approximation of the full mass matrix (inverse)  
107 instead of the traditional lumped one. Although this approach complexifies the  
108 imposition of boundary conditions, results were shown to be less dissipative  
109 and more accurate than both FLIP and XPIC( $m$ ).

110 In connection with the evergrowing popularity of the MPM, the present  
111 manuscript then aims to clarify the underlying reasons for so many choices  
112 regarding motion integration in a MPM implementation and review most of  
113 the currently available alternatives. The discussion is illustrated from original  
114 numerical results evidencing their respective performances in terms of energy  
115 and momentum conservation.

116 After recalling the general MPM background in Section 2, Section 3  
117 proposes a grouped presentation of the various possible motion integration  
118 strategies. Section 4 then investigates the conservative nature of several of  
119 those in the simple cases of an elastic solid moving in a purely translational  
120 motion or a rigid solid subject to a rotational motion, where analytical ground-  
121 truth results are available for reference. The analysis is finally extended to  
122 the more complex and rather classical case of the granular column collapse in  
123 Section 5, together with a discussion of the (time and spatial) discretization  
124 parameters.

125 Although the calculations performed in this study are likely to involve large  
126 transformations (large deformations and large rotations), our model is based  
127 on an infinitesimal transformation formulation. Taking large transformations  
128 into account, similar to [12, 23, 24, 25], would lead to more accurate results,  
129 but would not change the conclusions of the present work which is focused on  
130 the comparison of motion integration strategies and discretization choices in  
131 the MPM formulation. All simulations are executed with a modified version  
132 of the open-source code CB-Geo MPM [26], used together with its PyCBG  
133 interface [27].

## 134 2 Common MPM principles

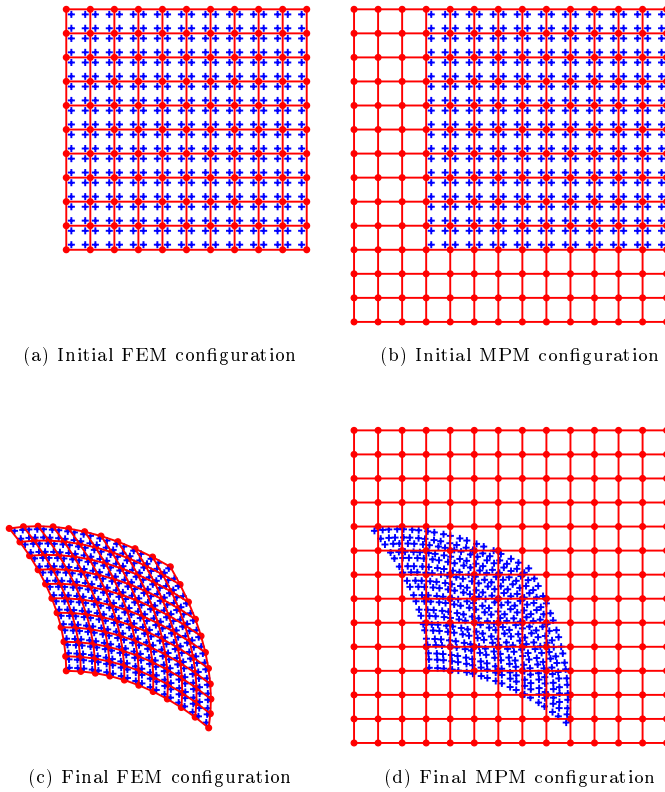
### 135 2.1 Governing equation and double spatial discretization

136 Considering a deformable solid being continuously present in the domain  $\Omega$ ,  
 137 mass density, velocity and stress fields are respectively denoted  $\rho$ ,  $\vec{V}$  and  $\sigma$ ,  
 138 omitting time and space variables for simplicity. External forces may apply  
 139 onto  $\Gamma$ , the surface of  $\Omega$ , as tractions  $\vec{\tau} = \sigma \cdot \vec{n}$ , with  $\vec{n}$  the outward normal  
 140 to  $\Gamma$ . If present, a constant, uniform, gravitational field is denoted as  $\vec{g}$ . With  
 141  $\vec{w}$  a test function (a kinematically admissible velocity field, typically) the weak  
 142 form of the virtual work principle classically reads:

$$\int_{\Omega} \rho \vec{w} \cdot \frac{\partial \vec{V}}{\partial t} d\Omega = \int_{\Omega} \rho \vec{w} \cdot \vec{g} d\Omega + \int_{\Gamma} \vec{w} \cdot \vec{\tau} d\Gamma - \int_{\Omega} \nabla \vec{w} : \sigma d\Omega \quad (1)$$

143 As proposed in [10], the MPM solves Eq. (1) for a Lagrangian velocity  
 144 field, which is defined on a first layer of spatial discretization in terms of a  
 145 finite set  $\{p\}$  of  $N_{mp}$  material points. Those material points will also carry  
 146 each a constant mass quantity  $m^p$  and history-dependent, e.g. elasto-plastic  
 147 hardening, variables and accordingly express material behaviour.

148 Being geometrically uncoupled to the set of material points, the MPM  
 149 additionally relies on an Eulerian mesh with a number  $N_{gp}$  of grid points  $i$   
 150 located at constant positions  $\vec{x}_i$  and connecting into mesh cells i.e. elements,  
 151 see Figure 1b. While being fixed in space as illustrated in Figure 1d in con-  
 152 trast to Figure 1c for the classical FEM in solid mechanics, that Eulerian grid  
 153 plays a FEM-like computational role in which the weak form (1) is eventually  
 154 transposed at every grid point. Grid points actually combine with associated  
 155 shape functions  $N_i(\vec{x})$  to serve as a nodal basis for expressing quantities, e.g.



**Fig. 1:** Mesh (in red) and integration or material points (in blue) evolutions during similar FEM and MPM simulations

156 for the test function:

$$\vec{w}(\vec{x}) = \sum_i \vec{w}_i N_i(\vec{x}) \quad (2)$$

157 while obeying partition of unity:

$$\sum_i N_i(\vec{x}) = 1 \quad (3)$$

158 Eq. (3) can indeed be seen as a necessary condition for (i.e., a consequence of)

159 Eq. (2).

8 *Comparison of MPM motion integration strategies and discretization choices*

160 From this double spatial discretization, the classical MPM procedure leads  
 161 to the following expression for nodal acceleration [10]:

$$\vec{a}^i = \frac{\vec{f}^i}{m^i} \quad (4)$$

$$\vec{f}^i = \vec{f}_{int}^i + \vec{f}_{ext}^i \quad (5)$$

$$m^i = \sum_p N_i(\vec{x}^p) m^p \quad (6)$$

162 where the first two terms on the right-hand side of Eq. (1), describing body  
 163 forces and external surface loads, have been turned into a nodal force quantity,

164  $\vec{f}_{ext}^i$ :

$$\vec{f}_{ext}^i = \sum_p N_i(\vec{x}^p) m^p \vec{g} + \int_{\Gamma} N_i \vec{\tau} d\Gamma \quad (7)$$

165 while the last term of Eq. (1) being representative of internal forces corresponds  
 166 to the following nodal force quantity  $\vec{f}_{int}^i$  obtained from the material points  
 167 stresses:

$$\vec{f}_{int}^i = - \sum_p \frac{m^p}{\rho} \sigma^p \overline{\nabla} \vec{N}_i(\vec{x}^p) = - \sum_p v^p \sigma^p \overline{\nabla} \vec{N}_i(\vec{x}^p) \quad (8)$$

168 It is to note that Eqs. (4)-(6) correspond to a lumped mass matrix approach  
 169 with known consequences for the conservative properties of the method [9, 10,  
 170 16, 28]. While mass is naturally conserved, in the sense that  $\sum_i m^i = \sum_p m^p$ ,  
 171 the cases of momenta and energy are more intricate and will be reviewed below.  
 172 It is also to note that, unlike the initial MPM formulation [10], mass density  
 173 is not considered constant here but instead computed from the constant  $m^p$   
 174 and a variable material point volume  $v^p$ . The latter is initialized from  $v^{j(p)}$   
 175 the volume of the mesh element,  $j$ , including  $p$  and  $N_{mp}^j$  the total number of

176 material points in that cell:

$$v^p(t = 0) = \frac{v^{j(p)}}{N_{mp}^{j(p)}} \quad (9)$$

177 before being updated at each iteration depending on a volumetric strain  
178 measure at the center of the cell  $j$ .

## 179 **2.2 Material points or material domains in the MPM** 180 **discretization**

181 The above discrete equations, Eqs (4)-(8), have been obtained in [10] from  
182 the consideration of a point-wise mass distribution within  $\{p\}$ , with material  
183 point-centered Dirac distributions for the mass density and the stress fields.  
184 As an alternative to this pure “material point” point of view, finite “material  
185 domains” have been proposed in a GIMP variant [11] to carry a piece-wise  
186 constant stress field, through the concept of particle characteristic function  
187 with finite support, unlike the Dirac distribution of classical MPM. Shape  
188 functions are then directly replaced in Eq. (7) (for its first right-hand side  
189 term) and Eq. (8) with weighting functions that convolute shape functions  
190 with particle characteristic functions and show more suitable properties, e.g.  
191 a non-linear nature even with linear shape functions (see below Section 2.3).

192 As imposed by partition of unity considerations, such material domains  
193 should however stay contiguous and non-overlapping for results to stay consis-  
194 tent, e.g., [29]. Particle domains then need to be updated according to material  
195 point kinematics, way beyond pure translation, and the CPDI [12] extension to  
196 GIMP for instance enables one to update those as deformable parallelograms  
197 while considering piece-wise constant deformation gradients and stresses across  
198 domains.

199 One can note that the discrete equations, Eq. (8) in particular, could also  
 200 be seen as a quadrature expression for a numeric integration of the weak form  
 201 terms, avoiding any assumption on the mechanical (e.g., stress) fields and  
 202 considering material points just as integration points or numeric tracers. From  
 203 this point of view, quadrature precision issues arise in connection with material  
 204 point displacements, as discussed in e.g. [13, 30] but those could possibly be  
 205 solved following [31], which is still out of the present scope.

206 Here, the remainder of the sequel adopts the “material point” point of view  
 207 of classical MPM to avoid the complexity of defining particle domains that  
 208 maintain a rigorous tessellation of  $\Omega$  whatever the kinematics.

## 209 **2.3 Chosen shape functions**

210 In this “material point” framework, piecewise linear shape functions such as  
 211 used in [32, 33, 34] are avoided due to their  $C^0$  nature that may lead to  
 212 ringing, i.e. cell crossing, instabilities [35] through the so-called internal force  
 213  $\vec{f}_{int}^i$  in Eq. (8) with the gradient of the shape functions, which would induce  
 214 discontinuous changes in  $\vec{f}_{int}^i$  when a material point crosses the boundary  
 215 between two cells [13, 30, 36]. Boundary modified cubic splines [15] with (twice)  
 216 continuous derivatives are adopted instead.

## 217 **3 Various MPM motion integration strategies**

### 218 **3.1 Intricacy of mappings between material points and** 219 **grid points as motion integration strategies**

220 While the nodal acceleration Eq. (4) is intended to serve as a time update of  
 221 nodal velocities, a salient keypoint of the MPM then appears in the yet unspec-  
 222 ified choices for initial nodal velocities  $\vec{V}^i$  and for the exact role of updated  
 223 values, being denoted  $\widetilde{\vec{V}}^i$  likewise to [8], in the material point kinematics, in

224 necessary connection with material point velocities  $\vec{V}^p$  for both cases. The  
 225 corresponding operations are generically denoted in this paper as the "motion  
 226 integration strategy". Before being reviewed in details below, these could be  
 227 introduced, broadly speaking, as a particular choice of back-and-forth map-  
 228 pings between grid points and material points. From the grid points to the  
 229 particles, a shape-function-inspired mapping for a given quantity  $s$  (which can  
 230 be any tensor of order  $n \geq 0$ ) would be:

$$s^p = \sum_i N_i(\vec{x}^p) s^i \quad (10)$$

231 which can be recast in matrix form:

$$\vec{s}^{\mathbb{P}} = \mathbf{G}^{\star P} \vec{s}^{\mathbb{G}} \quad (11)$$

$$(\mathbf{G}^{\star P})_{pi} = N_i(\vec{x}^p) \quad (12)$$

232 Here  $\vec{s}^{\mathbb{P}}$  denotes the whole set of material point values and  $\vec{s}^{\mathbb{G}}$  its nodal  
 233 counterpart. The arrow in Eq. (11) corresponds to the increase in tensor order  
 234 from  $n$  to  $n+1$  in order to cover the whole sets of material points or grid points,  
 235 with each line of  $\vec{s}^{\mathbb{G}}$  (among  $N_{gp}$ ), resp.  $\vec{s}^{\mathbb{P}}$  (among  $N_{mp}$ ), corresponding to  
 236 the quantity for a specific grid point, resp. material point. In the case of  $s$   
 237 being a vector quantity,  $\vec{s}^{\mathbb{G}}$  and  $\vec{s}^{\mathbb{P}}$  are both second order tensors in Eq. (11)  
 238 and will be denoted  $\mathbf{s}^{\mathbb{G}}$  and  $\mathbf{s}^{\mathbb{P}}$  in the rest of this paper.

239 From the particles to the grid, one could adopt, likewise to the lumped mass  
 240 expression in Eq. (6), the following expression which obeys  $\sum_i s^i = \sum_p s^p$ :

$$s^i = \sum_p N_i(\vec{x}^p) s^p \quad (13)$$

241 or in matrix form:

$$\vec{s}^{\mathbb{G}} = \mathbf{P}^{\star G} \vec{s}^{\mathbb{P}} \quad (14)$$

$$\mathbf{P}^{\star G} = (\mathbf{G}^{\star P})^T \quad (15)$$

242 It has to be observed that in the general case  $N_{mp} \neq N_{gp}$ , making it  
 243 impossible for  $\mathbf{G}^{\star P}$  to be square. There is furthermore even no reason for the  
 244 latter to be at least semi-orthogonal, i.e.  $\mathbf{G}^{\star P} \mathbf{P}^{\star G} \vec{s}^{\mathbb{P}} \neq \vec{s}^{\mathbb{P}}$ , contributing to  
 245 the subtleties of the mappings discussed below, through the introduction of  
 246 so-called null space errors [37].

### 247 **3.2 From known particle velocities to nodal velocities**

248 As for the nodal velocities  $\vec{V}^i$  at the beginning of a time step, those have  
 249 been usually expressed since the FLIP method from material point velocities  
 250 by explicitly conserving the linear momentum  $\vec{u}$  through the application of  
 251 Eq. (14) from  $\vec{u}^p = m^p \vec{V}^p$  to  $\vec{u}^i$  and then dividing by the nodal masses to give:

$$\vec{V}^i = \frac{\sum_p N_i(\vec{x}^p) m^p \vec{V}^p}{m^i} \quad (16)$$

252 The result of Eq. (16) can also be seen as a mass-weighted transport proce-  
 253 dure that directly applies to velocities  $\vec{V}$  and which can be written in matrix  
 254 form as:

$$\mathbf{V}^{\mathbb{G}} = \mathbf{P}_m^{\star G} \mathbf{V}^{\mathbb{P}} \quad (17)$$

$$(\mathbf{P}_m^{\star G})_{ip} = \frac{m^p N_i(\vec{x}^p)}{\sum_{p'} m^{p'} N_i(\vec{x}^{p'})} \quad (18)$$

255 where  $\mathbf{V}^G$  and  $\mathbf{V}^P$  are respectively  $(N_{gp} \times N_{dim})$  and  $(N_{mp} \times N_{dim})$  matrices  
 256 with  $N_{dim}$  the number of dimensions, and  $\mathbf{P}_m^{*G}$  a mass-weighted mapping  
 257 matrix that directly applies to velocity quantities. Similar to the previous case  
 258 of Eqs. (11)-(12), each line of  $\mathbf{V}^G$ , resp.  $\mathbf{V}^P$ , corresponds to the velocity of a  
 259 specific grid point  $\vec{V}^i$ , resp. material point  $\vec{V}^p$ .

260 Even though linear momentum is conserved by construction through  
 261 Eq. (17), it is to note that the present lumped mass approach makes the  
 262 nodal kinetic energy, computed from nodal masses in Eq. (6) and nodal veloc-  
 263 ities in Eq. (16), to be less than the kinetic energy defined on material points  
 264 [9, 10, 16, 38, 39]. In line with the absence of nodal history, this feature nev-  
 265 ertheless does not necessarily hinder overall conservation of energy in itself,  
 266 see [16] and a simple translation illustration under free fall in the forthcoming  
 267 Section 4.1 and Figure 3.

268 The Eqs. (16)-(18) are however unable to correctly infer nodal velocities  
 269 from the ones of material points in a number of situations. Considering for  
 270 instance a case where a grid node  $i$  would be under the influence (through  
 271  $N_i(\vec{x}^p) \neq 0$ ) of just one material point  $p$ , one would have  $\vec{V}^i = \vec{V}^p$  no matter  
 272 the position offset between  $i$  and  $p$ , which is in particular incorrect if velocities  
 273 are to correspond to an affine field, as discussed in [20]. As such, it has been  
 274 proposed in [20], and later coined as a Taylor-PIC (TPIC) strategy in [21], to  
 275 use a gradient-based extrapolation, i.e. a first order Taylor expression, for a  
 276 better description of affine velocity fields from material point values:

$$\mathbf{V}^G = \mathbf{P}_m^{*G} \mathbf{V}^P + \mathbf{W}^G \quad (19)$$

$$(W^G)_{ik} = \sum_p \sum_j (P_m^{*G})_{ip} (\nabla \vec{V}^p)_{jk} (b_i^p)_j \quad (20)$$

$$\vec{b}_i^p = \vec{x}^i - \vec{x}^p \quad (21)$$

277 where Eq. (19) introduces into Eq. (17) an additional term that combines the  
 278 branch vector between  $i$  and  $p$ ,  $\vec{b}_i^p$ , with the velocity gradient  $\nabla \vec{V}^p$ . The  
 279 latter is stored for every material point after being computed in a previous  
 280 iteration as:

$$\nabla \vec{V}^p = \sum_i \overline{\nabla N}_i(\vec{x}^p) \otimes \vec{V}^i \quad (22)$$

281 A rotational ensemble motion on a set of material points is an example of  
 282 an affine velocity field, with an antisymmetric  $\nabla \vec{V}^p$ , and the limitations of the  
 283 initial Eq. (17) for projecting velocity fields naturally appear in this case, with  
 284 a loss of angular momentum that is actually due to the lumped mass approach  
 285 of Eqs. (4)-(6) and (17) [16, 17, 18]. While keeping the convenient lumped  
 286 mass transport Eq. (6), it has been proposed in [17] as an APIC formulation  
 287 to explicitly consider angular momentum in the transport Eq. (17) through  
 288 additional terms at the material point level, namely:

$$\mathbf{V}^G = \mathbf{P}_m^G \mathbf{V}^P + \mathbf{U}^G \quad (23)$$

$$(\mathbf{U}^G)_{ik} = \sum_p \sum_j (\mathbf{P}_m^G)_{ip} (\mathbf{C}^p)_{jk} (\mathbf{b}_i^p)_j \quad (24)$$

$$\mathbf{C}^p = \mathbf{B}^p (\mathbf{D}^p)^{-1} \quad (25)$$

$$\mathbf{B}^p = \sum_i N_i(\vec{x}^p) \widetilde{\mathbf{V}}^i \otimes \vec{b}_i^p \quad (26)$$

$$\mathbf{D}^p = \sum_i N_i(\vec{x}^p) \vec{b}_i^p \otimes \vec{b}_i^p \quad (27)$$

289 Here, the affine augmentation of Eqs. (23)-(27) (with a more complex ver-  
 290 sion proposed in [18]) embeds every material point with an angular velocity-like  
 291 quantity  $\mathbf{C}^p$  which is computed from an angular momentum-like quantity  
 292  $m^p \mathbf{B}^p$  and an inertia-like quantity  $m^p \mathbf{D}^p$  (or its inverse). That angular

293 velocity-like quantity is then combined in Eq. (24) with  $\vec{b}_i^p$ , giving an addi-  
 294 tional rotational term,  $\mathbf{U}^p$ , in the velocity field to be mapped into the grid  
 295 in Eq. (23). In such an APIC formulation, both matrices  $\mathbf{B}^p, \mathbf{D}^p$  are stored  
 296 for every material point in order to form  $\mathbf{C}^p$  and Eq. (26) actually corre-  
 297 sponds to an end-of-step update (from updated nodal velocities  $\widetilde{\vec{V}}^i$  precised in  
 298 Section 3.3) for  $\mathbf{B}^p$ , providing its value that will be used in a subsequent step.

299 A proper initialization of  $\nabla \vec{V}^p$ , for the Taylor formulation, or  $\mathbf{B}^p$ , for  
 300 the APIC formulation, is however critical in case of assigned initial velocities  
 301 to material points. For this purpose, the initial value of  $\nabla \vec{V}^p$  is determined  
 302 by repeating Eqs. (19) to (22) until  $\nabla \vec{V}^p$  converges. Because it can usually  
 303 be observed that the APIC  $\mathbf{C}^p$  is numerically similar to velocity gradient, as  
 304 suggested by the formal similarities between Eq. (20) and (24), the former is  
 305 incidentally initialized to the same value of the latter.

### 306 3.3 From nodal acceleration to particle kinematics

307 Whether  $\vec{V}^i$  is affine-augmented or not, an updated nodal velocity field  $\widetilde{\vec{V}}^i$   
 308 is subsequently obtained after solving Eq. (1) in its discrete form Eq. (4) for  
 309 nodal acceleration and applying a simple, explicit, finite difference scheme in  
 310 time:

$$\widetilde{\vec{V}}^i = \vec{V}^i + \vec{a}^i \cdot \Delta t \quad (28)$$

311 Eq. (28) naturally leaves aside possible grid points serving as Dirichlet bound-  
 312 ary conditions with constant velocity components ( $a_k^i = 0$  and  $V_k^i$  fixed to  
 313 a chosen  $V_k^i|_{\text{lim}}$  for one or several directions  $k$ ). Time step  $\Delta t$  is computed  
 314 as a fraction of the characteristic time obtained with the material and mesh  
 315 parameters:

$$\Delta t = a_\tau l_{\text{cell}} \sqrt{\frac{\rho}{E}} \quad (29)$$

316 with  $l_{cell}$  the size of a mesh element,  $\rho$  the mass density of the material,  
 317  $E$  its Young modulus for the subsequent examples considering, at least in  
 318 part, Hooke's law, and  $a_\tau$  a proportion coefficient. The latter is determined  
 319 empirically in each specific case, in order to ensure a stable simulation.

320 In line with the absence of nodal history, the updated velocity field  $\tilde{\mathbf{V}}^i$  is  
 321 only temporary, before being overwritten at the beginning of the next iteration  
 322 with the one transported from the material points with one or another mapping  
 323 among Eqs. (17), (19) or (23). It rules however material point kinematics  
 324 during current time step, where different MPM strategies can again be found  
 325 for this purpose. Consistently with the above use of a mapping matrix  $\mathbf{P}_m^{\star G}$ ,  
 326 this procedure can be considered as equivalent to solving the following equation  
 327 for  $\mathbf{V}^{\mathbb{P},new}$ :

$$\begin{aligned} \mathbf{P}_m^{\star G} \mathbf{V}^{\mathbb{P},new} &= \tilde{\mathbf{V}}^{\mathbb{G}} \\ &= \begin{cases} \mathbf{P}_m^{\star G} \mathbf{V}^{\mathbb{P}} + \mathbf{W}^{\mathbb{G}} + \mathbf{a}^{\mathbb{G}} \Delta t & \text{for TPIC} \\ \mathbf{P}_m^{\star G} \mathbf{V}^{\mathbb{P}} + \mathbf{U}^{\mathbb{G}} + \mathbf{a}^{\mathbb{G}} \Delta t & \text{for APIC} \\ \mathbf{P}_m^{\star G} \mathbf{V}^{\mathbb{P}} + \mathbf{a}^{\mathbb{G}} \Delta t & \text{for other strategies} \end{cases} \end{aligned} \quad (30)$$

328 In the general case with  $N_{mp} \neq N_{gp}$  and a non-square  $\mathbf{P}_m^{\star G}$ , the set of  
 329 solutions to Eq. (30) can be infinite, justifying the existence of several strate-  
 330 gies. The most direct solution to Eq. (30) was inspired by the PIC method  
 331 and essentially maps the velocity from grid points back to the material points  
 332 with Eq. (11), leading to:

$$\mathbf{V}_{PIC}^{\mathbb{P},new} = \mathbf{G}^{\star P} \tilde{\mathbf{V}}^{\mathbb{G}} \quad (31)$$

$$\mathbf{x}_{PIC}^{\mathbb{P},new} = \mathbf{x}^{\mathbb{P}} + \mathbf{V}_{PIC}^{\mathbb{P},new} \times \Delta t \quad (32)$$

333 Although this strategy is straightforward to implement, it is known to improp-  
 334 erly filter the velocity which leads to an unrealistic damping [22, 40]. One  
 335 can note for instance that Eq. (31) erases the individual memory for material  
 336 point velocity one could expect since  $\mathbf{V}^{\mathbb{P},new}$  is not directly integrated from  
 337  $\mathbf{V}^{\mathbb{P}}$ . Moreover, it has been demonstrated in [22] that  $\mathbf{V}_{PIC}^{\mathbb{P}}$  is the solution to  
 338 Eq. (30) that minimizes  $\mathbf{V}^{\mathbb{P}}$ .

339 Aiming to achieve a better energy conservation than PIC, the FLIP method  
 340 [9] has been proposed with alternative time velocity increments from the same  
 341 nodal acceleration. In a FLIP approach, the latter is actually transported  
 342 into  $\mathbb{P}$  towards an incremental computation of the material points' velocity,  
 343 reinstating a direct, individual, link between  $\vec{V}^{p,new}$  and  $\vec{V}^p$ :

$$\mathbf{V}_{FLIP}^{\mathbb{P},new} = \mathbf{V}^{\mathbb{P}} + \mathbf{G}^{\rightarrow P} \mathbf{a}^{\mathbb{G}} \times \Delta t \quad (33)$$

344 Thinking in terms of solution to Eq. (30), it was established in [22] that  
 345  $\mathbf{V}_{FLIP}^{\mathbb{P}}$  is the solution which minimizes the fluctuations in  $\mathbf{V}^{\mathbb{P}}$ , i.e. acceleration.  
 346 However, FLIP still uses the same Eq. (32) as PIC to displace the material  
 347 points, which can be recast as:

$$\begin{aligned} \mathbf{x}^{\mathbb{P},new} &= \mathbf{x}^{\mathbb{P}} + \mathbf{V}_{PIC}^{\mathbb{P},new} \times \Delta t \\ &= \mathbf{x}^{\mathbb{P}} + \mathbf{G}^{\rightarrow P} \left( \mathbf{V}^{\mathbb{G}} + \mathbf{a}^{\mathbb{G}} \Delta t \right) \Delta t \end{aligned} \quad (34)$$

348 Since  $\mathbf{G}^{\rightarrow P} \mathbf{V}^{\mathbb{G}} \neq \mathbf{V}^{\mathbb{P}}$ , one could consider from comparing Eq. (34) and (33) the  
 349 FLIP scheme implies an unnatural description of the material points motion  
 350 in the sense that the time derivative of  $\vec{x}_{FLIP}^p$  is different from the FLIP  
 351 material point velocity (even though this was intended in [9]):

$$\vec{V}_{FLIP}^p \neq \frac{d\vec{x}_{FLIP}^p}{dt} \quad (35)$$

352 As such,  $\vec{V}_{PIC}^p$  in Eq. (32) has been replaced in [41] with  $\vec{V}_{FLIP}^p$ , forming  
 353 another NFLIP strategy (Naturally modified FLIP), as coined by [19]:

$$\mathbf{x}_{NFLIP}^{\mathbb{P},new} = \mathbf{x}^{\mathbb{P}} + \mathbf{V}_{FLIP}^{\mathbb{P},new} \times \Delta t \quad (36)$$

354 As previously mentioned, FLIP and NFLIP conserve better energy than  
 355 PIC but they are more prone to instabilities. A common practice is thus to  
 356 blend FLIP or NFLIP with PIC in order to make the simulation more stable  
 357 [13, 42, 43, 44], based on a blending parameter  $P_{PIC}$ . These blended strategies  
 358 are denoted FLIPX and NFLIPX, where  $X = 1 - P_{PIC}$  indirectly highlights  
 359 the proportion of PIC. Namely, a FLIPX velocity is given by:

$$\begin{aligned} \mathbf{V}_{FLIPX}^{\mathbb{P},new} &= X \mathbf{V}_{FLIP}^{\mathbb{P},new} + (1 - X) \mathbf{V}_{PIC}^{\mathbb{P},new} \\ &= (1 - P_{PIC}) \mathbf{V}_{FLIP}^{\mathbb{P},new} + P_{PIC} \mathbf{V}_{PIC}^{\mathbb{P},new} \end{aligned} \quad (37)$$

360 Such a FLIPX velocity would also be used by the NFLIPX strategy to  
 361 compute new positions for the material points, similarly to NFLIP Eq. (36):

$$\mathbf{x}_{NFLIPX}^{\mathbb{P},new} = \mathbf{x}^{\mathbb{P}} + \mathbf{V}_{FLIPX}^{\mathbb{P},new} \times \Delta t \quad (38)$$

362 On the other hand, it has been proposed in [22, 23, 45], that the blended  
 363 velocity update of Eq. (37) should be accompanied with deeper changes in  
 364 position updates that would eventually conform neither Eq. (34) nor Eq. (38)  
 365 but include a second order term.

366 Such a second order position update is actually found in the XPIC( $m$ )  
 367 strategy [22], which first aims for a definition of the updated velocity field  
 368  $\mathbf{V}^{\mathbb{P},new}$  that shows minimal variations from its previous value, in a smoothed

369 version  $\mathbf{V}_{sm}^{\mathbb{P}}$ :

$$\mathbf{V}_{sm}^{\mathbb{P}} = (\mathbf{I} - (\mathbf{I} - \mathbf{G}^*P\mathbf{P}_m^*G)^{m-1})\mathbf{V}^{\mathbb{P}} \quad (39)$$

370 with  $\mathbf{I}$  the  $(N_{mp} \times N_{mp})$  identity matrix and  $m$  a chosen smoothing parameter.

371 The resulting solution to Eq. (30) is then [22]:

$$\mathbf{V}_{XPIC(m)}^{\mathbb{P},new} = \mathbf{V}^{\mathbb{P}} - (\mathbf{I} - \mathbf{G}^*P\mathbf{P}_m^*G)^m\mathbf{V}^{\mathbb{P}} + \mathbf{G}^*P\mathbf{a}^G \times \Delta t \quad (40)$$

372 Material points are then moved using a second order time integration scheme:

$$\mathbf{x}_{XPIC(m)}^{\mathbb{P},new} = \mathbf{x}^{\mathbb{P}} + \mathbf{G}^*P\tilde{\mathbf{V}}^G\Delta t - \frac{1}{2}\left(\mathbf{G}^*P\mathbf{a}^G\Delta t^2 + (\mathbf{I} - \mathbf{G}^*P\mathbf{P}_m^*G)^m\mathbf{V}^{\mathbb{P}}\Delta t\right) \quad (41)$$

373 Note that with  $m = 1$ , a XPIC(1) strategy is equivalent to PIC, see Eq. (40) vs

374 Eq. (31) for velocity, and that the corresponding position update of Eq. (41)

375 has latter been further modified in [23] and be proposed as an optimal position

376 update for a PIC framework.

377 Most of the motion integration strategies mentioned above have been

378 declined into their affine-augmented version (APIC, AFLIP and ANFLIP,

379 AFLIPX and ANFLIPX) just by using Eq. (23) to express the grid points

380 velocity  $\mathbf{V}^G$  depending on the material points velocity  $\mathbf{V}^{\mathbb{P}}$ . Similarly, a TFLIP

381 counterpart to TPIC is defined as using Eq. (19) for initializing grid velocities

382 from material point ones, before using all other FLIP equations.

383 Since the FLIP logic is to incrementally update the material point velocity

384 rather than over-writing it through a grid interpolation, one could expect sim-

385 ilar time update equations for the matrices  $\mathbf{B}^p, \mathbf{D}^p, \mathbf{C}^p$  for AFLIP or  $\nabla\vec{V}^p$

386 for TFLIP. Since it is instead chosen to use the same equations than APIC or

Motion integration operations Motion integration strategy	$\mathbf{V}^G$ measure from current $\mathbf{V}^P$	$\mathbf{V}^P$ update from $\mathbf{a}^G$ , i.e. updated $\mathbf{V}^G$	$\mathbf{x}^P$ update
PIC [8]	$\mathbf{P}_m^{*G} \mathbf{V}^P$ Eq. (17)	$\mathbf{V}_{PIC}^P$ Eq. (31)	$\mathbf{x}_{PIC}^P$ Eq. (32)
APIC [17]	$\mathbf{P}_m^{*G} (\mathbf{V}^P + \mathbf{U}^P)$ Eq. (23)		
TPIC [20, 21]	$\mathbf{P}_m^{*G} (\mathbf{V}^P + \mathbf{W}^P)$ Eq. (19)		
FLIP [9, 10]	$\mathbf{P}_m^{*G} \mathbf{V}^P$ Eq. (17)	$\mathbf{V}_{FLIP}^P$ Eq. (33)	$\mathbf{x}_{NFLIP}^P$ Eq. (36)
AFLIP [17, 19]	$\mathbf{P}_m^{*G} (\mathbf{V}^P + \mathbf{U}^P)$ Eq. (23)		
TFLIP [20]	$\mathbf{P}_m^{*G} (\mathbf{V}^P + \mathbf{W}^P)$ Eq. (19)		
NFLIP [41, 19]	$\mathbf{P}_m^{*G} \mathbf{V}^P$ Eq. (17)	$X \mathbf{V}_{FLIP}^P + (1 - X) \mathbf{V}_{PIC}^P$ Eq. (37)	$\mathbf{x}_{NFLIPX}^P$ Eq. (38)
ANFLIP	$\mathbf{P}_m^{*G} (\mathbf{V}^P + \mathbf{U}^P)$ Eq. (23)		
FLIPX	$\mathbf{P}_m^{*G} \mathbf{V}^P$ Eq. (17)		
AFLIPX	$\mathbf{P}_m^{*G} (\mathbf{V}^P + \mathbf{U}^P)$ Eq. (23)	$\mathbf{V}_{XPIC(m)}^P$ Eq. (40)	$\mathbf{x}_{XPIC(m)}^P$ Eq. (41)
NFLIPX	$\mathbf{P}_m^{*G} \mathbf{V}^P$ Eq. (17)		
ANFLIPX	$\mathbf{P}_m^{*G} (\mathbf{V}^P + \mathbf{U}^P)$ Eq. (23)		
XPIC( $m$ ) [22]	$\mathbf{P}_m^{*G} \mathbf{V}^P$ Eq. (17)		

**Table 1:** Designations of the different MPM motion integration strategies with their chosen underlying equations

387 TPIC to express those matrices ( $\mathbf{V}^P$ , with a direct transport from the grid, a bet-  
 388 ter terminology could adopt AFLIP/PIC and TFLIP/PIC notations, which is  
 389 nevertheless not done here for the sake of simplicity.

390 Table 1 summarizes all these motion integration strategies.

### 391 3.4 Stress update scheme

392 Following [10, 46] and for the sake of simplicity, a small deformations (notwith-  
 393 standing possible large displacements) strain tensor  $\boldsymbol{\epsilon}$  is defined from its rate

394  $\dot{\epsilon}$ :

$$\dot{\epsilon} = \frac{1}{2} \left( \nabla \vec{V} + (\nabla \vec{V})^T \right) \quad (42)$$

395 and the corresponding increment  $d\epsilon$  is related with  $d\sigma$  through a material-  
 396 specific constitutive relation for non-viscous solids, taking into account history-  
 397 dependent variables when necessary. If necessary, large strain-compliant more  
 398 general formulations based on deformation gradient and, possibly, objective  
 399 stress rate (e.g., of Jaumann type) can be found for instance in [12, 23, 24, 25].  
 400 The MPM algorithm naturally applies the incremental stress update at each  
 401 material point, after computing a finite  $\Delta\epsilon^p$  from nodal velocities  $\vec{V}^i$ :

$$\Delta\epsilon^p = \dot{\epsilon}^p \Delta t \quad (43)$$

$$\dot{\epsilon}^p = \frac{1}{2} \sum_i \left( \mathbf{A}^{ip} + \mathbf{A}^{ipT} \right) \quad (44)$$

$$\mathbf{A}^{ip} = \overrightarrow{\nabla N}_i(\vec{x}^p) \otimes \vec{V}^i \quad (45)$$

402 with  $\mathbf{A}^{ip}$  defined for each pair of grid point  $i$  and material point  $p$ . Due to the  
 403 intricate relations between nodal velocity or even material point velocity and  
 404 time increments of material positions, it is again to note that such a strain  
 405 tensor may not be directly interpreted from the relative displacements among  
 406 material points.

407 The location of that stress update in the sequence of the MPM algorithm  
 408 is a matter of choice, with no prior justification for an execution before or  
 409 after the internal force computation of Eq. (8). The two immediate choices of  
 410 updating stresses before, resp. after, solving the equation of motion have been  
 411 denoted USF (“Update Stress First”), resp. USL (“Update Stress Last”) and  
 412 analysed in [38] for their consequences on energy conservation. It was shown

413 therein the USF scheme is more likely to conserve energy in average through  
 414 a mutual cancellation of two sources of errors that affect both material points  
 415 kinetic and strain energies at each MPM iteration. It has also been observed  
 416 herein during preliminary simulations that USF is better suited to use with  
 417 linear shape functions, unlike USL which exacerbates cell crossing instability  
 418 issues in this case.

419 The stress update scheme was further studied in [47], where a variation of  
 420 the USL scheme initially proposed in [48] was coined MUSL (“Modified Update  
 421 Stress Last”) and found to be very similar to the USF scheme in its results. In  
 422 its definition, the MUSL scheme executes the stress update likewise to USL,  
 423 after relating internal forces to nodal acceleration in Eq. (8), but uses for this  
 424 purpose a strain increment computed from updated nodal velocities, with  $\tilde{\vec{V}}^i$   
 425 replacing  $\vec{V}^i$  in Eq. (45). In line with these observations, [47] proposed the  
 426 USAVG (“Update Stress Averaged”) scheme which conserves almost perfectly  
 427 the energy, at the cost of computing the material behaviour twice in the same  
 428 MPM iteration.

429 In the present manuscript, a USF formulation is adopted unless otherwise  
 430 specified, including an updated stress value in Eq. (8). While a USL choice  
 431 may be more usual in the literature, the USF scheme is herein chosen since  
 432 it was shown in [38] that it can be completely conservative energy-wise, while  
 433 the USL scheme is systematically strictly dissipative.

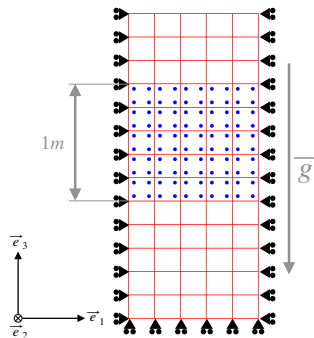
## 434 4 Energy conservation on heuristic stiff examples 435 for different motion integration strategies

436 This section determines the influence of the different strategies detailed in  
 437 the previous section on the MPM capacity to conserve energy, for two simple

438 cases with basic configurations that enable a ground-truth comparison. Pro-  
 439 posed results are necessarily affected by the explicit first order time integration  
 440 scheme (except for the XPIC case) chosen in the above equations for velocity  
 441 or positions update, e.g., Eq. (28). Other integration schemes, e.g., proposed  
 442 in [18], especially if symplectic, would improve energy conservation.

#### 443 4.1 Energy conservation in translation

444 The first case extends a similar analysis in [16, 18] (that is also directly  
 445 considered in Appendix B.1) by simulating a bouncing cube conforming a  
 446 translational motion under gravity, with a gravitational acceleration  $g = 9.81$   
 447  $m/s^2$ , and illustrated in Figure 2 for its initial configuration. The mesh con-  
 448 tains  $5 \times 5 \times 13$  cubic cells along the different axes, each with a side of  $l_{cell} = 20$   
 449  $cm$ . Each cell located between  $x_3 = 1 m$  and  $x_3 = 2 m$  contains initially  $2^3 = 8$   
 450 material points located at the Gauss-Legendre integration points, for a total  
 451 of  $8 \times 125 = 1000$  material points. The normal velocity is imposed to zero on  
 452 all grid points on the left, right, front, back and bottom boundaries.



**Fig. 2:** Simulation setup of the bouncing cube example (plane view)

453 Hyperelastic Hooke's law is assigned to the solid material (e.g., to describe  
 454 the deformations of the cube upon impact on the floor), with Young's modulus  
 455 being set to  $E = 12.84 \text{ MPa}$  and Poisson's ratio to  $\nu = 0.16$ . The density of  
 456 the material is set to  $\rho = 1748 \text{ kg} \times \text{m}^{-3}$ , meaning that each material point  
 457 will have a mass of  $m_p \approx 1.75 \text{ kg}$ . Using Eq. (29) along with  $a_\tau \approx 5 \times 10^{-2}$  or  
 458  $a_\tau \approx 5 \times 10^{-3}$ , the time step is chosen as  $\Delta t = 1.17 \times 10^{-4} \text{ s}$  or  $\Delta t = 1.17 \times 10^{-5}$   
 459  $\text{s}$ .

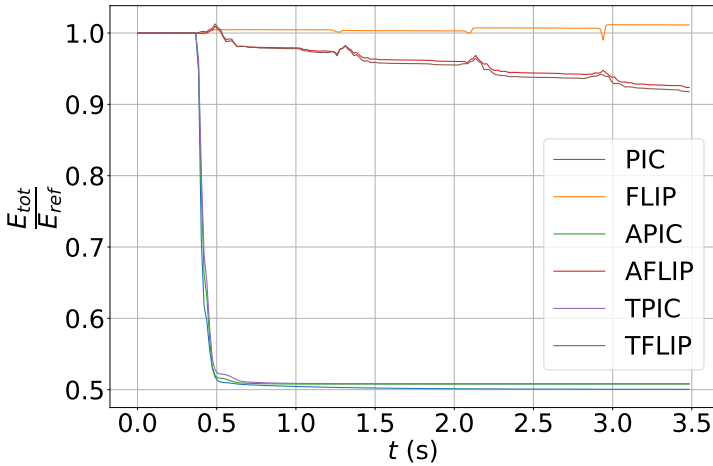
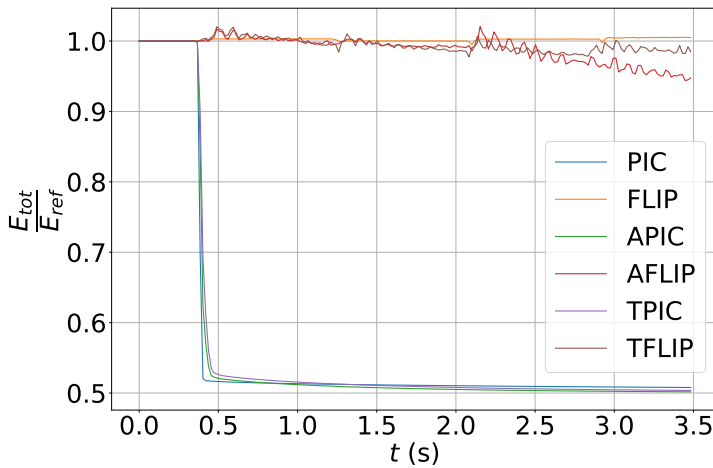
460 In line with the conservative nature of the problem, energy should theoreti-  
 461 cally conserve and just converts during time between elastic strain energy  $E_{el}$ ,  
 462 kinetic energy  $E_k$  and gravitational energy  $E_p$  (with  $x_3 = 0$  serving as refer-  
 463 ence for  $E_p = 0$ ). Measuring those quantities over all material points and using  
 464 Einstein's convention for summing over repeated indices, we namely have:

$$E_{el} = \sum_{p \in \{p\}} \frac{v^p}{2} (\sigma_{ij} \epsilon_{ij}) \quad (46)$$

$$E_k = \sum_{p \in \{p\}} \frac{1}{2} m^p \|\vec{V}^p\|^2 \quad (47)$$

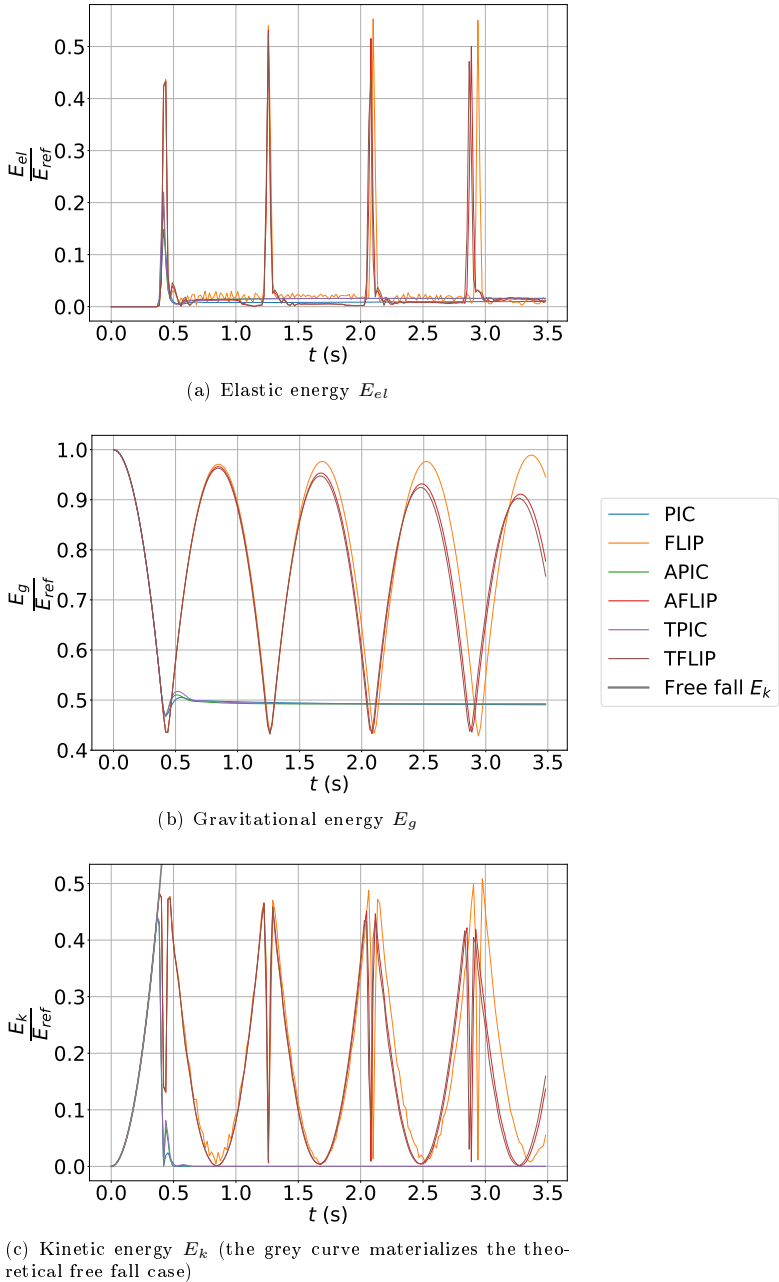
$$E_g = \sum_{p \in \{p\}} m^p g x_3^p \quad (48)$$

465 The following Figures 3 and 4 illustrate the obtained energy balance in  
 466 MPM for different motion integration strategies, namely PIC, FLIP, APIC,  
 467 AFLIP, TPIC, and TFLIP; when using the USF stress update scheme (USL  
 468 results being also presented for the same simulations in Appendix A). The  
 469 NFLIP formulation was also considered during preliminary simulations with a  
 470 coarser mesh and linear shape functions and was observed to yield unrealistic  
 471 results (e.g., segregation of material points within the cube cells for various  
 472 time steps ranging from  $a_\tau \approx 5 \times 10^{-6}$  to  $5 \times 10^{-2}$ ), as already reported in [23].

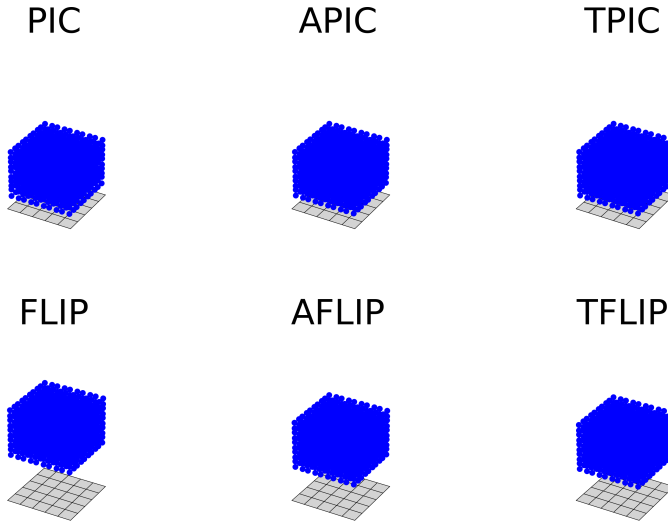
(a)  $a_r = 5 \times 10^{-2}$ (b)  $a_r = 5 \times 10^{-3}$ 

**Fig. 3:** Total energy for the bouncing cube example simulated with different motion integration strategies for different time steps ( $E_{ref} \approx 25.7 \text{ kJ}$ )

473 Figure 3 first shows the total energy  $E_{tot} = E_{el} + E_k + E_g$  relative to  
 474 its initial value of  $E_{ref} \approx 25.7 \text{ kJ}$ , for the two different time steps. At first,  
 475 during the initial free fall phase with neither strains nor stress, all results  
 476 are strictly equivalent and theoretically correct, whatever the motion integra-  
 477 tion strategy. After the first contact with the floor, all PIC-based cases (PIC,  
 478 APIC and TPIC) dissipate so much energy that the cube does not bounce, no



**Fig. 4:** Energy balance for the bouncing cube example simulated with different MPM motion integration strategies ( $E_{ref} \approx 25.7 \text{ kJ}$ ,  $a_\tau = 5 \times 10^{-2}$ )



**Fig. 5:** Material points positions at  $t \approx 3.5$  s ( $E_{ref} \approx 25.7$  kJ, see a corresponding video as a supplementary material)

479 matter the time step used. On the other hand, FLIP-based strategies appear  
 480 to be much more conservative, allowing for several bounces. In more details,  
 481 FLIP conserves energy almost perfectly, slightly increasing it, while AFLIP  
 482 and TFLIP let  $E_{tot}$  decrease by approximately 3% between each bounce (these  
 483 strategies are shown to be equivalent to FLIP when using the USL scheme in  
 484 Appendix A), although these defects are limited when using a lower time step.

485 Figure 4 then discriminates between the various energy terms in  $E_{tot}$  when  
 486 using  $a_\tau = 5 \times 10^{-2}$ . PIC-induced artificial damping therein appears both  
 487 in terms of elastic energy  $E_{el}$  (Figure 4 (a)) with no energy oscillations after  
 488 impacting the floor, and in terms of gravitational energy  $E_g$  (Figure 4 (b)) with  
 489 the height of the cube reaching its final value after the first contact with the  
 490 floor. With FLIP-based strategies, the cube continues to deform even during  
 491 its ascending phase, especially for FLIP, and the material points almost bounce

back to their initial heights. In fact, after the last bounce, the FLIP set of material points bounces higher than its previous peak position, in correspondence to  $E_{tot}$  observations. As for  $E_k$  (Figure 4 (c)), all motion integration strategies reproduce the theoretical velocity obtained during the free fall part of the simulation. Then, all FLIP-based strategies give a similar variation rate of  $E_k$ , even though it is shifted in time because of the different bouncing altitudes the cube reaches. One can lastly note that, due to the MPM discretization and the extended support of present cubic B-spline shape functions (two cells wide for the boundary case), impacts and bounces occur as soon as the lowest material points reach  $x_3 = 0.4 m$ .

On this very simple case involving translation only, FLIP thus appears a possible best candidate for conserving energy (both with present USF choice or with USL, see Appendix A).

## 4.2 Conservation of rotational kinetic energy and angular momentum

### 4.2.1 Simulation description

A second simplified case study includes a combination of rotational and linear motion, without gravity, which extends a somewhat similar previous analysis in [18] (also directly considered in Appendix B.2). Basically, a cube with a side  $l_{cube} = 3 m$  is thrown in space, i.e. is given an initial velocity with both a linear and rotational motion, with

$$\vec{V}_c = \frac{V_c}{\sqrt{3}}(\vec{e}_1 + \vec{e}_2 + \vec{e}_3) \quad (49)$$

$$V_c = 1.71 \text{ cm/s}$$

513 the linear velocity of the cube, computed so that the cube stays within the  
514 mesh during the whole simulation, and

$$\vec{\omega} = \omega \sqrt{\frac{2}{3}} (0.5 \vec{e}_1 + 0.5 \vec{e}_2 + \vec{e}_3) \quad (50)$$

$$\omega = 0.108 \text{ rad/s}$$

515 its angular velocity, computed so that the cube performs several revolutions  
516 during the simulation.

517 The whole space domain is covered by a mesh of cubic elements with a side  
518 length of  $l_{cell} = l_{cube}/3 = 1 \text{ m}$ , the solid cube thus spans over  $l_{cube}^3 = 3^3 =$   
519  $27$  mesh cells. Initially, the cube is located in a corner of the domain while  
520 including  $2^3$  material points per cell which are regularly spaced in the cells,  
521 and have initial velocities assigned in accordance with the desired linear and  
522 angular velocities mentioned in the above. The time step  $\Delta t$  is computed with  
523 Eq. (29) from the simulation parameters and a variable  $a_\tau$  (equal to  $5 \times 10^{-2}$   
524 unless otherwise specified, providing a time step of  $5.83 \times 10^{-4} \text{ s}$ ).

525 While the same elastic material properties are used here as in previous  
526 Section 4.1, inertial (centrifuge) effects are small enough to consider the cube  
527 as rigid. As a matter of fact, inertial effects can be quantified from the following  
528 dimensionless number  $C_a$  which is considerably small:

$$C_a = \frac{\rho \omega^2 R^2}{E} \approx 1.07 \cdot 10^{-5} \quad (51)$$

529 with  $R = \frac{\sqrt{3}}{2} l_{cube}$  the radius of the circumscribed sphere. While being analog  
530 to the Cauchy number in fluid mechanics,  $C_a$  is built from  $\rho \omega^2 R^2$  that rules  
531 stress quantities in a rotating elastic solid [49] and material stiffness  $E$ .

532 Strains being theoretically negligible, use can be made of rigid bodies  
 533 equations, e.g., Eq. (52) below, with the consideration of angular momentum  
 534  $\mathbf{I}\vec{\omega}$  (where  $\mathbf{I}$  is the inertia matrix), in order to theoretically predict the move-  
 535 ment. Since angular momentum is here constant in a body-attached local frame  
 536 due to the absence of external loads, while  $\mathbf{I}$  for a cube is a spherical tensor  
 537 with a constant expression in any frame, angular velocity can also directly be  
 538 considered as constant with its expression in global frame such as given in  
 539 Eq. (50). The position of any material point  $p$  can thus be predicted over time  
 540 instants being separated by a given  $\Delta t$  and corresponding to the MPM time  
 541 discretization as follows:

$$\vec{V}^p = \vec{V}_c + \vec{\omega} \times (\vec{x}^p - \vec{x}^c) \quad (52)$$

$$\vec{x}^{p,new} = \vec{x}^p + \vec{V}^p \Delta t \quad (53)$$

542 with  $\times$  denoting the cross product when being applied like in Eq. (52) to  
 543 two vectors,  $\vec{x}^p$  and  $\vec{x}^{p,new}$  corresponding to the positions at two successive  
 544 integration times, and

$$\vec{x}^c = \frac{\sum_p \vec{x}^p}{N_{mp}} \quad (54)$$

545 the position of the center of mass of the cube.

546 These will be compared with actual MPM results, together with the consid-  
 547 eration of the traveled distance  $d_c = \|\vec{x}^c(t) - \vec{x}^c(t_0)\|$ , with a corresponding  
 548 relative error as:

$$D_{err} = \left| \frac{d_c}{V_c t} - 1 \right| \quad (55)$$

549 The total angular momentum  $L^{tot}$  is also computed on the material points  
 550 as per the following Eq. (56):

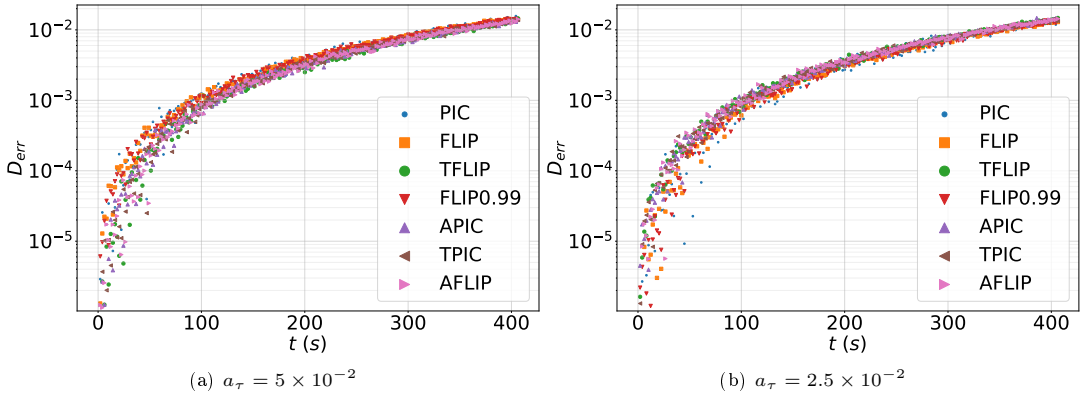
$$L^{tot} = \left\| \sum_p m^p (\vec{x}^p - \vec{x}^c) \times \vec{V}^p \right\| \quad (56)$$

551 in order to check to which extent MPM results do conform conservation of  
 552 angular momentum. The Eq. (56), also used in, e.g., [15] is chosen to be consis-  
 553 tent with the present “material point” point of view on the method discussed  
 554 in previous Section 2.2, by considering that material points carry only linear  
 555 velocity, and also consistent with the kinetic energy Eq. (47). On the other  
 556 hand, with a “material domain” point of view, it could be considered that  
 557 material points are centers of mass of some finite domain with a space-variable  
 558 velocity field that could define a material point angular velocity and addi-  
 559 tional terms would enter Eq. (56). Those additional terms would account from  
 560 angular momentum contributions corresponding to this material domain-scale  
 561 velocity field, which is a choice done in, e.g., [17, 18], in connection with the  
 562 consideration of an affine velocity field for the APIC transfers, and should in  
 563 principle be accompanied with corresponding modifications to kinetic energy  
 564 expression, necessitating additional terms in Eq. (47).

565 Without claiming for exhaustivity, the PIC, FLIP, FLIP0.99, APIC,  
 566 AFLIP, TPIC and TFLIP motion integration strategies are herein tested.

## 567 4.2.2 Results

568 For what concerns first the (unconstrained) linear motion, and similarly to  
 569 the free fall part of the previous case, all motion integration strategies provide  
 570 accurate predictions as shown in Figure 6 where the traveled distance exhibits  
 571 a negligible error with respect to its expected value.



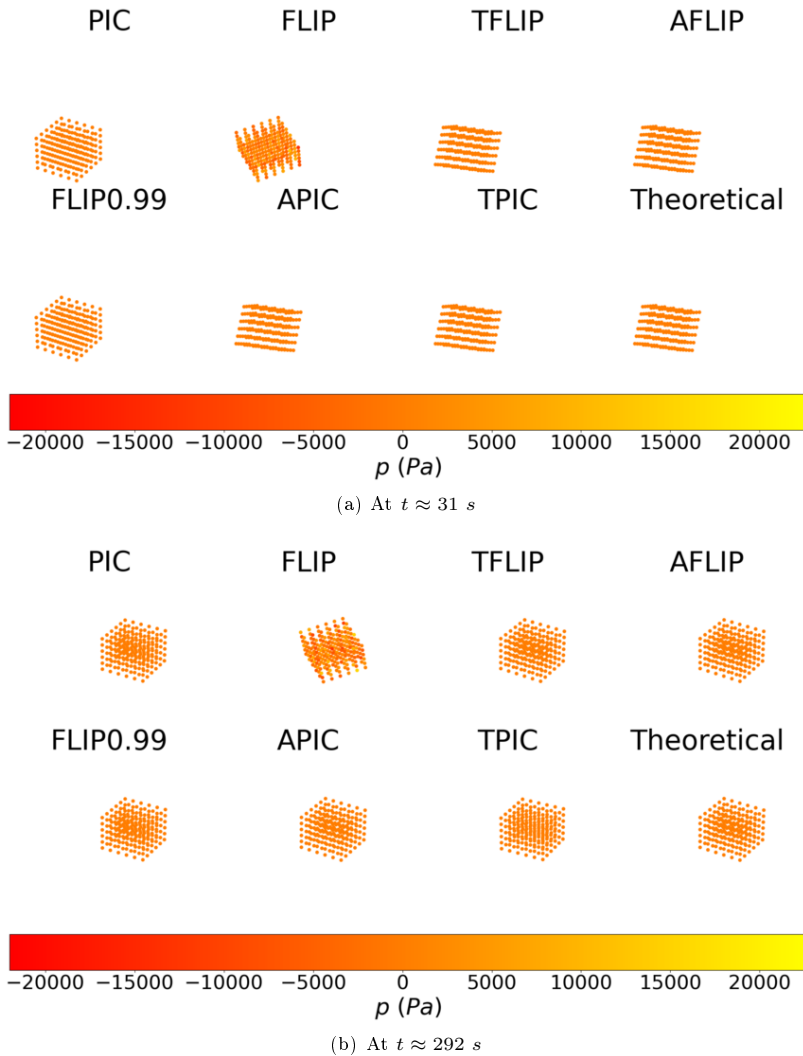
**Fig. 6:** Relative error  $D_{err}$  on the distance traveled by the center of the rotating cube

572 However, inspecting the individual movements of material points (Figure 7)  
 573 or the time evolution of angular momentum (Figure 8) recalls how rotational  
 574 motion gets lost for all strategies others than the affine-augmented or Taylor-  
 575 based ones.

576 In particular, PIC-based non-affine-augmented strategies (PIC and  
 577 FLIP0.99) cancel  $L^{tot}$  at the very beginning of the simulation (virtually im-  
 578 mediately for PIC), in line with PIC Eq. (31) which induces a drastic averaging  
 579 of velocity field that is unable to conserve the theoretical spatially-variable  
 580 velocity field inherent to rotational motion.

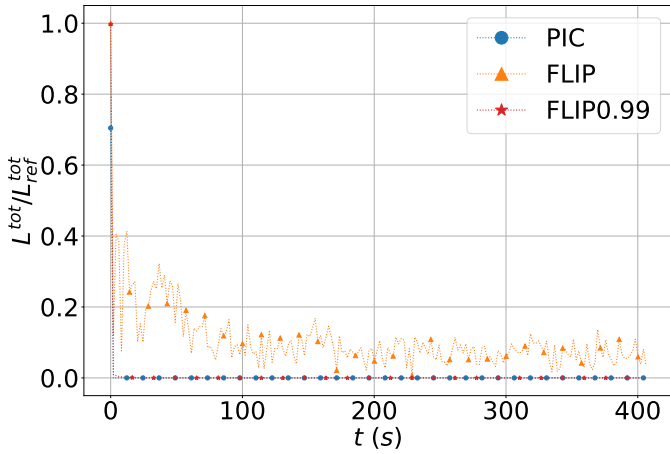
581 The FLIP strategy allows the cube to somewhat preserve a slight rotation  
 582 but it is clearly not the one imposed initially (Figure 7) and the angular  
 583 momentum eventually reaches a small value after evolving quite erratically  
 584 (Figure 8(a)). While those observations also apply even with a smaller time  
 585 step ( $a_\tau = 2.5 \times 10^{-2}$  vs  $5 \times 10^{-2}$ ), the Figure 26 in Appendix B.2 shows how  
 586 FLIP conserves much better angular momentum if the mesh is fine enough.

587 On the other hand, affine augmented strategies show themselves to be much  
 588 more conservative regarding this rotational motion, though with influences of

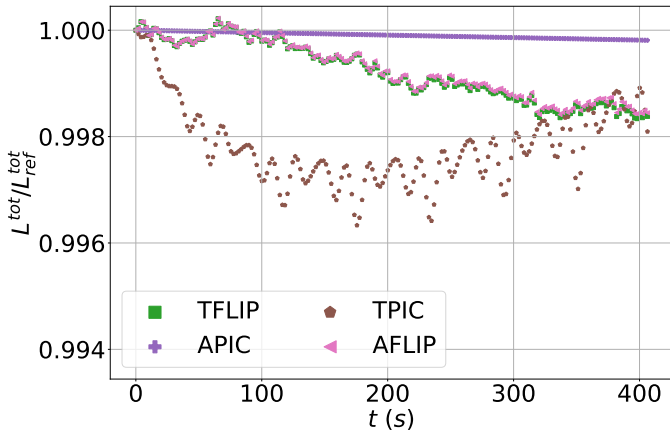


**Fig. 7:** Material point positions in the case of the rotating cube (see corresponding video as a supplementary material)

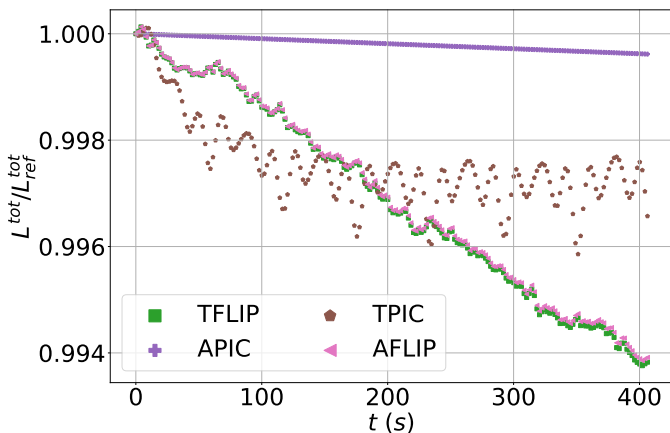
589 both the time step (Figure 8(b) and (c)) and the magnitude of the angular  
 590 velocity  $\vec{\omega}$  (Figure 9). More precisely, for the default case of  $\|\vec{\omega}\| = 0.108 \text{ rad/s}$ ,  
 591 APIC loses a minuscule amount of angular momentum at a rate of approxi-  
 592 mately  $4.6 \times 10^{-5} \text{ \%}/s$  for  $a_\tau = 2.5 \times 10^{-2}$  (Figure 8 (b)), and roughly twice as  
 593 much with a twice higher time step, i.e.  $a_\tau = 5 \times 10^{-2}$  (Figure 8 (c)). AFLIP has



(a) PIC, FLIP, and FLIP0.99 and  $a_\tau = 5 \times 10^{-2}$  (same results for  $a_\tau = 2.5 \times 10^{-2}$ )

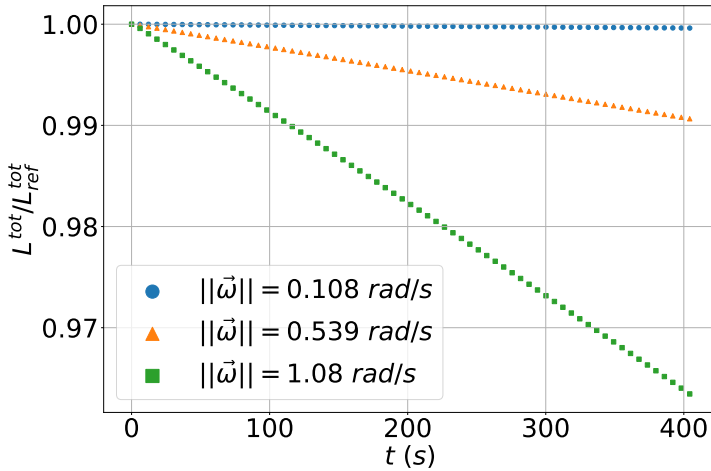


(b) Affine- and Taylor-based cases and  $a_\tau = 2.5 \times 10^{-2}$



(c) Affine- and Taylor-based cases and  $a_\tau = 5 \times 10^{-2}$

**Fig. 8:** (Non-)Conservation of angular momentum for the rotating cube, with various motion integration strategies and time steps ( $L_{ref}^{tot} \approx 7,413 \text{ kg.m}^2/\text{s}$ ).



**Fig. 9:** Impact of  $||\vec{\omega}||$  onto the (non-)conservation of the angular momentum for the rotating cube up to  $t \approx 400$  s for APIC, with  $a_\tau = 5 \times 10^{-2}$  ( $7,413$   $kg.m^2/s \leq L_{ref}^{tot} \leq 74,128$   $kg.m^2/s$ ).

594 a similar behaviour, with a steeper and more noisy decrease in  $L^{tot}$ , as well as  
 595 a higher sensitivity to  $\Delta t$ . Also, one can see that TFLIP gives almost the same  
 596 results as AFLIP, while TPIC differs from APIC by both its non-monotonous  
 597 evolution and the amount of  $L^{tot}$  it dissipates, which is approximately 0.3%  
 598 more than APIC at  $t \approx 400$  s. Various simulations conducted in [21] also  
 599 showed that the TPIC conservation of some angular momentum was less than  
 600 the APIC one.

601 For what concerns the influence of the angular velocity magnitude on the  
 602 APIC results, Figure 9 shows that the faster the cube rotates, the faster its  
 603 angular momentum is lost. As a matter of fact, multiplying the velocity by 5  
 604 makes the angular momentum dissipates approximately 24 times faster, and  
 605 multiplying it by 10 makes it dissipate approximately 95 times faster, suggest-  
 606 ing that the dissipation ratio evolves as the square of the angular velocity ratio.  
 607 Keeping in mind the results given in Figure 8, one could accommodate this  
 608 loss on  $L^{tot}$  by dividing the time step by the squared angular velocity ratio.

609 Although all affine augmented or Taylor-based motion integration strate-  
610 gies are able to accurately conserve the angular momentum, APIC stands out  
611 as the most predictable and performant in that regard. However, its incapacity  
612 to preserve the energy through an impact, as shown in Section 4.1, makes  
613 AFLIP or TFLIP emerges as the possibly optimal choice for integrating motion  
614 equations. The following section pursues the analysis on a more realistic MPM  
615 simulation setting.

## 616 5 Granular column collapse case study

617 The influence of the MPM motion integration strategy is now examined on  
618 a more realistic case in the form of a granular column collapse, which is a  
619 classical case study for granular materials being seen as solids prone to large  
620 displacements and often simulated with MPM [19, 32, 44, 50], among other  
621 numerical approaches [51, 52, 53, 54].

### 622 5.1 Simulation setup

#### 623 5.1.1 General description

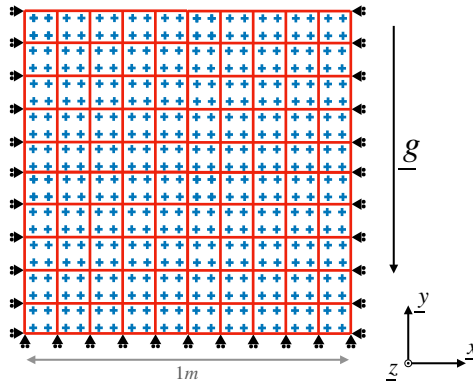
624 For simplicity, the problem is considered to be invariant in the out-of-plane  
625 direction, enabling a plane-strain numerical analysis with a 2D mesh geometry,  
626 likewise to [32, 44, 50, 51, 55, 56]. While a third principal stress is naturally  
627 still computed along the out-of-plane direction, the material point volume is  
628 computed in Eq. (9) as a surface and all energies will be given in  $J/m$ , i.e.,  
629 normalized with respect to the out-of-plane length.

630 The simulation includes two steps: first, the material is let to settle verti-  
631 cally under gravity in a lateral displacement-constrained column; second, the  
632 material is triggered to collapse under its own weight after releasing the pre-  
633 vious lateral constraint. While the first step basically consists in computing

634 a lithostatic stress state in the column which could also be directly assigned,  
 635 its simulation will enable further discussion of the MPM results. Gravitational  
 636 acceleration is taken to be of magnitude  $g = 9.81 \text{ m/s}^2$ .

### 637 5.1.2 Geometry, mesh and material points

638 The width of the column is considered to span over the  $x$ -axis while its height  
 639 spans over the  $y$ -axis. The aspect ratio  $AR$  is defined as the column initial  
 640 height  $L_y^{init}$  divided by its initial width  $L_x^{init} = 1 \text{ m}$ , the latter being the same  
 641 for all simulations (Figure 10).



**Fig. 10:** Initial conditions for the granular column collapse simulation, illustrated for the specific case of  $AR = 1$ ,  $N_{mppc} = 4$  and  $l_{cell} = 10 \text{ cm}$  (see text)

642 The mesh consists of square elements with a side of  $l_{cell} \in \{10 \text{ cm}, 7.69 \text{ cm},$   
 643  $5.88 \text{ cm}, 5 \text{ cm}\}$ , i.e.  $L_x^{init}/l_{cell} \in \{10, 13, 17, 20\}$ . For the initial settling phase,  
 644 the mesh cells just cover material extents:

$$W_{mesh}^{settling} = L_x^{init} \quad (57)$$

$$H_{mesh} = L_y^{init} = AR \times W_{mesh}^{settling} \quad (58)$$

645 with  $H_{mesh}$  the mesh height which will be constant for the whole simulation  
 646 while  $W_{mesh}^{settling}$  is the mesh width only during this settling phase. The second  
 647 simulation phase will actually adopt a wider mesh for enabling a collapse  
 648 on the right side. Taking advantage of the absence of mesh-history in MPM,  
 649 the mechanical state of material points obtained after the settling phase is  
 650 imported on the left-side of another mesh defined as a substantial enlargement  
 651 of the settling one:

$$W_{mesh}^{collapse} = 6 \times W_{mesh}^{settling} \quad (59)$$

652 All cells contain initially  $N_{mppc}$  (a squared integer) material points. For  
 653 the purpose of verifying quadrature rules, at least in the initial setup, those  
 654 are located at the roots of Legendre's polynomials (in local coordinates), given  
 655 in e.g. [57]. As an example, the roots of the second Legendre's polynomial  
 656 correspond to the classical local positions of the points in each direction for  
 657  $N_{mppc} = 2^2 = 4$ :

$$\tilde{s} = \pm \frac{1}{\sqrt{3}}, \quad \forall s \in \{x, y\} \quad (60)$$

658 In terms of boundary conditions, a nil orthogonal velocity is imposed at  
 659 the left, right and bottom boundary nodes. For convenience, the left and right  
 660 boundaries are referred to as walls, while the bottom boundary is referred to  
 661 as a floor. A Coulomb friction condition is imposed on the floor, driven by a  
 662  $\mu = 0.3$  coefficient.

663 Figure 10 illustrates the initial configuration of the simulation for  $AR = 1$ ,  
 664  $N_{mppc} = 4$  and  $l_{cell} = 10 \text{ cm}$ .

Mass density $\rho$	Young's modulus $E$	Poisson's ratio $\nu$	Friction angle $\phi$	Dilatancy angle $\psi$	Cohesion $c$	Tension cut-off $T_{cut}$
$1,748 \text{ kg.m}^{-3}$	$1.284 \times 10^7 \text{ Pa}$	0.16	$33.75^\circ$	$0^\circ$	$1 \text{ Pa}$	$\frac{C}{\tan\phi}$

**Table 2:** Parameters used with the Mohr-Coulomb model

### 665 5.1.3 Material parameters and numerical damping

666 In order to have a simple access to all energy quantities including material dissi-  
667 pation (see below), the collapse is simulated adopting the simplest constitutive  
668 model for frictional materials, i.e. the elastic-plastic Mohr-Coulomb model that  
669 combines Hooke's law for the elastic regime and Mohr-Coulomb perfect plas-  
670 ticity with a non-associated flow rule. Corresponding material parameters are  
671 calibrated (in the best possible way for this simple model) from the triaxial  
672 behavior of a real sandy soil, Camargue's sand studied in [58], and are given  
673 in Table 2.

674 As for the initial settling phase which can be seen as purely numerical,  
675 an elastic behaviour is chosen in order to prevent plastic deformations that  
676 would otherwise occur during the P-wave-like oscillations of the model from  
677 zero initial stresses (1 Pa, actually, for all material points in order to avoid  
678 edge-cases in the Mohr-Coulomb model) towards lithostatic equilibrium. When  
679 explicitly mentioned for some of those settling simulations whose dynamics is  
680 not of interest, a fictitious Cundall's damping force [59] is also introduced in  
681 Eq. (5) in order to dissipate energy in a user-controlled way and converge to  
682 the intended lithostatic equilibrium. Such damping force is implemented as  
683 follows, being computed from the total nodal force supplemented by a damping  
684 parameter  $D \geq 0$  (used with  $D = 0.1$  when mentioned to be present) and  
685 oriented to be component-wise dissipative:

$$\vec{f}_{damp} = -D \|\vec{f}^i\|_{cws} \overrightarrow{(\vec{V}^i)} \quad (61)$$

$$\text{where } \text{cws}_k(\vec{u}) = \frac{u_k}{|u_k|} \quad \text{for any vector } \vec{u} = (u_k), \quad k \in \llbracket 1, 3 \rrbracket \quad (62)$$

#### 5.1.4 Energy balance and other post-processing quantities

In the present usage of the Mohr-Coulomb constitutive model, the elastic energy  $E_{el}$  and the energy dissipated during plastic deformation, i.e. the plastic work  $W_{pl}$ , are computed incrementally as follows (with  $\delta_{ij}$  the Kronecker's symbol):

$$d\epsilon_{ij}^{el} = \frac{1}{E} ((1 + \nu)d\sigma_{ij} - \nu d\sigma_{kk}\delta_{ij}) \quad (63)$$

$$dE_{el} = \sum_p \sigma_{ij} d\epsilon_{ij}^{el} v^p \quad (64)$$

$$dW_{pl} = - \sum_p \sigma_{ij} (d\epsilon_{ij} - d\epsilon_{ij}^{el}) v^p \quad (65)$$

with  $E_{el}(0) = W_{pl}(0) = 0$   $J/m$  in line with chosen initial conditions and  $dW_{pl} < 0$  during plastic deformation by convention. For the numerical evaluation of Eq. (64),  $\sigma_{ij}$  is replaced by the average of its two values obtained before and after the constitutive update (i.e., at the very beginning and at the very end of a MPM iteration), which enables one to avoid finite size (of  $d\epsilon_{ij}^{el}$ ) effects and keep an exact numerical integration by virtue of the linear relationship between  $\sigma_{ij}$  and  $\epsilon_{ij}^{el}$ .

The energy dissipated by frictional forces  $W_{frict}$  during the collapse is also computed incrementally from:

$$dW_{frict} = \sum_i \vec{F}_{frict}^i \cdot \vec{V}^i \Delta t \quad (66)$$

with  $\vec{F}_{frict}^i$  the frictional force at grid point  $i$  and  $dW_{frict} < 0$  during sliding by construction.

702 In the general absence of Cundall's damping, the energy balance of the  
703 system is then:

$$dE_k + dE_g + dE_{el} = dW_{pl} + dW_{frict} + dW_{MPM} \quad (67)$$

704 where  $dW_{MPM}$  (and its integral  $W_{MPM}$ ) will quantify the numerical MPM  
705 energy error (dissipation, if negative) related with the chosen motion integra-  
706 tion strategy, after being computed as:

$$dW_{MPM} = dE_k + dE_g + dE_{el} - (dW_{pl} + dW_{frict}) \quad (68)$$

707 As for the kinetic and gravitational energies, they are computed from  
708 Eq. (47) and Eq. (48) respectively. The difference in potential energy between  
709 two stable states (before and after the collapse)  $|\Delta E_p^{stable}|$  will often be used  
710 as a reference value to normalize energy balance and is computed as follows:

$$|\Delta E_p^{stable}| = |\Delta E_g + \Delta E_{el}| \quad (69)$$

711 In addition to the above energy consideration, the time evolution of the  
712 front of the column is measured as a major insight on how the collapse unfolds,  
713 with an upper bound equal to the mesh right boundary  $x = W_{mesh}^{collapse}$ . Because  
714 the left edge does not move during the collapse, the latter front position is  
715 simply equal to the width of the column,  $L_x$ , and is tracked from the furthest  
716 material point along the  $x$ -axis and the mesh geometry (naturally considering  
717 the material to be present in a whole cell as long as at least one material point  
718 is inside):

$$L_x = \text{floor} \left( \frac{\max_{p \in \{p\}}(x^p)}{l_{cell}} + 1 \right) l_{cell} \quad (70)$$

719 Because of this "voxelized" point of view,  $L_x$  logically increases by steps  
 720 of  $l_{cell}$  as the column spreads to the right. Note that  $L_x$  is directly related to  
 721 the so-called runout distance  $d_r$ :

$$d_r = L_x - L_x^{init} \quad (71)$$

722 Additionally, the normalized spreading length  $\tilde{L}$  and collapse time  $\tilde{t}$  are  
 723 other common dimensionless quantities used to describe the dynamics of the  
 724 collapse, both being computed as in [60]:

$$\tilde{L} = \frac{d_r}{L_x^{init}} \quad (72)$$

$$\tilde{t} = \frac{t}{\sqrt{\frac{AR \times L_x^{init}}{g}}} \quad (73)$$

725 Finally, in order to quantify how much the column is sheared, the second  
 726 invariant  $\epsilon_D$  of material point strain tensor  $\epsilon^p$  is also systematically monitored,  
 727 using its classical expression:

$$\epsilon_{dev}^p = \epsilon^p - \frac{\text{tr}(\epsilon^p)}{3} \mathbf{I}_3 \quad (74)$$

$$\epsilon_D = \|\epsilon_{dev}^p\| = \sqrt{(\epsilon_{dev})_{ij}(\epsilon_{dev})_{ij}} \quad (75)$$

$l_{cell}$	$N_{mppc}$	$\Delta t$	$a_\tau$	Motion integration strategy	AR
10 cm	4	$2.92 \times 10^{-6}$ s	$2.25 \times 10^{-3}$	Each of the following FLIP, PIC, APIC, FLIP0.9, AFLIP, TFLIP	1
		$1.46 \times 10^{-6}$ s	$1.25 \times 10^{-3}$		
		$3.65 \times 10^{-7}$ s	$3.125 \times 10^{-4}$		
		$1.46 \times 10^{-7}$ s	$1.25 \times 10^{-4}$		
		$7.29 \times 10^{-8}$ s	$6.25 \times 10^{-5}$		

**Table 3:** Parameters used to investigate the time step influence for different motion integration strategies (series S1, 30 simulations)

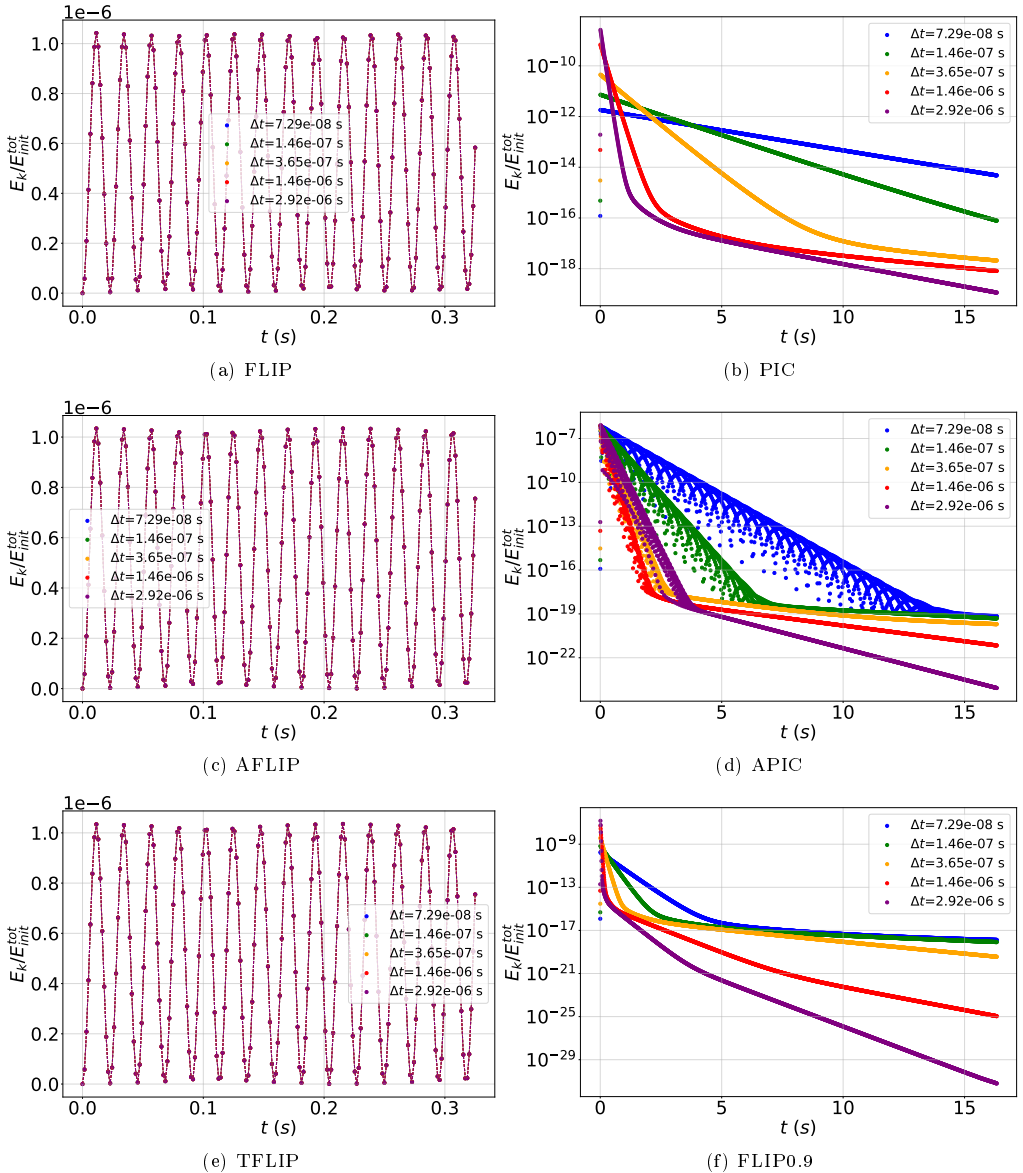
## 728 5.2 Time step influence for PIC damping

729 As a first interesting result, the chosen value for the time step appears to possi-  
730 sibly have a strong influence on the settling dynamics, looking at the so-called  
731 S1 series of simulations that combine different motion integration strategies  
732 with a variable time step (Table 3).

733 Observing kinetic energy during this settling phase (Figure 11), one can see  
734 that for all FLIP-based strategies (Figure 11 (a), (c) and (e))  $\Delta t$  has virtually  
735 no impact on the simulated dynamics: all tested values lead the column to  
736 oscillate indefinitely with the same period, in line with the conservative nature  
737 of FLIP and the elastic nature of the settling process.

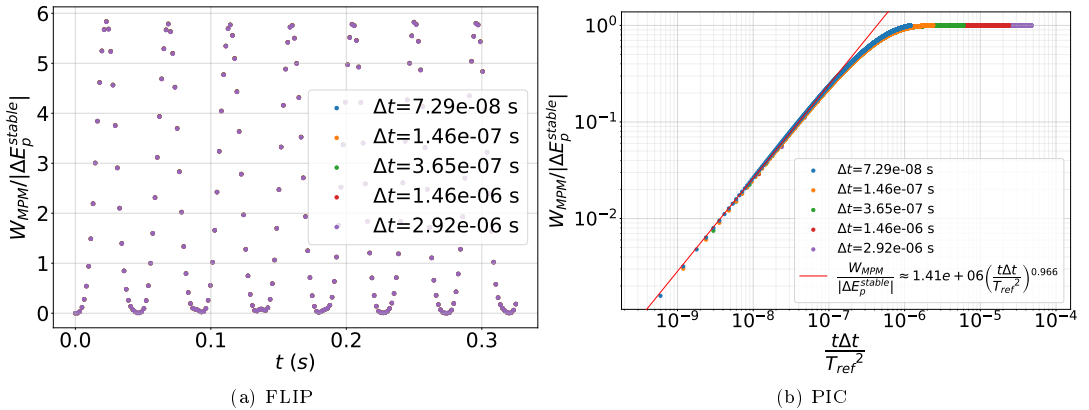
738 However, a drastic influence of  $\Delta t$  appears with the PIC-based strategies  
739 (for the same  $\Delta t$  values being below the critical one, Figure 11 (b), (d) and (f))  
740 where the dissipation rate of  $E_k$  is lower for smaller  $\Delta t$ . From this point of view,  
741 one can interpret the PIC damping as being even more artificial and numeric  
742 in nature than Cundall's damping of Eq. (61) since it is purely cumulative  
743 according to MPM iterations instead of being time-proportional to a given  
744 dissipative power as is the case for Cundall's damping which is introduced in  
745 the form of an ad-hoc additional force.

746 A more striking evidence of this very artificial nature of the PIC damping  
747 is presented in Figure 12 (b), where all values of  $W_{MPM}$  are nicely grouped  
748 together when plotted with respect to  $t\Delta t$ , demonstrating a direct dependence



**Fig. 11:** Kinetic energy during the settling phase for different  $\Delta t$  and motion integration strategy (simulation series S1)

749 on  $\Delta t$ . As a contrast, the  $W_{MPM}$  observed for FLIP on Figure 12 (a) has no  
 750 noticeable correlation to  $\Delta t$ .

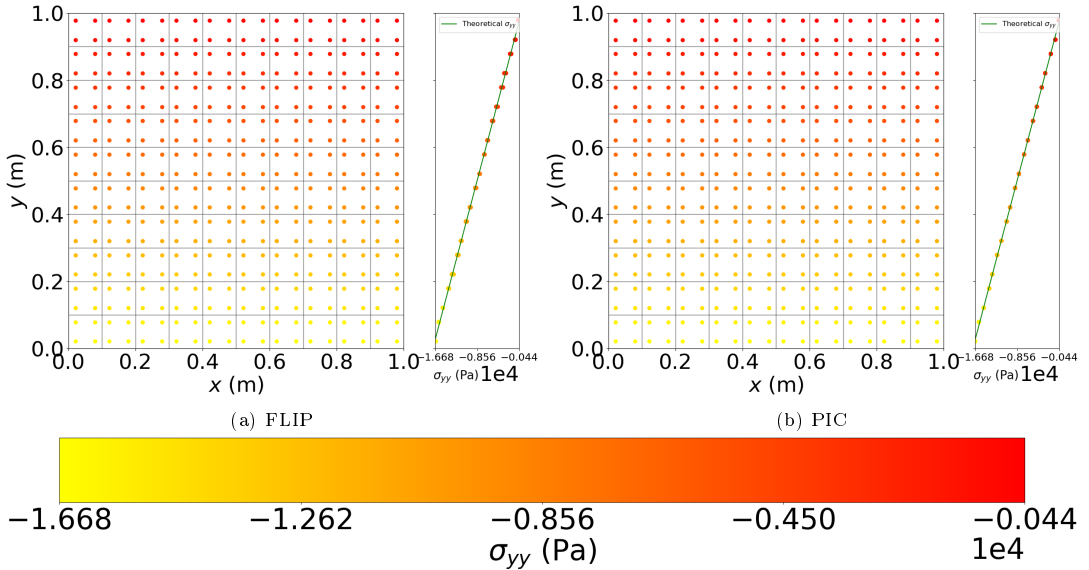


**Fig. 12:** Energy difference caused by the MPM procedure  $W_{MPM}$  during the settling (simulation series S1,  $|\Delta E_p^{stable}| \approx 2.45 \text{ J/m}$ )

751 Because FLIP-based strategies induce permanent oscillations of the col-  
 752 umn, the settling step was performed again for these simulations using  
 753 Cundall's damping, see appendix C. Figure 13 (a) and (b) show that the  
 754 final vertical stress field obtained with both FLIP-based (with  $D = 0.1$ ) and  
 755 PIC-based (with  $D = 0$ ) strategies is the expected lithostatic one.

756 These first observations on the settling phase suggest that FLIP-based  
 757 strategies are to be preferred from both a theoretical and practical point of  
 758 view. Indeed, since PIC significantly decreases the kinetic energy, the column  
 759 takes approximately 50 times longer (with the finest time discretization) to  
 760 settle, requiring much more computational resources.

761 Looking then at the collapse phase where displacements are much more  
 762 significant, the time evolution of the column width  $L_x$  is plotted on Figure 14  
 763 for the same simulation series S1 (Table 3). Here again, one can see that  $\Delta t$   
 764 does not have a significant influence on the results with FLIP-based strategies.  
 765 Indeed, both the collapse dynamics and the final  $L_x$  obtained are in these  
 766 cases virtually identical for different  $\Delta t$ . However, the final column obtained  
 767 with AFLIP and TFLIP is approximately 9% lower than the one obtained

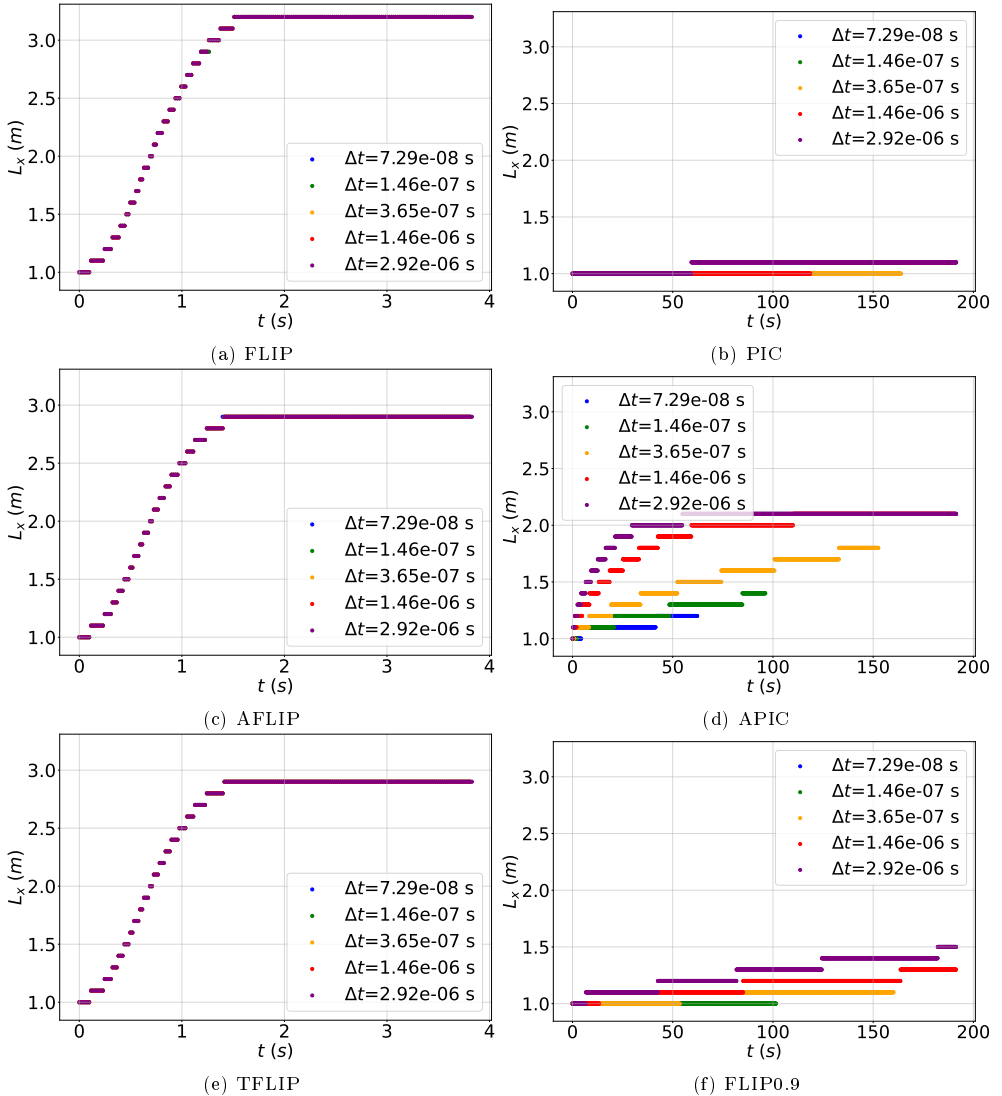


**Fig. 13:** Vertical stress at the end of the settling phase with  $\Delta t = 1.46 \times 10^{-6}$  s (simulation series S1)

768 with FLIP. Regarding the kinetic energy  $E_k$  observed on Figure 15 (a), (c),  
 769 and (e), it is also independent from  $\Delta t$  for all FLIP-based collapses, but one  
 770 can notice that at the end a slight amount of kinetic energy (less than 0.001%  
 771 of the initial potential energy) remains, although the material points do not  
 772 appear to be moving. The use of Cundall's damping could solve this issue, by  
 773 decreasing globally the energy level.

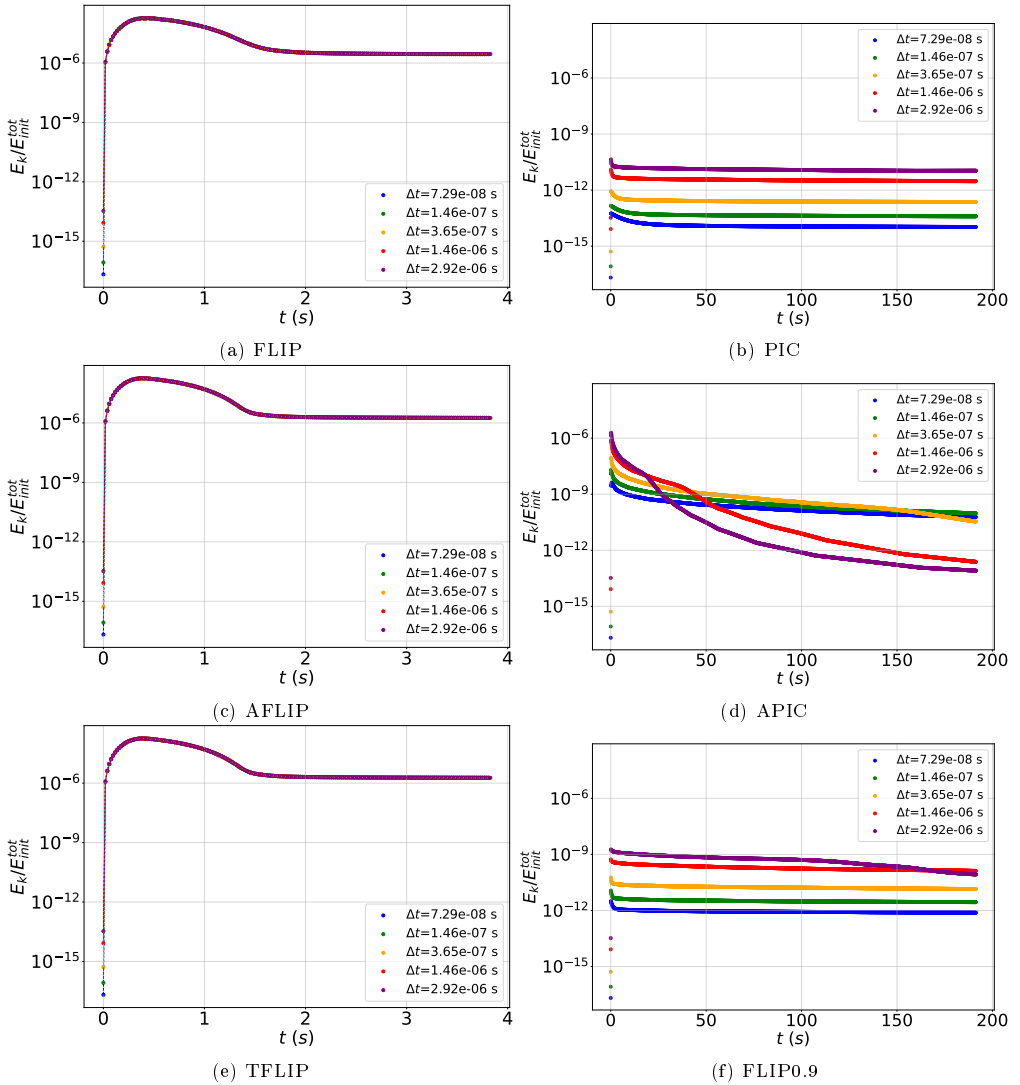
774 As for PIC-based strategies, Figures 14 (b), (d) and (f) clearly show that  
 775 they induce the column to collapse at an unrealistically low rate, as it can also  
 776 be observed on Figure 16 (d), where the collapse is still in an early stage 3  
 777 minutes after releasing the right constraint on the column. As a comparison,  
 778 it takes approximately a second and a half with FLIP-based strategies for the  
 779 1 m-high column to completely collapse, which is much more realistic.

780 Similarly to the settling phase, the PIC damping observed during the col-  
 781 lapse is dependent on the value of  $\Delta t$  used, with higher time steps leading to



**Fig. 14:** Width of the column during the collapse for different  $\Delta t$  and motion integration strategies (simulation series S1)

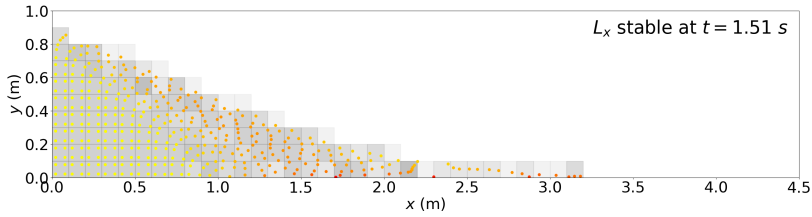
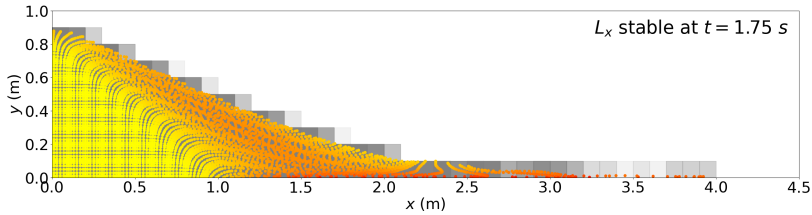
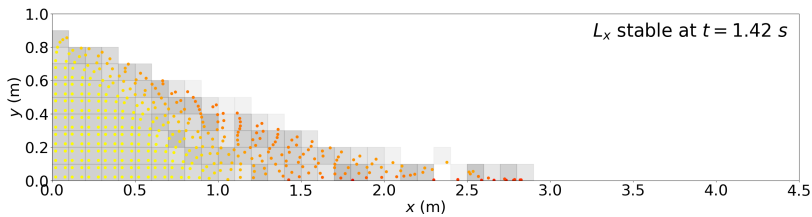
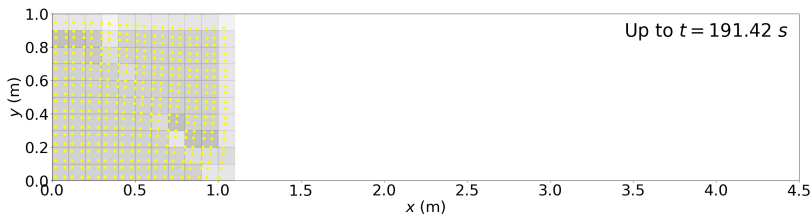
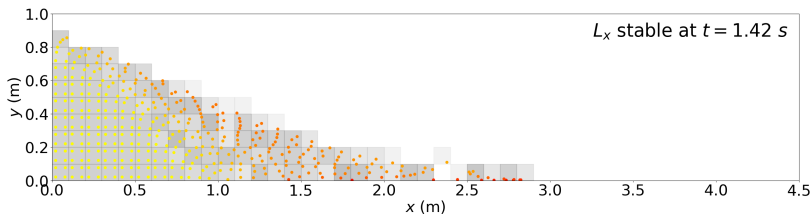
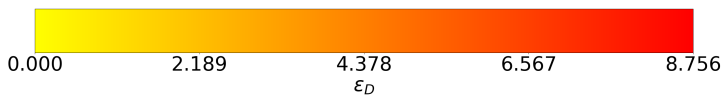
782 faster collapses. The developed kinetic energy  $E_k$  in the PIC simulation is at  
 783 least 4 million times lower than with FLIP, see Figure 15 (a) and (b), and the  
 784 gap increases with low values of  $\Delta t$ , which require more MPM iterations to  
 785 cover the same model time. Yet again, using APIC and the PIC-FLIP blend



**Fig. 15:** Kinetic energy during the collapse for different  $\Delta t$  and motion integration strategies (series S1,  $E_{init}^{tot} \approx 8.57 \cdot 10^3 \text{ J/m}$ )

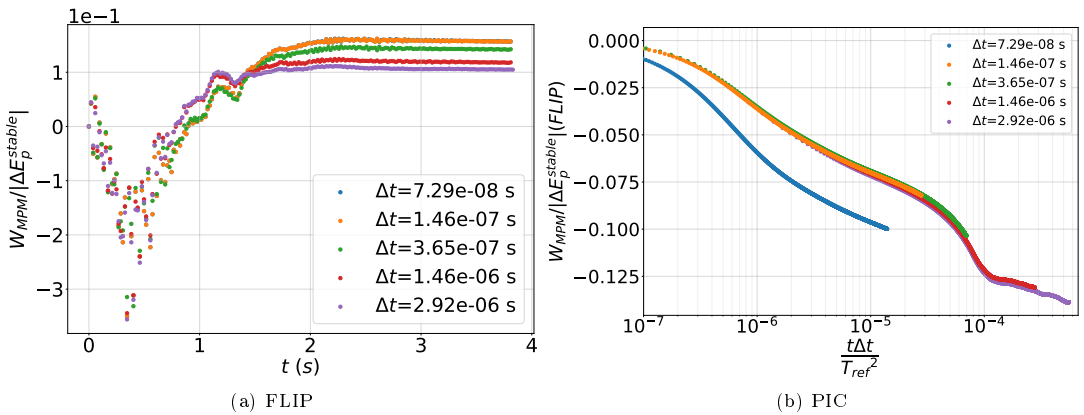
786 FLIP0.9 (with a non-negligible 0.1 portion of PIC) enables one to mitigate the  
 787 PIC damping and describe slightly faster column collapses, but it is certainly  
 788 not enough for the results to be realistic.

789 Figure 17 (a) shows that for  $\Delta t \leq 1.46 \times 10^{-7} \text{ s}$ , FLIP dissipates the  
 790 same small amount of energy (approximately 0.1% of the expected energy

(a) With FLIP,  $N_{mppc} = 4$ ,  $l_{cell} = 10$  cm, and  $\Delta t = 7.29 \times 10^{-7}$  s (from series S1)(b) With FLIP,  $N_{mppc} = 49$ ,  $l_{cell} = 10$  cm, and  $\Delta t = 7.29 \times 10^{-6}$  s (from series S3)(c) With AFLIP,  $N_{mppc} = 4$ ,  $l_{cell} = 10$  cm, and  $\Delta t = 7.29 \times 10^{-7}$  s (from series S1)(d) With PIC,  $N_{mppc} = 4$ ,  $l_{cell} = 10$  cm, and  $\Delta t = 2.92 \times 10^{-6}$  s (from series S1)(e) With TFLIP,  $N_{mppc} = 4$ ,  $l_{cell} = 10$  cm, and  $\Delta t = 1.46 \times 10^{-7}$  s (from series S1)

**Fig. 16:** Positions of material points after the collapse (except for PIC), with deviatoric strains  $\epsilon_D$  in colorbar (see corresponding video in the supplementary material)

791 difference) at the end of the collapse. For higher values of  $\Delta t$ , the column gains  
 792 energy as  $\Delta t$  increases. In Figure 17 (b), an overlap of the data similar to one  
 793 obtained during the settling is found when plotting the energy lost by MPM as  
 794 a function of  $\Delta t$ , although the result for  $\Delta t = 7.29 \times 10^{-8}$  somewhat deviates  
 795 from the other results. Note that in this figure, the  $W_{MPM}$  obtained with PIC  
 796 is normalized by the  $\Delta E_p^{stable}$  obtained with FLIP because PIC columns did  
 797 not have the time to reach a stable state.



**Fig. 17:** Energy difference caused by the MPM procedure  $W_{MPM}$  during the collapse (series S1,  $|\Delta E_p^{stable}| \approx 4 \times 10^3$  J/m)

### 798 5.3 Spatial discretization influence

799 While the previous results were obtained using  $N_{mppc} = 4$  material points  
 800 per mesh element, likewise to [32], the possibility for the MPM results to con-  
 801 verge with respect to the spatial discretization is often an open question, be  
 802 it in terms of  $N_{mppc}$  or the size of a mesh element  $l_{cell}$ . Considering various  
 803 simulations and/or various quantities for similar granular column collapse sim-  
 804 ulations such as shown here, convergence was for instance usually obtained  
 805 for what regards  $l_{cell}$  in [50] but that was less the case in [29, 32]. Here, two

$l_{cell}$	$N_{mppc}$	$\Delta t$	$a_\tau$	Motion integration strategy	AR
10 cm	4	$1.46 \times 10^{-6} s$	$1.25 \times 10^{-3}$	FLIP	1
7.69 cm		$1.12 \times 10^{-6} s$			
5.88 cm		$8.58 \times 10^{-7} s$			
5 cm		$7.29 \times 10^{-7} s$			

**Table 4:** Parameters used to investigate the influence of the cell size,  $l_{cell}$  (series S2, 4 simulations)

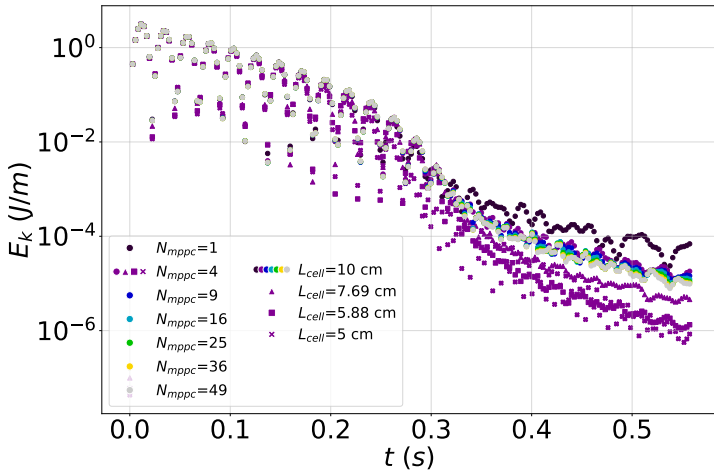
$l_{cell}$	$N_{mppc}$	$\Delta t$	$a_\tau$	Motion integration strategy	AR
10 cm	1	$1.46 \times 10^{-6} s$	$1.25 \times 10^{-3}$	FLIP	1
	4				
	9				
	16				
	25				
	36				
	49				

**Table 5:** Parameters used to investigate the influence of the number of particles per cell,  $N_{mppc}$  (series S3, 7 simulations)

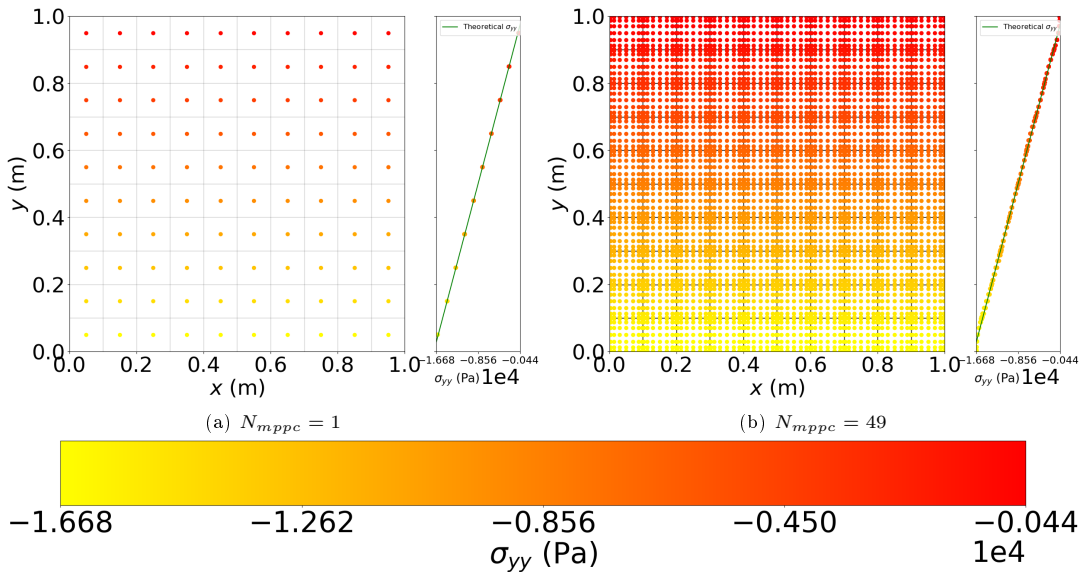
806 other simulation series, S2 and S3, investigate this aspect in terms of both  $l_{cell}$   
 807 (series S2, Table 4) and  $N_{mppc}$  (S3 series, Table 5).

808 Regarding first the simulation of the settlement process, Figure 18 shows  
 809 that neither  $l_{cell}$  nor  $N_{mppc}$  has a noticeable influence for the most part of the  
 810 stabilization, for  $t < 0.3 s$ . After that, lower values of  $l_{cell}$  lead to lower  $E_k$ ,  
 811 but the value of  $N_{mppc}$  has still almost no impact on  $E_k$ . In fact, a difference  
 812 can be observed between  $N_{mppc} = 1$  and  $N_{mppc} \geq 4$ . Indeed, when using only  
 813 1 material point per mesh element,  $E_k$  is higher than for all other values of  
 814  $N_{mppc}$ , but since its value is already less than 0.1% of the maximum  $E_k$ , this  
 815 variation can be considered negligible. Moreover, Figure 19 (a) confirms that  
 816 even with  $N_{mppc} = 1$ , the vertical stress obtained after stabilization is the one  
 817 expected.

818 As for the collapse phase, Figure 20 shows  $L_x$  and  $E_k$  for all tested values  
 819 of  $l_{cell}$  and  $N_{mppc}$ , and Figure 16 (b) (or the corresponding video) shows the

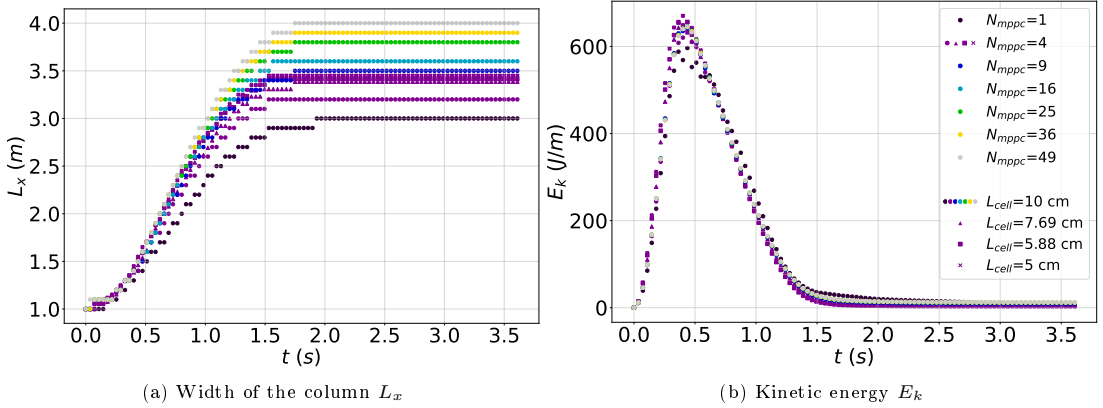


**Fig. 18:**  $E_k$  during the settling phase for different spatial discretization parameters (simulation series S2 and S3 with FLIP)



**Fig. 19:** Vertical stress at the end of the settling phase in various configurations and with  $\Delta t = 1.46 \times 10^{-6}$  s (simulation series S3)

820 positions of all material points throughout the collapse for  $N_{mppc} = 49$ . Lower  
 821 values of  $l_{cell}$  increase  $L_x$  by approximately 7% (see Figure 20 (a)), with only



**Fig. 20:** Influence of the spatial discretization parameters during the collapse phase (series S2 and S3)

822 a slight impact for  $l_{cell} < 10$  cm, the mesh element size has thus no significant  
 823 impact on the collapsing column. A stronger influence is here observed for  
 824 what concerns  $N_{mppc}$  since the final  $L_x$  is 25% higher for  $N_{mppc} = 49$  than  
 825 for  $N_{mppc} = 1$ . Although the influence of  $N_{mppc}$  is less important for  $25 \leq$   
 826  $N_{mppc} \leq 49$ , no clear convergence is observed for  $L_x$ .

827 This dependency to  $N_{mppc}$  is probably caused by the "voxelized" point of  
 828 view inherent to the MPM, along with the extensive shear experienced by the  
 829 material points on the right side of the column (see Figure 16 (a) and (b)).  
 830 Indeed, if one was to attribute an initial domain to each material point, likewise  
 831 to the use of GIMP in [50], at some point the ones on the right side of the  
 832 column should span over several mesh elements, but this can not be taken into  
 833 account in the present MPM formulation. Increasing  $N_{mppc}$  decreases the size  
 834 of these domains, improving their chances to restrict in only 1 mesh element.  
 835 This observation suggests that a splitting procedure similar to the one used in  
 836 FEMLIP [31] could reduce the dependency on  $N_{mppc}$ . Basically, the size of the  
 837 domains previously mentioned would be tracked, and a material point be split  
 838 into two if its domain became too large. Nevertheless, this procedure is left

$l_{cell}$	$N_{mppc}$	$\Delta t$	$a_\tau$	Motion integration strategy	$AR$
10 cm	4	$1.46 \times 10^{-6}$ s	$1.25 \times 10^{-3}$	FLIP and AFLIP	0.6
				FLIP	1
				FLIP and AFLIP	2.4
				FLIP and AFLIP	3

**Table 6:** Parameters used for comparison with results from the literature (series S4, 4 simulations)

839 for future work as it would require particular attention on the velocities and  
 840 state parameters to attribute to each new material point in order to conserve  
 841 important quantities (e.g. the momentum) in the process.

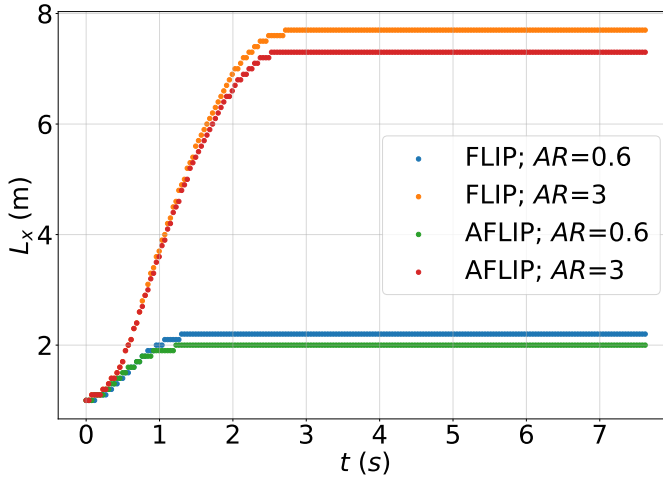
## 842 5.4 (Non-)necessity of an affine-augmented motion 843 integration strategy for different aspect ratio and 844 comparison with the literature

845 A last simulation series (S4, see Table 6) combines a variable aspect ratio  $AR$   
 846 and two different motion integration strategies among the most conservative  
 847 choices FLIP and AFLIP.

848 Looking at the collapse dynamics in terms of spreading length, i.e.  $L_x(t)$   
 849 (Figure 21), one can see that both FLIP and AFLIP provide similar results for  
 850  $AR = 0.6$  and  $AR = 3$ . More precisely, FLIP gives a wider collapsed column  
 851 by approximately 9% for  $AR = 0.6$ , and 5% for  $AR = 3$ . Considering the  
 852 conclusions from Section 4, these results suggest that the collapse of a granular  
 853 column up to  $AR = 3$  doesn't involve much rotational motion and can thus be  
 854 modelled using the simple FLIP strategy.

855 For further validation of the results, the spreading lengths obtained with  
 856 the FLIP scheme in this same S4 series are finally compared to results from  
 857 the literature in Figure 22. The latter include:

- 858 • experimental results conducted on glass beads in [61];



**Fig. 21:** Evolving width of the column during the collapse for different aspect ratios and motion integration strategies (simulation series S4)

- 859 • numerical results performed in [51] with a FEM-based hybrid Eulerian-
- 860 Lagrangian method, in conjunction with the Mohr-Coulomb model with
- 861  $\phi \in [25^\circ; 40^\circ]$ .
- 862 • numerical results obtained in [52] using a 3D DEM model, with a viscous
- 863 elasto-plastic contact law;
- 864 • numerical results from [53], obtained with the SPH method and the
- 865 micromechanical 3D-H model (denoted in this paper SPHx3D-H).

866 While no attempt was made in the present study to define material param-

867 eters similar to those of these literature results, one should note that the latter

868 form a consistent data set which seems to be independent of material proper-

869 ties and should therefore be appropriate to serve as a comparison basis for our

870 results.

871 A first observation is that our MPM columns take slightly longer to reach

872 their final length, for all values of  $AR$ . Considering that PIC-based motion

873 integration strategies lead to even longer spreading times, as previously shown

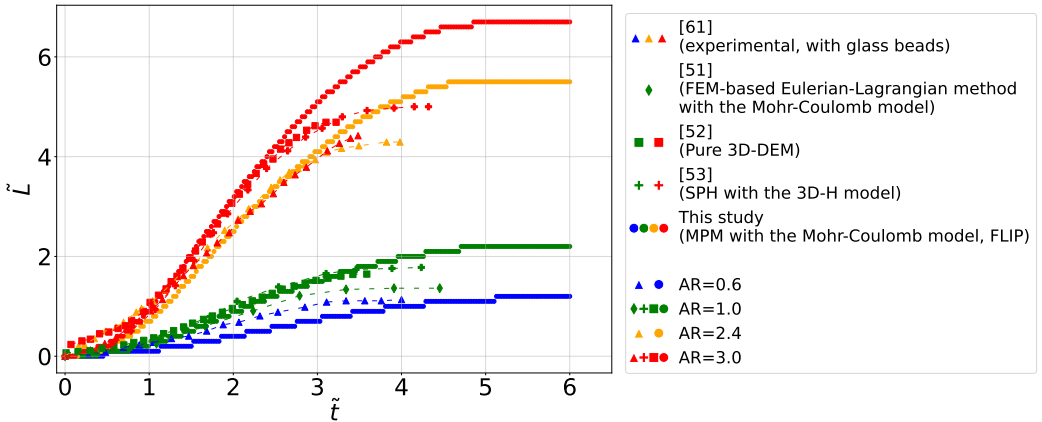
874 in Figure 14, the energy dissipation observed with FLIP in Section 4.2 might  
875 be to blame.

876 Figure 22 also shows that the final spreading length is higher in MPM, with  
877 respect to collapses performed with other numerical models for the same  $AR$ .  
878 For instance, with  $AR = 1$ , the MPM simulation gives a final length approx-  
879 imately 38% higher than the results presented in [51], while the results from  
880 [53] are only approximately 19% lower. A similar gap is observed between our  
881 results and the DEM ones from [52]. Because both SPHx3D-H and DEM are  
882 supposed to be more accurate for granular materials than the Mohr-Coulomb  
883 model, the proximity of our results with the SPHx3D-H and DEM results is  
884 quite comforting.

885 According to the experimental results from [61], the real collapsing columns  
886 are clearly less wide than the MPM ones, more precisely by 34% for  $AR = 3$ .  
887 However, for  $AR = 0.6$ , this difference is restricted to approximately 5%,  
888 and the experimental  $\tilde{L}$  is higher than the MPM one for the most part of  
889 the collapse. This difference could come from the way the measurement of  
890  $\tilde{L}$  is performed: experimentally, some grains that are isolated from the rest  
891 of the column might be excluded from the measurement, but with numerical  
892 results, all of the material present initially in the column is included in the  
893 measurement.

## 894 6 Conclusion

895 This paper reviewed the implications of using different motion integration  
896 strategies in MPM, whose possibility stems from the necessity to express and  
897 integrate in time kinematic fields on a double layer of spatial discretization  
898 with uncoupled grid nodes and material points. Numerical investigations, in  
899 a USF scheme by default, focused on the impact of these motion integration



**Fig. 22:** Obtained collapse dynamics in terms of normalized spreading length  $\tilde{L}$  (series S4), in comparison with results from the literature

900 strategies for conserving energy in a number of cases of increasing complexity,  
 901 up to the simulation of a granular column collapse.

902 Even though theoretically correct for describing the simplest rigid linear  
 903 motions, PIC was recalled to dissipate the energy during an elastic deformation  
 904 event in a MPM simulation and considerably dampen, i.e., slow down, the  
 905 description of a granular column collapse. Moreover, this unnatural damping  
 906 was shown herein to increase with lower time steps, unveiling an impossible  
 907 convergence of the PIC strategy with respect to the temporal discretization.

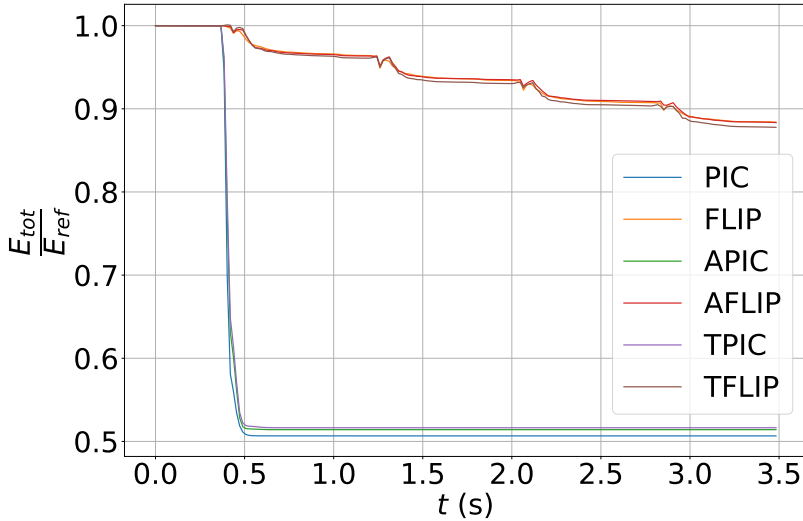
908 On the other hand, it was shown that if only a translational motion and  
 909 elastic deformations are involved in the simulation, the FLIP strategy was the  
 910 most performant at preserving the total energy in the system and unaffected  
 911 by the chosen time step, when below the divergence Courant-Friedrichs-Lewy  
 912 limit.

913 In cases involving rotational motions, it has been recalled how both PIC  
 914 and FLIP are unable to describe rigid body rotations, even though spatial  
 915 discretization affects FLIP performances in this aspect, and that an affine aug-  
 916 mentation procedure such as the APIC strategy is necessary to conserve the

angular momentum. As a matter of fact, APIC was found to conserve accurately the angular momentum in the simple case of a rigid rotating object, with the condition that the time step is low enough compared to the object's angular velocity. In that very simple case, APIC is not only more predictable than AFLIP, TPIC and TFLIP, but also more performant. However, in the general case involving deformations, APIC suffers from the unnatural damping inherited from PIC, along with its impossibility to converge with respect to the temporal discretization. The AFLIP and TFLIP strategies are thus better than APIC when rotational motions and large deformations are involved, both giving almost identical results. Ultimately, because the implementation of TFLIP is more straightforward than AFLIP, TFLIP is certainly the most suitable motion integration strategy to recommend.

## A Bouncing cube with the USL scheme

This appendix gives insights on the effect of the stress update scheme on the bouncing cube simulation presented in Section 4.1. The evolution of total energy  $E_{tot}$  during the simulation is plotted for the USL scheme in Figure 23, similarly to previous Figure 3 (a) for the USF case. One can see that all motion integration strategies are strictly dissipative when using USL, even FLIP-based ones, unlike the previous USF case. This observation is consistent with the conclusions of [38], where the USL scheme was demonstrated to be strictly dissipative by formulation. One can also note that with the present USL choice, a given motion integration strategy can not be distinguished from its affine-augmented version (e.g., PIC vs APIC or FLIP vs AFLIP) in this translation regime, in some contrast with the USF case.



**Fig. 23:** Total energy for the bouncing cube example simulated with different motion integration strategies and the USL stress update scheme ( $E_{ref} \approx 25.7$   $kJ$ ,  $a_\tau = 5 \times 10^{-2}$ )

## 941 B Reproduction of results from the literature

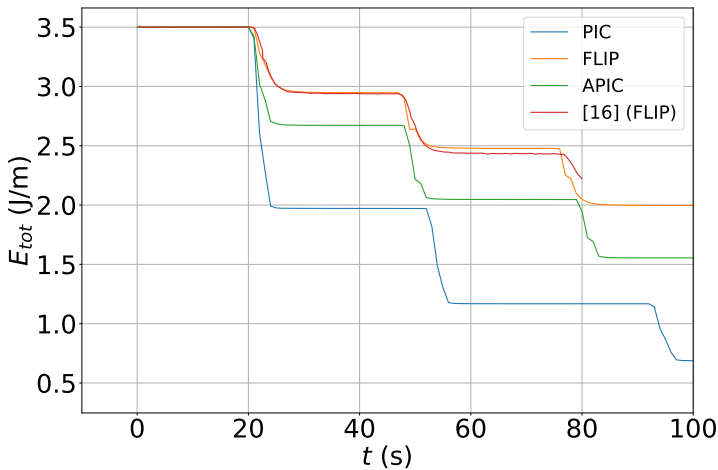
942 For validation purposes of the used MPM implementation, this section aims  
 943 to reproduce two different simulations taken from the literature: a bouncing  
 944 disk case conducted in [16], Section 4.1 therein, and a rotating disk simulation  
 945 of [18], Section 6.1 therein.

946 Although a Neo-Hookean material model was used in [16] and [18], the  
 947 results in this Appendix are obtained using the same hyperelastic Hooke's law  
 948 as used in previous Section 4.1, assigning the Neo-Hookean linearized elastic  
 949 parameters of [16] and [18] to their constant Hooke's counterparts. Also, the  
 950 simulations are performed using the USL scheme as well as boundary modified  
 951 cubic B-spline shape functions.

## 952 B.1 Bouncing elastic disk in comparison with [16]

953 The bouncing disk simulation from [16], §4.1 therein, is very similar to the  
 954 one presented in previous Section 4.1, the main differences being the number  
 955 of dimensions (2D in this Appendix, 3D in Section 4.1), the shape of the  
 956 bouncing object (a disk in this section, a cube in Section 4.1), and the absence  
 957 of gravity in the present Appendix (the movement is caused by an initial  
 958 velocity instead).

959 The simulation is performed with the same spatial and time discretiza-  
 960 tions as [16] and three different motion integration strategies (PIC, FLIP, and  
 961 APIC), although [16] only used FLIP.



**Fig. 24:** Bouncing elastic disk simulation using various motion integration strategies

962 These results show that our MPM implementation is in accordance with  
 963 the one from [16] in terms of energy conservation. Indeed, it is herein obtained,  
 964 with the FLIP strategy, virtually the same  $E_{tot}$  than [16]. The minor differ-  
 965 ences observed starting from  $t \approx 50$  s can be attributed to the more complex  
 966 constitutive model used in [16]. The results obtained for PIC and APIC are

noticeably different one from another for this 2D gravitation-less simulation, while they were almost identical for the 3D cube bouncing under gravity from Section 4.1.

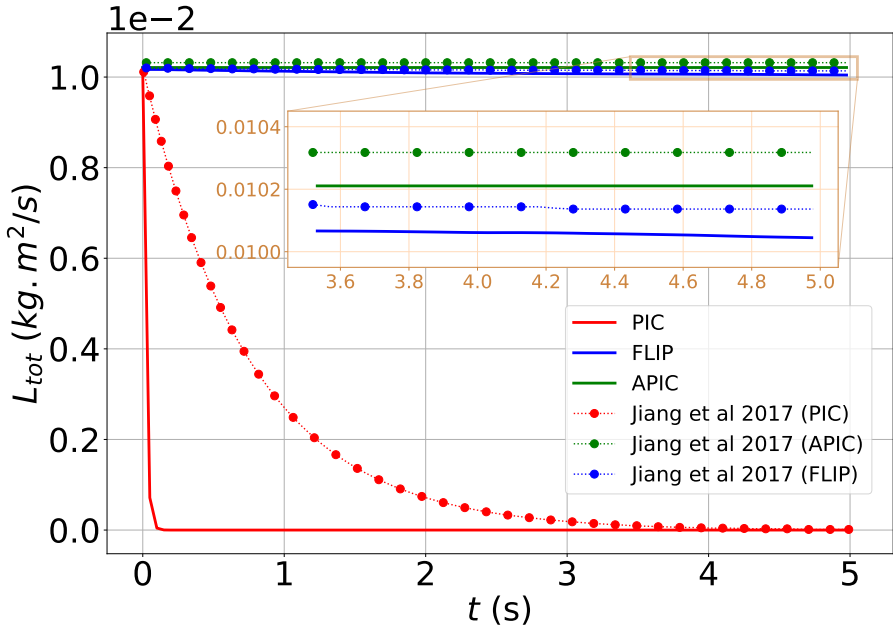
## B.2 Rotating disk in comparison with [18]

The rotating disk simulation from [18], §6.1 therein, is similar to the one in Section 4.2 of the present study, the main differences being the number of dimensions (2D in this Appendix, 3D in Section 4.2) and the shape of the rotating object (a disk in this Appendix, a cube in Section 4.2).

For reproducing that case of a rotating disk, several values of the time step are herein tested, with  $a_\tau$  ranging from  $10^{-2}$  to 0.2 ( $1.4 \times 10^{-5} \text{ s} \leq \Delta t \leq 2.8 \times 10^{-4} \text{ s}$ ), as well as three different motion integration strategies (PIC, FLIP, and APIC) and two different mesh cell sizes, between 0.03125  $m$  and 0.0625  $m$  for a 0.6  $m$  disk diameter.

Figure 25 shows that both PIC and FLIP results are in agreement with the results from [18], although we obtain a slightly lower angular momentum. However, our implementation used in conjunction with PIC dissipated almost all the angular momentum as soon as the first iteration, while the PIC simulation from [18] loose its angular momentum slowly over approximately 4  $s$ . This might be due to the use of a more advanced time integration scheme in [18], which can implicit depending on the value of a parameter  $\lambda$ .

Figure 26 shows that, even though the variations are small, higher time steps actually promote the conservation of angular momentum, thanks to a reduced number of MPM iterations that accumulate errors. More importantly, the size of the mesh cells is shown to have an significant impact on the conservation of the angular momentum in the case of using FLIP.

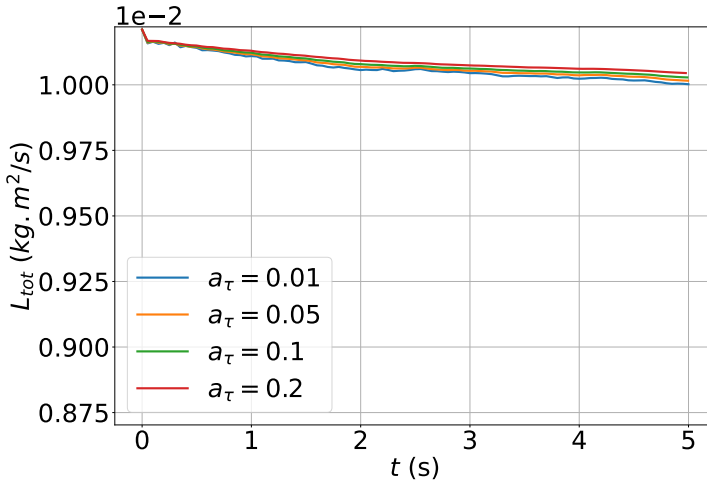
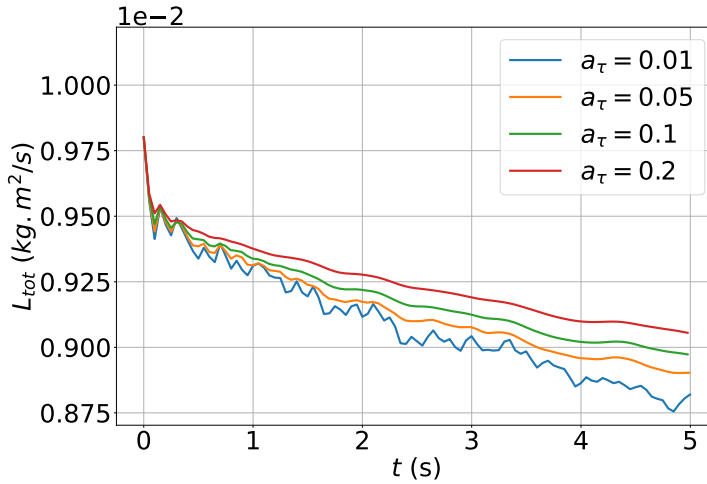


**Fig. 25:** Rotating disk simulation of [18] using various motion integration strategies (for  $a_\tau = 0.2$ )

## C Granular column settling with Cundall's damping for FLIP-based strategies

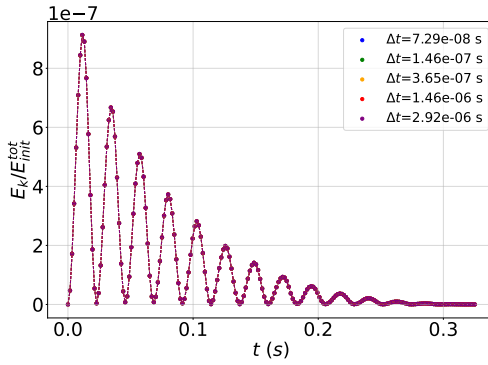
Although not using Cundall's damping is necessary to investigate the energy dissipation coming from the MPM procedure, during the settling phase this consideration lead the column modelled using FLIP-based strategies to perpetually oscillate. As a consequence, the settled states used to initialize the material points positions and stresses before the collapse presented in Figure 14 were obtained after using  $D = 0.1$  for FLIP-based settling simulations. This appendix shows the evolution of  $E_k$  during this settling phase in Figures 27, corresponding to Figures 11.

One can notice that Cundall's damping has no influence on the non-dependance of the results on the motion integration strategy and  $\Delta t$  during

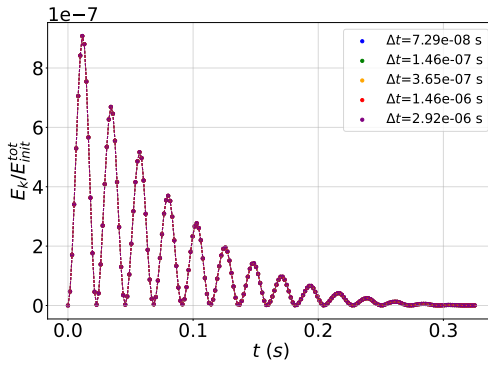
(a)  $l_{cell} = 3.125 \times 10^{-2} \text{ m}$ (b)  $l_{cell} = 6.25 \times 10^{-2} \text{ m}$ 

**Fig. 26:** Effect of  $\Delta t$  and  $l_{cell}$  on the angular momentum during the rotating disk (of  $0.6 \text{ m}$  diameter) simulation of [18] using FLIP

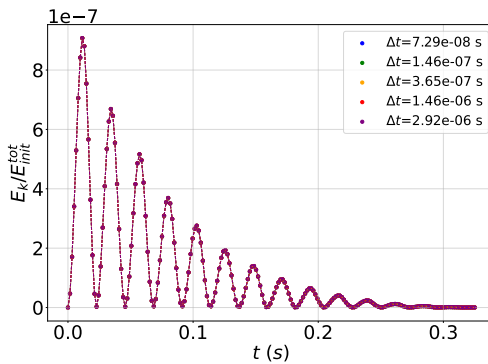
1004 the settling phase. Indeed, FLIP, AFLIP and NFLIP all lead the column to  
 1005 stabilize in as much time with a same period for the pseudo-oscillations, no  
 1006 matter  $\Delta t$ .



(a) FLIP



(b) AFLIP



(c) TFLIP

**Fig. 27:** Kinetic energy during the settling phase for different  $\Delta t$  and motion integration strategy (series S1 with Cundall's damping, FLIP-based)

## 1007 **Statements and Declarations**

### 1008 **Competing interests**

1009 The authors have no competing interests to declare that are relevant to the  
1010 content of this article.

### 1011 **Data deposition**

1012 The source code of the used MPM code will be made available at <https://forgemia.inra.fr/mpm-at-recover/cbgeo> (currently under private access) in  
1013 case of publication.  
1014

## 1015 **Acknowledgements**

1016 Inspiring discussions with Frédéric Dufour (Univ. Grenoble-Alpes, Grenoble  
1017 INP, CNRS, 3SR) are gratefully acknowledged. Constructive and thorough  
1018 reviews by John Nairn (Oregon State University) and another anonymous  
1019 reviewer greatly contributed to improve the manuscript.

## 1020 **References**

- 1021 [1] A Hrennikoff. Solution of problems of elasticity by the framework method.  
1022 *Journal of Applied Mechanics*, 8(4):A169–A175, 1941.
- 1023 [2] R Courant. Variational methods for the solution of problems of equi-  
1024 librium and vibrations. *Bulletin of the American Mathematical Society*,  
1025 49(1):1–23, 1943.
- 1026 [3] Jung-Ho Cheng. Automatic adaptive remeshing for finite element simu-  
1027 lation of forming processes. *International Journal for Numerical Methods*  
1028 *in Engineering*, 26(1):1–18, 1988.

- 1029 [4] SB Petersen and PAF Martins. Finite element remeshing: a metal forming  
1030 approach for quadrilateral mesh generation and refinement. *International*  
1031 *Journal for Numerical Methods in Engineering*, 40(8):1449–1464, 1997.
- 1032 [5] Annika Trädegård, Fred Nilsson, and Sören Östlund. Fem-remeshing tech-  
1033 nique applied to crack growth problems. *Computer Methods in Applied*  
1034 *Mechanics and Engineering*, 160(1-2):115–131, 1998.
- 1035 [6] Massimiliano Cremonesi, Alessandro Franci, Sergio Idelsohn, and Eugenio  
1036 Oñate. A state of the art review of the particle finite element method  
1037 (PFEM). *Archives of Computational Methods in Engineering*, 27(5):1709–  
1038 1735, 2020.
- 1039 [7] S.R. Idelsohn, E. Oñate, and F. Del Pin. The particle finite element  
1040 method: a powerful tool to solve incompressible flows with free-surfaces  
1041 and breaking waves. *International Journal for Numerical Methods in*  
1042 *Engineering*, 61(7):964–989, 2004.
- 1043 [8] Francis H Harlow. The particle-in-cell method for numerical solution  
1044 of problems in fluid dynamics. Technical report, Los Alamos National  
1045 Lab.(LANL), Los Alamos, NM (United States), 1962.
- 1046 [9] Jeremiah U Brackbill and Hans M Ruppel. FLIP: A method for adap-  
1047 tively zoned, particle-in-cell calculations of fluid flows in two dimensions.  
1048 *Journal of Computational physics*, 65(2):314–343, 1986.
- 1049 [10] Deborah Sulsky, Zhen Chen, and Howard L Schreyer. A particle method  
1050 for history-dependent materials. *Computer Methods in Applied Mechanics*  
1051 *and Engineering*, 118(1-2):179–196, 1994.
- 1052 [11] Scott G Bardenhagen and Edward M Kober. The generalized interpo-  
1053 lation material point method. *Computer Modeling in Engineering and*  
1054 *Sciences*, 5(6):477–496, 2004.
- 1055 [12] A. Sadeghirad, R. M. Branmon, and J. Burghardt. A convected particle

- 1056 domain interpolation technique to extend applicability of the material  
1057 point method for problems involving massive deformations. *International*  
1058 *Journal for Numerical Methods in Engineering*, 86(12):1435–1456, 2011.
- 1059 [13] Alban de Vaucorbeil, Vinh Phu Nguyen, Sina Sinaie, and Jian Ying  
1060 Wu. Material point method after 25 years: Theory, implementation, and  
1061 applications. *Advances in applied mechanics*, 53:185–398, 2020.
- 1062 [14] Wojciech T. Sołowski, Martin Berzins, William M. Coombs, James E.  
1063 Guilkey, Matthias Möller, Quoc Anh Tran, Tito Adibaskoro, Seyedmo-  
1064 hammadjavad Seyedan, Roel Tielen, and Kenichi Soga. Chapter Two  
1065 - Material point method: Overview and challenges ahead. *Advances in*  
1066 *Applied Mechanics*, 54:113–204, 2021.
- 1067 [15] Vinh Phu Nguyen, Alban de Vaucorbeil, and Stephane Bordas. *The Mate-*  
1068 *rial Point Method Theory, Implementations and Applications*. Springer,  
1069 2023.
- 1070 [16] E Love and Deborah L Sulsky. An unconditionally stable, energy–  
1071 momentum consistent implementation of the material-point method.  
1072 *Computer Methods in Applied Mechanics and Engineering*, 195(33-  
1073 36):3903–3925, 2006.
- 1074 [17] Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and  
1075 Alexey Stomakhin. The affine particle-in-cell method. *ACM Transactions*  
1076 *on Graphics (TOG)*, 34(4):1–10, 2015.
- 1077 [18] Chenfanfu Jiang, Craig Schroeder, and Joseph Teran. An angular momen-  
1078 tum conserving affine-particle-in-cell method. *Journal of Computational*  
1079 *Physics*, 338:137–164, 2017.
- 1080 [19] Yun Fei, Qi Guo, Rundong Wu, Li Huang, and Ming Gao. Revisiting  
1081 integration in the material point method: a scheme for easier separation  
1082 and less dissipation. *ACM Transactions on Graphics (TOG)*, 40(4):1–16,  
1083 2021.

- 1084 [20] PC Wallstedt and JE Guilkey. Improved velocity projection for the mate-  
1085 rial point method. *Computer Modeling in Engineering and Sciences*,  
1086 19(3):223, 2007.
- 1087 [21] Keita Nakamura, Satoshi Matsumura, and Takaaki Mizutani. Taylor  
1088 particle-in-cell transfer and kernel correction for material point method.  
1089 *Computer Methods in Applied Mechanics and Engineering*, 403:115720,  
1090 2023.
- 1091 [22] Chad C. Hammerquist and John A. Nairn. A new method for material  
1092 point method particle updates that reduces noise and enhances stability.  
1093 *Computer Methods in Applied Mechanics and Engineering*, 318:724–738,  
1094 2017.
- 1095 [23] John A Nairn and Chad C Hammerquist. Material point method simula-  
1096 tions using an approximate full mass matrix inverse. *Computer Methods*  
1097 *in Applied Mechanics and Engineering*, 377:113667, 2021.
- 1098 [24] Alban de Vaucorbeil, Vinh Phu Nguyen, and Christopher R. Hutchinson.  
1099 A total-lagrangian material point method for solid mechanics problems  
1100 involving large deformations. *Computer Methods in Applied Mechanics*  
1101 *and Engineering*, 360:112783, 2020.
- 1102 [25] Marc Kohler, Andreas Stoecklin, and Alexander M. Puzrin. A MPM  
1103 framework for large-deformation seismic response analysis. *Canadian*  
1104 *Geotechnical Journal*, 59(6):1046–1060, 2022.
- 1105 [26] Krishna Kumar, Jeffrey Salmond, Shyamini Kularathna, Christopher  
1106 Wilkes, Ezra Tjung, Giovanna Biscontin, and Kenichi Soga. Scalable and  
1107 modular material point method for large-scale simulations. *arXiv preprint*  
1108 *arXiv:1909.13380*, 2019.
- 1109 [27] Sacha Duverger and Jérôme Duriez. PyCBG, a python module for  
1110 generating CB-Geo MPM input files (1.1.4), 2021.

- 1111 [28] D. Burgess, D. Sulsky, and J.U. Brackbill. Mass matrix formulation of the  
1112 flip particle-in-cell method. *Journal of Computational Physics*, 103(1):1–  
1113 15, 1992.
- 1114 [29] P.C. Wallstedt and J.E. Guilkey. An evaluation of explicit time integration  
1115 schemes for use with the generalized interpolation material point method.  
1116 *Journal of Computational Physics*, 227(22):9628–9642, 2008.
- 1117 [30] Michael Steffen, Robert M Kirby, and Martin Berzins. Analysis and reduc-  
1118 tion of quadrature errors in the material point method (MPM). *Inter-  
1119 national journal for numerical methods in engineering*, 76(6):922–948,  
1120 2008.
- 1121 [31] Louis Moresi, Frédéric Dufour, and H-B Mühlhaus. A lagrangian integra-  
1122 tion point finite element method for large deformation modeling of vis-  
1123 coelastic geomaterials. *Journal of computational physics*, 184(2):476–497,  
1124 2003.
- 1125 [32] Elliot James Fern and Kenichi Soga. The role of constitutive models  
1126 in mpm simulations of granular column collapses. *Acta Geotechnica*,  
1127 11(3):659–678, 2016.
- 1128 [33] Gregory Abram, Andrew Solis, Yong Liang, and Krishna Kumar. In situ  
1129 visualization of regional-scale natural hazards with galaxy and material  
1130 point method. *Computing in Science & Engineering*, 24(2):31–39, 2022.
- 1131 [34] Weijian Liang, Shiwei Zhao, Huanran Wu, and Jidong Zhao. Bear-  
1132 ing capacity and failure of footing on anisotropic soil: A multiscale  
1133 perspective. *Computers and Geotechnics*, 137:104279, 2021.
- 1134 [35] JU Brackbill. The ringing instability in particle-in-cell calculations of  
1135 low-speed flow. *Journal of Computational Physics*, 75(2):469–492, 1988.
- 1136 [36] M Steffen, PC Wallstedt, JE Guilkey, RM Kirby, and M Berzins. Exam-  
1137 ination and analysis of implementation choices within the material

- 1138 point method (mpm). *Computer Modeling in Engineering and Sciences*,  
1139 31(2):107–127, 2008.
- 1140 [37] Quoc-Anh Tran and Wojciech Sołowski. Temporal and null-space filter for  
1141 the material point method. *International Journal for Numerical Methods  
1142 in Engineering*, 120(3):328–360, 2019.
- 1143 [38] SG Bardenhagen. Energy conservation error in the material point method  
1144 for solid mechanics. *Journal of Computational Physics*, 180(1):383–403,  
1145 2002.
- 1146 [39] Martin Berzins. Energy conservation and accuracy of some MPM  
1147 formulations. *Computational Particle Mechanics*, pages 1–13, 2022.
- 1148 [40] Chuyuan Fu, Qi Guo, Theodore Gast, Chenfanfu Jiang, and Joseph Teran.  
1149 A polynomial particle-in-cell method. *ACM Transactions on Graphics  
1150 (TOG)*, 36(6):1–12, 2017.
- 1151 [41] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and  
1152 Andrew Selle. A material point method for snow simulation. *ACM  
1153 Transactions on Graphics (TOG)*, 32(4):1–10, 2013.
- 1154 [42] Kun Zhang, Shui-Long Shen, Annan Zhou, and Daniel Balzani. Trun-  
1155 cated hierarchical b-spline material point method for large deformation  
1156 geotechnical problems. *Computers and Geotechnics*, 134:104097, 2021.
- 1157 [43] Joel Wretborn, Rickard Armiento, and Ken Museth. Animation of crack  
1158 propagation by means of an extended multi-body solver for the material  
1159 point method. *Computers & Graphics*, 69:131–139, 2017.
- 1160 [44] Weijian Liang and Jidong Zhao. Multiscale modeling of large deforma-  
1161 tion in geomechanics. *International Journal for Numerical and Analytical  
1162 Methods in Geomechanics*, 43(5):1080–1114, 2019.
- 1163 [45] John A. Nairn. Numerical simulation of orthogonal cutting using the  
1164 material point method. *Engineering Fracture Mechanics*, 149:262–275,  
1165 2015.

- 1166 [46] Anura3D MPM Research Community. *Anura3D Scientific Manual version*  
1167 *2022*, 2022.
- 1168 [47] John A Nairn. Material point method calculations with explicit cracks.  
1169 *Computer Modeling in Engineering and Sciences*, 4(6):649–664, 2003.
- 1170 [48] Deborah Sulsky, Shi-Jian Zhou, and Howard L. Schreyer. Applica-  
1171 tion of a particle-in-cell method to solid mechanics. *Computer Physics*  
1172 *Communications*, 87(1):236–252, 1995.
- 1173 [49] Stephen P. Timoshenko and J. N. Goodier. Theory of elasticity (2nd ed.).  
1174 *McGraw-Hill*, 1951.
- 1175 [50] W. T. Sołowski and S. W. Sloan. Evaluation of material point method for  
1176 use in geotechnics. *International Journal for Numerical and Analytical*  
1177 *Methods in Geomechanics*, 39(7):685–701, 2015.
- 1178 [51] GB Crosta, S Imposimato, and D Roddeman. Numerical modeling of 2-  
1179 d granular step collapse on erodible and nonerodible surface. *Journal of*  
1180 *Geophysical Research: Earth Surface*, 114(F3), 2009.
- 1181 [52] L Girolami, V Hergault, G Vinay, and A Wachs. A three-dimensional  
1182 discrete-grain model for the simulation of dam-break rectangular col-  
1183 lapses: comparison between numerical results and experiments. *Granular*  
1184 *Matter*, 14(3):381–392, 2012.
- 1185 [53] Hao Xiong, Zhen-Yu Yin, François Nicot, Antoine Wautier, Marie Miot,  
1186 Félix Darve, Guillaume Veylon, and Pierre Philippe. A novel multi-  
1187 scale large deformation approach for modelling of granular collapse. *Acta*  
1188 *Geotechnica*, 16(8):2371–2388, 2021.
- 1189 [54] Nhu H. T. Nguyen, Ha H. Bui, and Giang D. Nguyen. Effects of material  
1190 properties on the mobility of granular flow. *Granular Matter*, 22(3), 2020.
- 1191 [55] Lydie Staron and EJ Hinch. Study of the collapse of granular  
1192 columns using two-dimensional discrete-grain simulation. *Journal of Fluid*

- 1193 *Mechanics*, 545:1–27, 2005.
- 1194 [56] Gert Lube, Herbert E Huppert, R Stephen J Sparks, and Armin Fre-  
1195 undt. Collapses of two-dimensional granular columns. *Physical Review E*,  
1196 72(4):041301, 2005.
- 1197 [57] Arnold N Lowan, Norman Davids, and Arthur Levenson. Table of the  
1198 zeros of the Legendre polynomials of order 1-16 and the weight coeffi-  
1199 cients for Gauss’ mechanical quadrature formula. *Bull. Amer. Math. Soc.*,  
1200 48(12):739–743, 1942.
- 1201 [58] Rodaina Aboul Hosn, Luc Sibille, Nadia Benahmed, and Bruno Chareyre.  
1202 Discrete numerical modeling of loose soil with spherical particles and  
1203 interparticle rolling friction. *Granular matter*, 19(1):1–12, 2017.
- 1204 [59] P. A. Cundall. Distinct element models of rock and soil structure. In E. T.  
1205 Brown, editor, *Analytical and Computational Methods in Engineering*  
1206 *Rock Mechanics*, pages 129–163. George Allen and Unwin, 1987.
- 1207 [60] Stefano Utili, Tao Zhao, and GT Houlsby. 3d dem investigation of granular  
1208 column collapse: evaluation of debris motion and its destructive power.  
1209 *Engineering geology*, 186:3–16, 2015.
- 1210 [61] E Lajeunesse, JB Monnier, and GM Homsy. Granular slumping on a  
1211 horizontal surface. *Physics of fluids*, 17(10):103302, 2005.