Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-183-RC1, 2020

Geoscientific Model Development Discussions

Interactive comment on "Fast and efficient MATLAB-based MPM solver (fMPMM-solver v1.0)" *by* Emmanuel Wyser et al.

Anonymous Referee #1

Received and published: 11 August 2020

The paper presents an explicit vectorised form of the material point method for the generalised interpolation and CPDI2 variants of the method. The paper is reasonably written but is missing many details on the implemented algorithms and the numerical analyses are too focused on reproducing the results of others rather than on numerical performance which is the main trust of the article.

I have the following specific comments:

- the introduction to the paper appears to have picked a random selection of MPM articles rather than focusing on articles that look at the numerical implementation of the method. The introduction should be made more coherent and focused.

- the authors have not picked up on any of the papers that extend the MILAMIN ap-

proach. In particular the authors should refer to the following paper that extends the MILAMIN ideas to non-linear problems: O'Sullivan, S, Bird, R.E., Coombs, W.M. & Giani, S. (2019). Rapid non-linear finite element analysis of continuous and discontinuous Galerkin methods in MATLAB. Computers and Mathematics with Applications 78(9): 3007-3026. There are others and the authors should review the appropriate literature.

- If performance is the focus, why use MATLAB? Why not adopt one of the existing MPM codes that are written in compiled code which will always be faster than MATLAB? Such as: jixiefx.com, github.com/yuanming-hu/taichi_mpm, cimne.com/kratos/, github.com/nairnj/nairn-mpm-fea, github.com/cb-geo/mpm/, sourceforge.net/p/mpmgimp, github.com/xzhang66/MPM3D-F90. There may be others.

- the authors mention an implicit implementation but do not present any results in terms of the speed gains of the implicit vectorised algorithm. These should be included and the vectorised algorithm explained.

- CPDI2 methods suffer from issues associated with domain distortion and are not suitable for problems involving large shear/rotation. This point should be acknowledged, see for example: Wang, L., Coombs, W.M. , Augarde, C.E. , Cortis, M. Charlton, T.J. , Brown, M.J. Knappett, J., Brennan, A. Davidson, C. Richards, D. & Blake, A. (2019). On the use of domain-based material point methods for problems involving large distortion. Computer Methods in Applied Mechanics and Engineering 355: 1003-1025.

- What large deformation formulation has been used in this paper in terms of stresses and strains? Logarithmic strains and Kirchhoff stresses? Explain and justify the large deformation framework.

- How has the plasticity algorithm been implemented within the large deformation framework? The authors mention a prediction/correction type algorithm so how have they recovered the additive decomposition of the elastic and plastic strains from the

multiplicative decomposition of the deformation gradient? Critical details are missing here.

- How are boundary conditions imposed in this model?

- Figure 6 - what causes the step in the red line around 10^3 material points? Is there a shift in terms of the cost within the algorithm?

- It is well known that MPMs can provide a reasonable global approximation (in terms of force-displacement response) but the computed stress field can be spurious/highly oscillatory due to issues such as locking and/or cell crossing. The authors should present the predicted stress response for the numerical examples presented in the paper and compare them to analytical solutions where available.

- The elastic column problem will only have around 2% deformation in terms of the deformed to original height so this problem is not a good test of the convergence of MPMs. The authors refer to the work of Coombs et al. for this problem but they show convergence for the case where the column compresses to around 50% of its initial height. Run the convergence for the case with a Young's modulus of 10kPa, not 1MPa.

- How have the authors avoided volumetric locking when using isochoric plastic flow? What do the pressure distributions look like through the deformed domains?

- the various comments on the use of cpGIMP and uGIMP are confused and misleading. Just because an analysis is stable it does not mean that it is "appropriate" as the results may be meaningless. The authors cite the work of Coombs to justify the use of the stretch to update the domains, however this technique does not work for problems involving simple shear type deformation, refer to Table 2 of: Coombs, WM, Augarde, CE, Brennan, AJ, Brown, MJ, Charlton, TJ, Knappett, JA, Ghaffari Motlagh, Y & Wang, L (2020). On Lagrangian mechanics and the implicit material point method for large deformation elasto-plasticity. Computer Methods in Applied Mechanics and Engineering 358: 112622. The authors later adopt a determinant of the deformation

gradient-type updating method but this has been shown to not converge for simple compression problems as the domains artificially shrink in the non-compressed direction and overlap in the compressed direction (see Figure 8 from the above reference). For example, the statement on lines 76-77 is not correct, or rather it is only correct for some types of problem.

- The speed gains of MILAMIN come from the combination of blocking and vectorisation. Have both techniques been used in this paper? If so, what are the relative speed gains from the different sources? How does the speed gain change with different block sizes?

- The authors have not explained by MATLAB is inefficient when working with small amounts of data. This point should be discussed with reference to CPU cache size and RAM-to-cache overheads.

- It would be more appropriate to present the speed gains in terms of flops. There should be a peak in performance if you consider large enough problems.

Other minor points: - Figure 2 is misleading as the GIMP domain is fully inside a single element and will have the same connectivity as the standard MP case. - Line 162 - the authors mention a 30% speed up - what problem/size of problem/number of points, etc? - where are material points located within the elements when the problems are set up? - some key information is missing from the numerical analysis, such as timestep sizes and total times which would make it impossible to reproduce the results. This information must be added. - incomplete sentence on line 359.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2020-183, 2020.