Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-183-AC2, 2020

Geoscientific Model Development Discussions

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

Emmanuel Wyser et al.

emmanuel.wyser@unil.ch

Received and published: 1 September 2020

Referee's comment 1

This manuscript presents a vectorized material point method for MATLAB and quantifies the increase in computational efficiency gained from using vectorized rather than iterative code. To my knowledge, this paper provides the only formal analysis of the performance gains from vectorizing MPM code. The presented vectorization approach could be easily implemented within existing and future MPM models. However, as already thoroughly noted by Referee #1, many details of the algorithms, setup of simulations, and numerical analysis are missing. I would like to add the following comments to those already given.

Author's reply 1

We agree many details of the algorithms are missing and, we will provide more details in the revised version, i.e. a clearer description of the constitutive model chosen, the large deformation framework based on the Jaumann rate. In addition, we will also provide further details concerning the setup of the simulation of the cases presented.

General comments:

Referee's comment 2

I found the structure of the paper to be confusing, especially in Section 4, where results from five test cases are reported. These test cases are all nearly exact reproductions of previously-published work. The first two test cases – the elastic compaction of a column (Section 4.1) and elastic cantilever beam (Section 4.2) – appear to solely serve as benchmark examples for verification of the MPM model, though this is not clearly stated. The third test case – elasto-plastic column collapse (Section 4.3) – also appears to serve as further verification of the model until it is used again in Section 4.4, where the main results of the paper concerning the computational efficiency gained from vectorization are presented. A fourth test case (collision of two elastic disks) is also presented in 4.4 for further analysis of computational efficiency. The final test case (elasto-plastic landslide) is then presented in Section 4.5, which seems to serve the dual purpose of further model verification and a geomechanical application.

Author's reply 2

It is true that the structure might be confusing. We will focus on Section 4 and make our point clearer by adding, as suggested in the following by the Referee, a quick introduction to this section. We think it would improve the readability of this section. We will also clearly state our motivation of exact reproduction of previously published work. We think it is also important to showcase the efficiency of our solver with benchmarks from past studies. However, we will reorganise the subsection in which computational performances are shown, in order to make our point more obvious to the reader.

Referee's comment 3

The motivation behind most of the test cases is not clear, especially on the first read. Section 4 would benefit from a short introduction (before 4.1) that outlines what test cases were selected and for what purpose.

Author's reply 3

Thank you for this comment. We agree that the motivation behind most of the test cases is not clear and, we will add a short introduction in Section 4, before 4.1 as suggested. We will also present the test cases and, we will motivate our choices for each of these to clarify our motivation.

Referee's comment 4

It may help readability if the test cases for elasto-plastic column collapse and collision of two elastic disks are separated into an entirely different section from the rest of the examples, as these two test cases provide the main results in the paper regarding the computational efficiency gained by vectorization.

Author's reply 4

It is a good suggestion and it should even significantly help the readability of the manuscript. We will create a new section focused on the computational efficiency, divided in two subsections, i.e., the first one for the elastic disk case and the second one for the elasto-plastic column collapse.

Referee's comment 5

Many of the statements regarding the effectiveness of cpGIMPM vs. uGIMPM vs. CPDI are misleading or lacking in detail:

Author's reply 5

We agree with the Referee that the statements related to the effectiveness of uGIMPM, cpGIMPM and CPDI are misleading or lacking in detail. Therefore, we created a new subsection 2.2 Domain-based material point method variants, which summarises the major difference between GIMPM and CPDI (the way the material point's domain is updated). We also present the different domain updating methods for GIMPM and we more clearly highlight the suitability of each of these domains updating methods based on the investigation 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, and, 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. Finally, we explicitly state at the end of new 2.2 subsection that the domain-based method as well as the domain update method should be carefully chosen accordingly to the deformation mode expected for a given case. Moreover, we also clearly state that the domain update method will be clearly stated in each case presented in the Results section. We believe that now this additional 2.2 subsection provides a better overview of both domain-based method and domain update method.

Referee's comment 6

Line 288: in what way did domain updates based on the deformation gradient result in failure? The domain updates based on the deformation gradient showed some flaws of GIMP-based implementation, i.e. the domain artificially shrinks under large rotation, which was already demonstrated in Fig.4 in 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. As such, we will clearly mention the reason why the simulation in which the domain updates based on the deformation gradient failed.

Referee's comment 7

Line 301: "[The elasto-plastic MPM solver] demonstrates the inability of the MPM variants based on a domain update (GIMPM or CPDI) to resolve extremely large plastic deformations when relying on the normal components of the deformation gradient or its stretch part to update the material point domain". This is too general of a statement. There are many cases in which these domain updates would work well; for example, if simple shear is minimal and the "stretch" update is used (Coombs et al 2020). A similarly-flawed statement is made in the conclusion section (lines 394-396). Better conclusions regarding GIMPM/CPDI might incorporate the performance gains reported using the vectorization scheme for calculation of the shape functions and the difference in computational efficiency measured for GIMPM vs CPDI.

Author's reply 7

We agree that the statement on Line 301 is too general. As such, we will specify our concern regarding the deformation mode involved avoiding such flawed statements in the revised version of the manuscript, referring explicitly to the work 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. As suggested by the Referee, we will also focus our discussion on the performance gain of GIMPM and CPDI variants, which is indeed the main point of our contribution.

Referee's comment 8

Line 305: As already pointed out by Referee #1, it should be noted that the determinant of the deformation gradient-type GIMPM domain update is problematic for simple

compression problems. Have the authors tried updating the GIMPM domains with the "corner" scheme from Eqs 35-37 in Coombs et al (2020)? Perhaps it would be more robust.

Author's reply 8

This is a good comment. First, we will add a presentation of the suitability of domain updating methods in GIMPM for the possible deformation modes, directly referring to the work 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. This is important and is lacking up to now. Concerning the investigation of the corner update, we did not have tried to implement it. We decided to quickly investigate this scheme but we quickly figured out it was necessary to compute additional shape functions between material point's corners and nodes, similarly to the CPDI variant. Our concern comes from the need to compute these additional quantities, alongside to the shape function relating material points to their associated nodes. This extra cost is significant, especially when our major concern is computational performance. We therefore decided not to pursue such investigation for that reason.

Referee's comment 9

There does not appear to be any reference to Fig. 1 in the main text. The caption for Fig. 1 also appears to lack any description of panels B and C.

Author's reply 9

Indeed, there was no reference to Fig. 1 in the main text. We added a description of panels B and C as highlighted by the Referee and, we refer now to this Fig. 1 in the main text, as mentioned in the following reply #10.

Referee's comment 10

The list of the three steps of a typical MPM cycle at the beginning of Section 3.1 seems misplaced and is somewhat repetitive of the description of MPM in the previous sections. I suspect this list should link with Fig. 1 and may be more appropriately located within Section 2

Author's reply 10

We agree with the Referee; this list is misplaced. Consequently, we moved it in Section 2 and, we made a direct reference to this list with Fig. 1. Minor comments:

Referee's comment 11

Fig 2. In the caption, "GIMP" should be changed to "GIMPM" to match the rest of the paper. The authors have already fixed the error in Fig. 2b regarding which nodes are associated with the GIMPM domain

Author's reply 11

Right, we will make the change in the revised paper.

Referee's comment 12

Line 39: which numerical considerations from MILAMIN are used? This is not specifically addressed

Author's reply 12

Due to the comment of the first reviewer concerning the lack of focus in the introduction section, we partially have rewritten it. We now specify in the introduction section the considerations (from MILAMIN but also from the works of Birds et al, 2017 and O'Sullivan et al, 2019), which are the use of the function accumarray(), to reduce the number of BLAS calls.

Referee's comment 13

several citations are missing parentheses (e.g. lines 52 and 84).

Author's reply 13

We will fix these numerous citation problems in the revised paper.

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