2nd International Conference on the Material Point Method for Modelling Soil-Water-Structure Interaction

Anura3D
MPM Research Community

Paper number:
mpm2019.xxx
www.anura3d.com

On Implementation Aspects of Implicit MPM for 3D Analysis

Lei Wang[1*], Michael Cortis[2], William Coombs[1], Charles Augarde[1], Michael Brown[3], Jonathan Knappett[3], Andrew Brennan[3], Craig Davidson[3], David Richards[4], and Anthony Blake[4]

[1] Department of Engineering, University of Durham, Durham, UK
[2] Coventive Composites, Chestereld, UK
[3] School of Science and Engineering, University of Dundee, Dundee, UK
[4] Faculty of Engineering and the Environment, University of Southampton, UK

[*] E-mail: lei.wang@durham.ac.uk

ABSTRACT

An implicit MPM program has been developed to model the 3D installation of screw piles for offshore wind turbine foundations as part of a UK EPSRC-funded project. The program is to be used to study the torque and vertical force requirements during installation of screw piles for different geometries and soil conditions, and has required the development of innovative numerical techniques which may be of interest for other geotechnical problems. In this paper, we introduce some of the features developed, including an efficient walking-in-triangulation algorithm for searching material points and an $\bar{\boldsymbol{F}}$-patch method for avoiding volumetric locking.

KEY WORDS: 3D MPM, implicit, tetrahedron mesh, volumetric-locking, efficiency

INTRODUCTION

The MPM is ideal for application to large deformation problems in geotechnics and elsewhere, e.g. (Ceccato et al., 2016; Stomakhin et al., 2013) and in this paper we highlight some features that have been developed in an implicit MPM code, to model the installation of screw piles in soft ground. A key feature of this particular geotechnical problem is the need to model the large rotation of the complex geometry of a screw pile, something not attempted to date with the MPM to our knowledge. In developing the code, we have investigated standard MPM as well as CPDI approaches, and found that the former is more suitable (Wang et al., 2018). Therefore, in the developments described here we only consider the standard MPM.

The code uses unstructured tetrahedral meshes to cope with the complex geometry of a screw pile and the need for local refinement. A moving mesh concept is used, similar in nature to (Jassim, 2013) for instance, but for rotation rather than translation (Wang et al., 2018). The first interesting feature to be described here relates to the search needed to locate material points in the unstructured mesh. An efficient walk-in-triangulation algorithm is used which relies on an efficient procedure for building mesh neighbourhoods (Devillers et al., 2002). The second development covered here is mitigation of volumetric locking. The $\bar{\boldsymbol{F}}$ approach, established for standard finite elements in (de Souza Neto et al.,1996), has been implemented in the standard MPM for quadrilateral and hexahedral elements in (Coombs et al., 2018). Here we describe a similar approach applied to tetrahedral elements in the standard MPM, based on an alternative $\bar{\boldsymbol{F}}$-patch approach for finite elements (de Souza Neto et al., 2005).

MATERIAL POINT SEARCH ALGORITHM

Determining the *parent element*, i.e. the element in which the material point is located, is a crucial step at the beginning of each load step. For a regular grid, this can be directly computed based on the coordinates of material points and the grid size, but this is not the case for an unstructured mesh and an alternative efficient and robust search algorithm is required. In the code used here, at the beginning of a simulation, the material points are generated as Gauss points of the elements in the background mesh, and therefore at this stage the parent element of each material point is directly determined. The search algorithm adopted at subsequent steps is a "walking-in-

triangulation" procedure. Based on the coordinates of a material point, updated at the end of the previous load step, the current parent element is determined through a walking path composed of tested elements, starting from the previous parent element, in the previous load step. As shown in Figure 1 the old parent element is first tested, followed by any of the neighbour elements with a visible line in 2D or a face in 3D; and then the neighbour element is tested. This repeats until the correct parent element is found, as shown in Algorithm 1. The *visible line in 2D or face in 3D* is the line or face through which the element and the material point can be divided into two half spaces. An element with a visible line or face is denoted as a visible element.

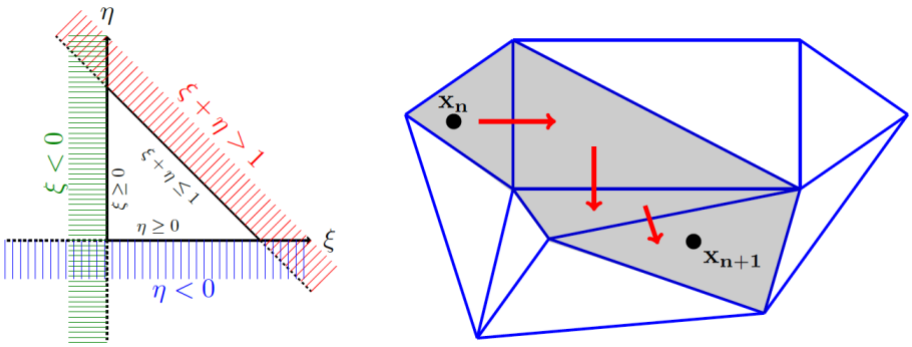

Figure 1 Conditions for different areas with respect to the standard reference element. The red arrows and shaded elements indicate the walking path for the material point, which has moved right and down over the *n*-th load step.

Algorithm 1: Seeking the parent element for a material point at beginning of the $(n+1)$-th load step.

E^n is the parent element of a material point MP_i;
set E^n as the walking path head;
while *MP_i is not in the element indicated by the path head* **do**
  replace the path head by any of the neighbour visible elements;
end
The element indicated by the path head is the current parent element E^{n+1} for the material point MP_i;

The visibility of all faces of a tested element can be determined through the relative position of the material point by mapping them into the isoparametric space. For an element, the isoparametric coordinates, (ξ, η, ζ), of the material point is computed. Based on the value of this coordinate, the relative location of the point with respect to this element can be determined, as demonstrated in Figure 1 for a 2D problem. The 3D problem is similar. Therefore, the visibility of all faces can be determined once, i.e.

- if $\xi + \eta + \zeta > 1$, the first face is visible,
- if $\xi < 0$, the second face is visible,
- if $\eta < 0$,the third face is visible,
- if $\zeta < 0$, the fourth face is visible,
- if $\xi + \eta + \zeta \leq 1,\ \xi \geq 0, \eta \geq 0, \zeta \geq 0,$ the point is inside of the element.

The success of this walking algorithm also depends on *efficiently* obtaining all neighbours of an element. In the program used here, this neighbourhood is built only once at the beginning of the simulation, because it is assumed that the topology of the mesh is not changed over load-steps[1]. A tetrahedral element has four neighbours, one associated with each face. Assume there are *n* tetrahedrons in a mesh, then a matrix of dimension $n \times 4$, $\boldsymbol{NE}$, can be used to store the neighbourhood relationship. Each row of the $\boldsymbol{NE}$ is built one-to-one corresponding to a row of the element connectivity $\boldsymbol{CONN}$. For example, the element id of the first neighbour in $\boldsymbol{NE}(i, 0)$ is the element which has a face opposite to the first vertex $\boldsymbol{CONN}(i, 0)$. If a face is a boundary, its neighbour element id is set to be -1. The algorithm to build $\boldsymbol{NE}$ is shown in Algorithm 2.

[1] Note that the element neighbourhood list could be reconstructed if the original mesh was replaced.

Algorithm 2: Build neighbourhood data structure **NE**

```
NE_{n×4} is initialized with -1;
for element i, i=0,1,···,n-1 do
    Get all faces of this element;
    for face f_j, j=0,···,3 do
        Get all elements with f_j as an face;
        if there are two elements then
            Assign the element id, not equal to i, into NE(i, j);
        end
    end
end
```

Some efficiencies can be obtained within the procedure outlines above. Computing the inverse of a matrix, which only depends on the nodal coordinate of an element, to obtain (ξ, η, ζ) of a material point, is time-consuming. This inverse is only however computed when an element is checked the first time, and is then saved. Therefore, when computing the isoparametric coordinates of other material points with respect to this element, this time-consuming operation is avoided. The isoparametric coordinates obtained can also be used later when computing stresses and internal force.

$\bar{F}$-PATCH MPM

The issue of volumetric locking is well-known for standard finite elements, affecting modelling of incompressible or nearly-incompressible materials and isochoric plasticity. An $\bar{F}$ approach modified for the MPM is described in (Coombs et al., 2018) for quadrilateral and hexahedral meshes, and here we explain something similar for tetrahedral meshes. In the code used here, linear displacement tetrahedral elements are used for simplicity and robustness. An $\bar{F}$-patch method, described in detail for standard finite elements in (de Souza Neto et al., 2005), is used here suitable for linear triangle or tetrahedral elements. We have tailored this approach for the MPM in the following two aspects: 1) a patch is generated by subdividing a 10-node quadratic tetrahedron element into 8 linear tetrahedron elements (Figure 2), and 2) only elements including material points are considered in a patch.

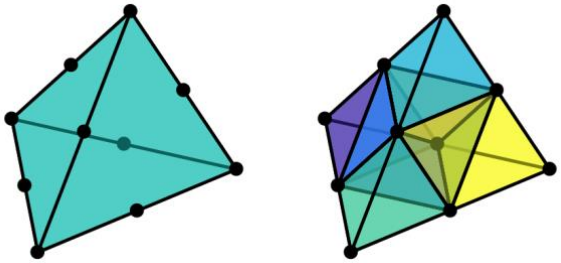

Figure 2 A quadratic 10-node tetrahedral element, regarded as a patch, further subdivided into 8 linear tetrahedral elements.

The essential idea of the $\bar{F}$-patch is to modify the deformation gradient for an element with consideration of all elements in a surrounding 'patch'. For example, for the element e in a patch $\mathscr{P}$,

$$\overline{\mathbf{F}}_e = \left[\frac{v_{patch}}{V_{patch}(\det \mathbf{F}_e)}\right]^{1/3} \mathbf{F}_e, \tag{1}$$

where $\mathbf{F}_e$ is the deformation gradient obtained from the standard linear displacement interpolation and v_{patch} and V_{patch} denote, respectively, the deformed and undeformed volume of the patch. As a result, the computation of the internal force and element tangent stiffness is also modified. For the internal force, it is necessary to replace $\mathbf{F}_e$ by $\bar{\mathbf{F}}_e$. The tangent stiffness of an element e also has contributions from all elements in its patch as

$$\mathbf{K}_{ee} = \int_{\Omega_e^t} \mathbf{G}_e^T \mathbf{a} \mathbf{G}_e \, dv + \left(\frac{v_e}{v_{patch}} - 1 \right) \int_{\Omega_e^t} \mathbf{G}_e^T \mathbf{q} \mathbf{G}_e \, dv \tag{2}$$

$$\mathbf{K}_{es} = \frac{v_e}{v_{patch}} \int_{\Omega_e^t} \mathbf{G}_e^T \mathbf{q} \mathbf{G}_s \, dv, \qquad s \in \mathscr{P}; \quad s \neq e, \tag{3}$$

where $\mathbf{a}$ is the matrix form of the fourth order spatial elasticity tensor now evaluated at $\mathbf{F}_e = \bar{\mathbf{F}}_e$, $\mathbf{G}_i$ denotes the conventional discrete gradient operator of an element and $\mathbf{q}$ is the matrix form of the fourth order tensor defined by

$$\mathbf{q} = \frac{1}{3}\mathbf{a} : (\mathbf{I} \otimes \mathbf{I}) - \frac{2}{3}(\boldsymbol{\sigma} \otimes \mathbf{I}), \tag{4}$$

in which $\mathbf{I}$ is the identity matrix and $\boldsymbol{\sigma}$ is the Cauchy stress.

RESULTS

As a demonstration of the capabilities of the code which includes the above features, some numerical examples are now presented.

Simple Stretch

The uniaxial stretch of a cube (initial edge length 2) is the first example. The material response is elasto-plastic with a von Mises yield surface where material parameters are $E = 10^3, \nu = 0$, and yield strength $\rho_c = 400$ with consistent units. The cube is discretised into five 10-noded tetrahedral elements (each containing eleven material points). There are five patches for the $\bar{\boldsymbol{F}}$-patch computation, which are then subdivided into forty linear tetrahedrons (eight per initial 10-noded tetrahedron as seen in Figure 3. Roller boundary conditions are applied on the three surfaces, $x = 0, y = 0, z = 0$, and displacement boundary condition, $\Delta u = 0.4$, applied on the surface, $y = 2$. The mesh is simply stretched along the y-axis direction and since the boundary of the physical domain aligns with the mesh, the Dirichlet boundary conditions can be applied straightforwardly.

The mesh and material point distribution at load step 17 is shown in Figure 3. The Kirchhoff stress component τ_{22} at all material points are plotted against the load step for both unmodofied ($\mathbf{F}$) and modified $\bar{\boldsymbol{F}}$-patch methods are plotted in Figure 4. The average value agrees with the analytical solution but there is variation as indicated, more pronounced in the unmodified case. It is clear that oscillations in stress have been reduced by using the $\bar{\boldsymbol{F}}$-patch approach. The smoothing effect seen with the $\bar{\boldsymbol{F}}$-patch approach occurs because the stiffness and internal force for an element receives contributions from other elements in the same patch.

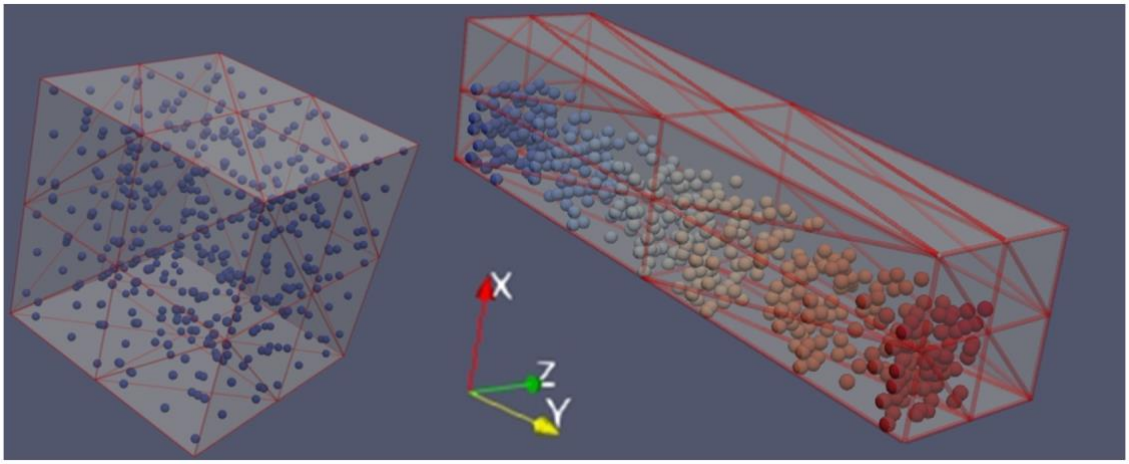

Figure 3 The mesh and material points at the start and at load step 17 in the simulation of the uniaxial stretch problem.

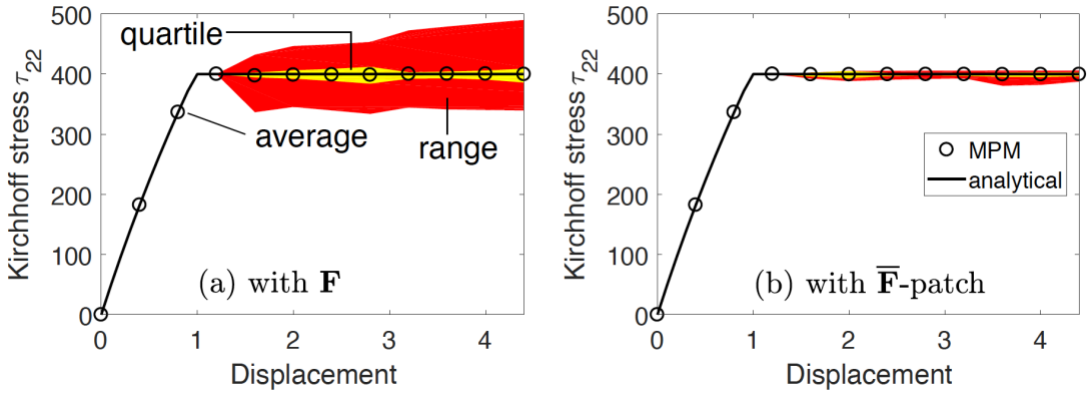

Figure 4 Kirchhoff stress component τ_{22} at all material points with average, range and quartile.

Vane Shear

The second example to be modelled is a vane shear test, which is a simple field test for obtaining undrained soil strength where the vane (a pair of steel planes set in a cruciform as seen in Figure 5) is pushed into the ground and rotated and the torque against rotation noted. The modelling here follows the 2D simulation in (Griffiths and Lane, 1990) where the radius of vane is 0.05m denoted by r_v and other dimensions are defined in terms of multiples of r_v, i.e. radius of external circle is $3r_v$, height of the vane is r_v, and the thickness of the vane blade is $0.1r_v$. The soil is discretized into 95456 tetrahedral elements (Figure 5) and material parameters are the same as those in (Griffiths and Lane, 1990), i.e. are $E = 10^8$ Pa, $\nu = 0.3$, and $\rho_c = \sqrt{3} \times 10^4$ Pa. The external surface is fixed, while both top and bottom are constrained by a roller boundary condition. An incremental rotation, 5×10^{-5} rad per step, is applied on the vane blade surfaces. Figure 5 shows torque measured in the simulation against rotation for both unmodified (**F**) and modified ($\overline{\boldsymbol{F}}$-patch) approaches where it is clear that the latter avoids volumetric locking.

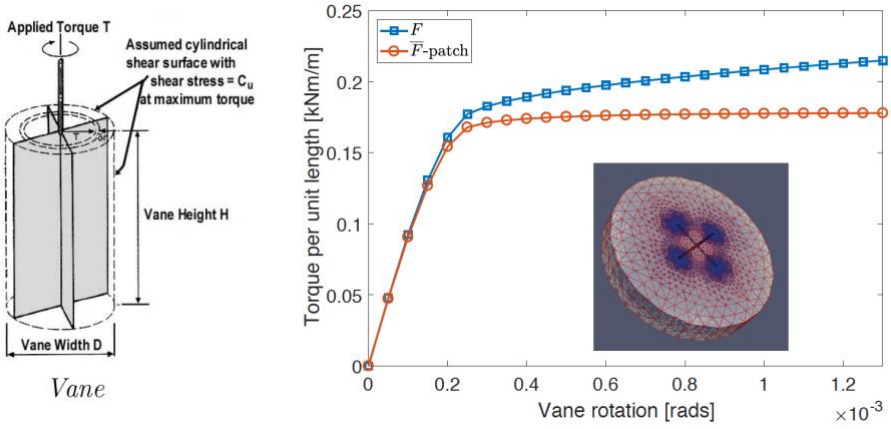

Figure 5 The simulation of the vane shear test.

CONCLUSIONS

Use of the standard MPM for challenging geotechnical problems has prompted the development of the novel techniques and procedures which are described above. Both developments described here are required due to the use of an unstructured, tetrahedral background mesh. A novel and robust searching algorithm is described for the location of material points in elements during an analysis, and a method for avoiding volumetric locking has also been demonstrated.

ACKNOWLEDGEMENTS

We are grateful for the support by the UK Engineering and Physical Sciences Research Council grant No. EP/N006054/1.

REFERENCES

Ceccato F, Beuth L, Vermeer PA & Simonini P. (2016). Two-phase material point method applied to the study of cone penetration. *Computers and Geotechnics*, 80: 440–452.

Coombs WM, Charlton TJ, Cortis M & Augarde CE. (2018). Overcoming volumetric locking in material point methods. *Computer Methods in Applied Mechanics and Engineering*, 333:1–21.

de Souza Neto EA, Andrade Pires FM & DRJ Owen DRJ. (2005). F-bar-based linear triangles and tetrahedra for finite strain analysis of nearly incompressible solids. part I: formulation and benchmarking. *International Journal for Numerical Methods in Engineering*, 62(3):353–383.

de Souza Neto EA, Peric D, Dutko M & Owen DRJ. (1996). Design of simple low order finite elements for large strain analysis of nearly incompressible solids. *International Journal of Solids and Structures*, 33:3277–3296.

Devillers O, Pion S & Teillaud M. (2002). Walking in a triangulation. *International Journal of Foundations of Computer Science*, 13(02):181–199.

Griffiths DV & Lane PA. (1990). Finite element analysis of the shear vane test. Computers & Structures, 37(6):1105–1116.

Jassim I. (2013). Formulation of a Dynamic Material Point Method (MPM) for Geomechanical Problems. Ph.D Thesis, Institut fur Geotechnik der Universitat Stuttgart, Germany.

Stomakhin A, Schroeder C, Chai L, Teran J & Selle A. (2013). A material point method for snow simulation. *ACM Transactions on Graphics (TOG)*, 32(4):102:1–102:10.

Wang L, Coombs WM, Augarde CE, Brown M, Knappett J, Brennan A, Richards D & Blake A. (2017). Modelling screwpile installation using the MPM. *Procedia Engineering*, 175:124–132.

Wang L, Coombs WM, Augarde CE, Brown M, Knappett J, Brennan A, Davidson C, Richards D & Blake A. (2018). On the use of the material point method to model problems involving large rotational deformation, *Proceedings of the 9th european conference on numerical methods in geotechnical engineering--Numerical Methods in Geotechnical Engineering IX*, Porto, Portugal, vol. 1, pp. 585–592.