



A strategy for generating pushover curves of block assemblies including post-peak branch using the discrete element method

Igor Bouckaert – Institute of Mechanics, Materials and Civil Engineering, UCLouvain, Louvain-la-Neuve, Belgium, igor.bouckaert@uclouvain.be

Michele Godio – RI.SE Research Institutes of Sweden, Borås, Sweden, michele.godio@ri.se

João Pacheco de Almeida – Institute of Mechanics, Materials and Civil Engineering, UCLouvain, Louvain-la-Neuve, Belgium, joao.almeida@uclouvain.be

Abstract: Pushover analyses are often used to evaluate the seismic performance of a structure. They give an estimate of the ultimate displacement a structure can undergo, as well as of the residual resisting forces in the post-peak response. When modelling masonry structures composed of multiple blocks, obtaining the post-peak branch of the pushover curve can be difficult with a classic displacement-control strategy. This paper describes a strategy designed to compute this branch for multi-block systems subjected to a given pattern of forces, without the need to apply a displacement-control algorithm. The strategy is general, therefore straightforwardly implementable in different software tools and applicable to complex block assemblies. In the present work, it is implemented in two different DEM software, namely LMGC90 and UDEC, and tested on a benchmark problem for evaluating the in-plane response of masonry walls.

Keywords: masonry, URM, discrete elements, displacement-control, nonlinear response

1. Introduction

In the context of seismic design of buildings, pushover curves are a powerful tool to assess seismic performance (Krawlinker & Seneviratna, 1998). To simulate an earthquake loading on a structure composed of multiple blocks, a common strategy consists of applying on each block a horizontal force proportional to the weight of the block, simulating an equivalent static lateral force (Ferris & Tin-Loi, 2001). For the family of multi-block structures submitted to a pattern of forces, difficulties can arise when applying a simple displacement-control method to obtain the descending branch of the pushover curve, because it is not always possible to impose multiple forces at once while monitoring the value of one degree of freedom (Bocciarelli & Barbieri, 2017; Mordanova & de Felice, 2018).

This investigation describes an algorithm designed to overcome the above difficulty and obtain the force-displacement response of block assemblies subjected to a given pattern of forces, including the descending branch. The algorithm is implemented in two different discrete element method (DEM) software packages, namely LMGC90 and UDEC.

UDEC is one of the most popular DEM software tools used in the masonry community (Lemos, 2007). It is a proprietary software developed by Itasca, and it implements the Distinct Element Method, first presented by Cundall in the 1970s (Cundall, 1971). In this work, we use UDEC 6.0 (Itasca, 2013).

LMGC90 is an open-source DEM software tool implementing the non-smooth contact-dynamics (NSCD) method (Dubois et al., 2018; Jean, 1999). It was originally developed for spherical elements and granular flows but eventually adapted to polyhedral elements. Its relevance for modelling URM structures has been growing ever since (Taforel, 2012), with

numerous examples available in the literature (Isfeld & Shrive, 2015; Rafiee et al., 2008). In this work, we use LMGC90 2021.rc1 (Mozul, 2021).

LMGC90 has been investigated as an alternative to UDEC for modelling masonry structures through different benchmark problems, namely for assessing the vulnerability of ancient masonry towers (Ferrante et al., 2021), for simulating the rocking behaviour of columns and arches under dynamic loads, and for studying collapse mechanisms and deriving the in-plane force capacity of masonry panels (Bouckaert et al., 2021).

This paper continues the investigation started with Bouckaert et al. (2021). By making use of the present algorithm, the analysis of the collapse mechanisms and force capacity of masonry panels is extended here to the computation of their full force-displacement response for the same panels. The pushover curves obtained with the present algorithm implemented in UDEC and LMGC90 are compared.

2. Algorithm description

The present algorithm enables the user to run a pushover analysis in a dynamic numerical time-domain integration setting, in which a given pattern of forces is applied and controlled by a load multiplier, while monitoring the value of a given degree of freedom. This method differs from a classical displacement-control method by the fact that it does not require to solve a constraint equation to determine the load increment applied to the force pattern (Anthoine, 2006). In what follows the algorithm is described in detail. The flowchart of the algorithm is depicted in Fig. 1.

We denote by $\mathbf{F}$ the load pattern $[F_1, \dots, F_n]$, where F_i represents a horizontal force applied on the centre of gravity of block i . The sum of all the applied forces, which is also horizontal, is denoted by $\Sigma \mathbf{F}$. During the simulation, the load multiplier λ_b controls the intensity of the applied forces: in particular, at the time step i , the structure is subjected to the set of forces $\lambda_b \mathbf{F}$. When the simulation starts $\lambda_b^0 = 0$. At each time step i , the parameter λ_x^i is obtained as the sum of the horizontal reactions at the supports at the end of the current time step i divided by $\Sigma \mathbf{F}$.

At the end of each time step, the algorithm checks if $\Sigma \mathbf{F}$ is equal to the reaction sum at the support, i.e. if $\lambda_x^i = \lambda_b^i$, which corresponds to a structure in static equilibrium. If this is the case, the load multiplier is increased by a user-defined load increment $\Delta \lambda$ at the next step, i.e. $\lambda_b^{i+1} = (i+1) \cdot \Delta \lambda$, which corresponds to the determination of the ascending branch of the pushover curve. Note that the load multiplier increment $\Delta \lambda$ should be small enough to stay in quasi-static conditions during the whole simulation.

When reaching the peak of the pushover curve, increasing further the value of λ_b leads to applied forces that the structure can no longer sustain, therefore that $\lambda_x^i < \lambda_b^i$. This means that a portion of the applied forces is converted into inertia forces, causing some of the blocks to accelerate and entering the post-peak branch of the pushover curve. In this case, and since the proposed algorithm also intends to determine the subsequent post-peak branch of the pushover curve, the following modifications are introduced: the applied load multiplier at the next time step is set equal to the portion of applied forces that are effectively in equilibrium with the reaction forces (i.e. $\lambda_b^{i+1} = \lambda_x^i$).

To avoid accumulation of unbalanced applied forces causing excessive acceleration of some of the blocks composing the structure, the velocity of each of the blocks is set to zero at each time step when the algorithm enters the descending branch. The pushover analysis is carried

out until the value of the load multiplier, after reaching its peak, starts reducing to a value smaller than 0.01, as indicated in Fig. 1. Below this value, partial collapse of the structure typically results in blocks falling on the support, leading to values of λ_x that no longer verify static equilibrium.

Once the algorithm enters the simulation of the descending branch, thus imposing $\lambda_b^{i+1} = \lambda_x^i$ at each time step, the forces applied at time step $i+1$ can be insufficient to increase further the value of the controlled displacement, probably due to the velocity-reset at each time step. In other words, instead of going past the peak of the pushover curve, the system tends to unload and revert to its original, undeformed state. This issue appears to be consistently solved by increasing by 5% the value of λ_b^{i+1} compared to the value of λ_x^i , i.e. to apply $\lambda_b^{i+1} = 1.05 \cdot \lambda_x^i$. We notice that λ_b^{i+1} is a decreasing quantity as the time steps progress, because λ_x^i reduces in the post-peak branch. This adjustment enables the algorithm to get past the peak of the pushover curve, without causing the structure to collapse abruptly. Finally, the pushover curve is obtained by plotting the value of the displacement calculated at the end of each time step versus the value of the load multiplier λ_b that was applied during that step.

Previous works have presented results obtained by applying the present strategy to masonry walls in UDEC (Godio & Beyer, 2019; Portioli et al., 2021). In this study, the strategy is implemented and tested in LMGC90 and the results from the two tools are compared. A similar strategy to obtain the full force-displacement response of masonry structures using DEM has been described in (Gobbin et al., 2021). Gobbin et al. (2021) include a method to compute the mass participating to the collapse mechanism, which represents a possible extension of the present strategy.

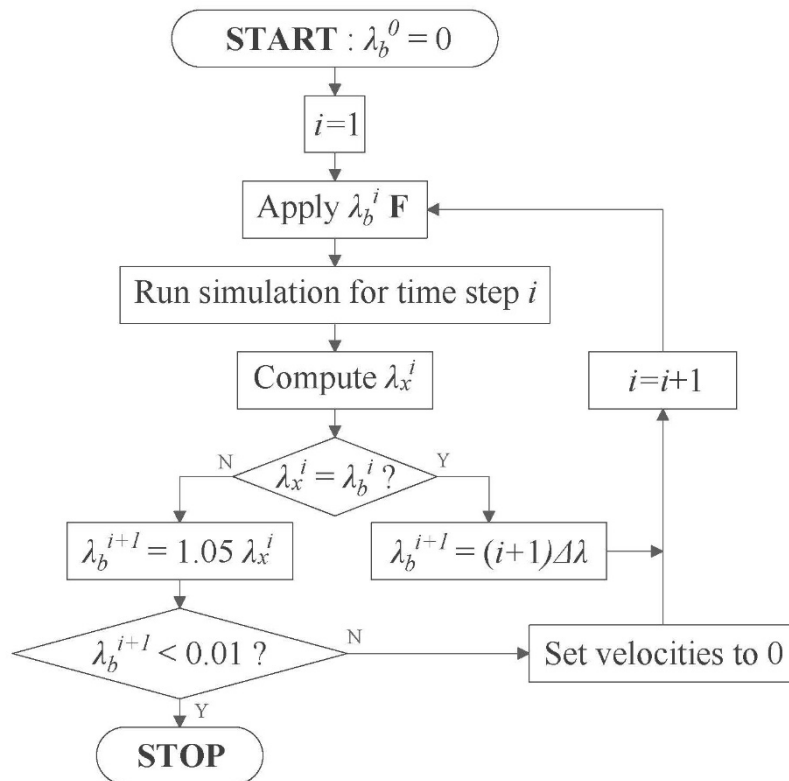


Fig. 1 – Flowchart representation of the algorithm for generating pushover curves of block assemblies using the DEM method

3. Application example

The algorithm is applied on three dry-joint masonry panels with different geometries, one of which includes an opening (Fig. 2). The masonry units are 0.4 m wide and 0.175 m high and the edges are modelled with a rounding radius of 0.02 m, to simulate imperfections at the corners. The specific weight of the blocks is 10 kN/m³. These benchmark problems were first introduced in (Baggio & Trovalusci, 1998; Ferris & Tin-Loi, 2001). Both papers presented a limit analysis method used to compute the load multiplier leading the panels to collapse. The values of the collapse load multiplier computed by Ferris and Tin-Loi (2001) were later corrected by Gilbert et al. (2006). The collapse load multiplier is also computed with the present algorithm and corresponds here to the peak of the pushover curve. Besides computing the collapse load multiplier, the algorithm also computes the full force-displacement response of the wall up to near-zero residual resisting force.

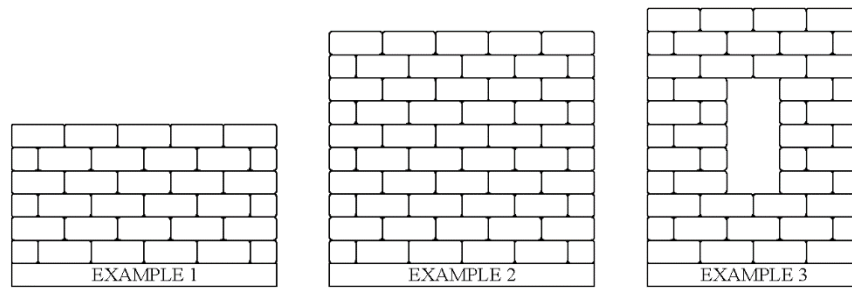


Fig. 2 – Wall geometries

This benchmark problem is implemented in UDEC and LMGC90. In both tools, the contact between bricks is modelled as a no-tension, non-associated Coulomb friction contact law, with friction coefficient $\mu = 0.65$. The blocks are modelled as rigid bodies, and the deformability of the masonry is concentrated at the interfaces between blocks by the introduction of a contact stiffness parameter. The definition of this parameter differs between UDEC and LMGC90. In UDEC, for modelling elastic behaviour of dry-joint masonry under compression, the normal stiffness is set to $k_n = 10^8$ Pa/m to compare with Portioli et al. (2021). The normal stiffness parameter K_n in LMGC90 is expressed in N/m, obtained by multiplying the value of k_n introduced in UDEC by the surface associated to each contact, namely 0.1 m² for the bed joints and 0.0875 m² for the head joints (head and bed joints have respectively 2 and 4 contact points in UDEC and LMGC90). Default input parameters in UDEC are also the dilatancy angle ψ and the shear stiffness of the joints k_s . These parameters do not have an equivalent in the contact law used in LMGC90. Therefore, the values of the dilatancy angle is set to 0° and the shear stiffness is set to a value 1000 times higher than the normal stiffness ($k_s = 10^{11}$ Pa/m) to use comparable contact laws between both software tools. Another difference between UDEC and LMGC relies on how often contacts are detected. UDEC does not perform contact detection at every time step, but triggers detection of new contacts and deletion of old ones only when relative displacements between contact points exceed a threshold value within two consecutive time steps (Itasca, 2014). On the contrary, LMGC90 performs the contact detection at each time step.

During the simulation, each block is subjected to its self-weight W and to a horizontal force pointing to the right, made proportional to W by the load multiplier λ_b described in the previous chapter. During the simulation, the monitored displacement is the horizontal displacement of the upper right corner of the upper right block, in each wall. This displacement is denoted as δ_c .

Fig. 3, Fig. 4 and Fig. 5 show the full pushover curve of each of the three walls obtained with UDEC and LMGC90, along with the collapse mechanism at $\delta_c = 0.2\text{m}$. The collapse mechanism obtained with LMGC90 is plotted with the software ParaView (Ahrens et al., 2005). Table 1 indicates the values of the maximal load multiplier λ_b^{max} from Gilbert et al. (2006) and computed with LMGC90 and UDEC, as well as the maximal displacement δ_c^{max} . The indicated error compares the value of the load multiplier obtained with UDEC and with LMGC90.

Table 1. Maximal load multiplier λ_b^{max} and displacement δ_c^{max} at $\lambda_b = 0.01$ for the three walls

	λ_b^{max} (Gilbert et al., 2006)	λ_b^{max} (LMGC90)	λ_b^{max} (UDEC)	Error (%)	δ_c^{max} (LMGC90)
Wall 1	0.643	0.579	0.574	+0.86%	0.449m
Wall 2	0.564	0.512	0.497	+2.93%	0.556m
Wall 3	0.356	0.331	0.323	+2.42%	0.591m

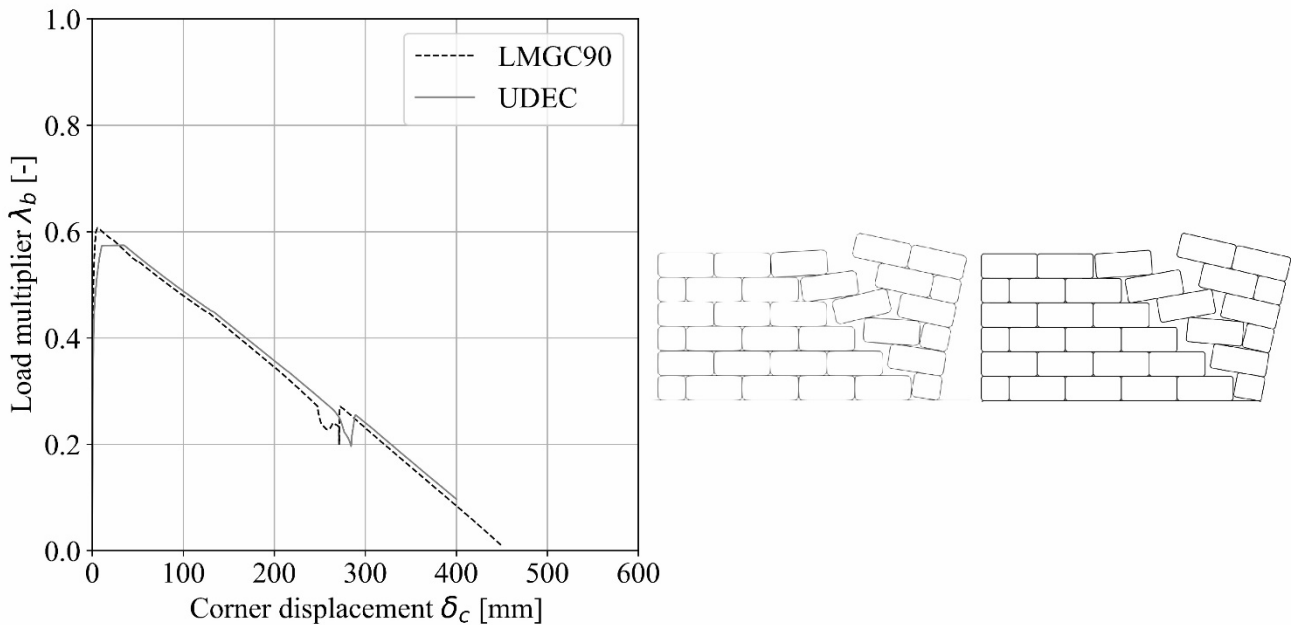


Fig. 3 – Wall 1: Pushover curves (left) and collapse mechanism at $\delta_c = 0.2\text{m}$ from LMGC90 (left) and UDEC (right)

5. Discussion and conclusion

Good agreement is observed between the results obtained with UDEC and LMGC90, both in terms of maximal load multiplier (maximal difference of 2.93% for the second wall), and overall shape of the force-displacement response. The load multiplier appears to be up to 10 % lower than the one found in Gilbert et al. (2006) by limit analysis, due to the rounding of the corners and the introduction of joint stiffness, which affect the elasticity of the system.

With the presented numerical strategy, the full force-displacement response of three different block assemblies has been computed with small differences in the response obtained with both software tools. Since a different DEM formulation is adopted in LMGC90 and UDEC, the position of local peaks in the pushover curves can differ. These local peaks are due to a sudden detachment of one or multiple blocks, which is not detected at the exact same time instant, as a consequence of the difference in the contact detection algorithm mentioned in the previous section.

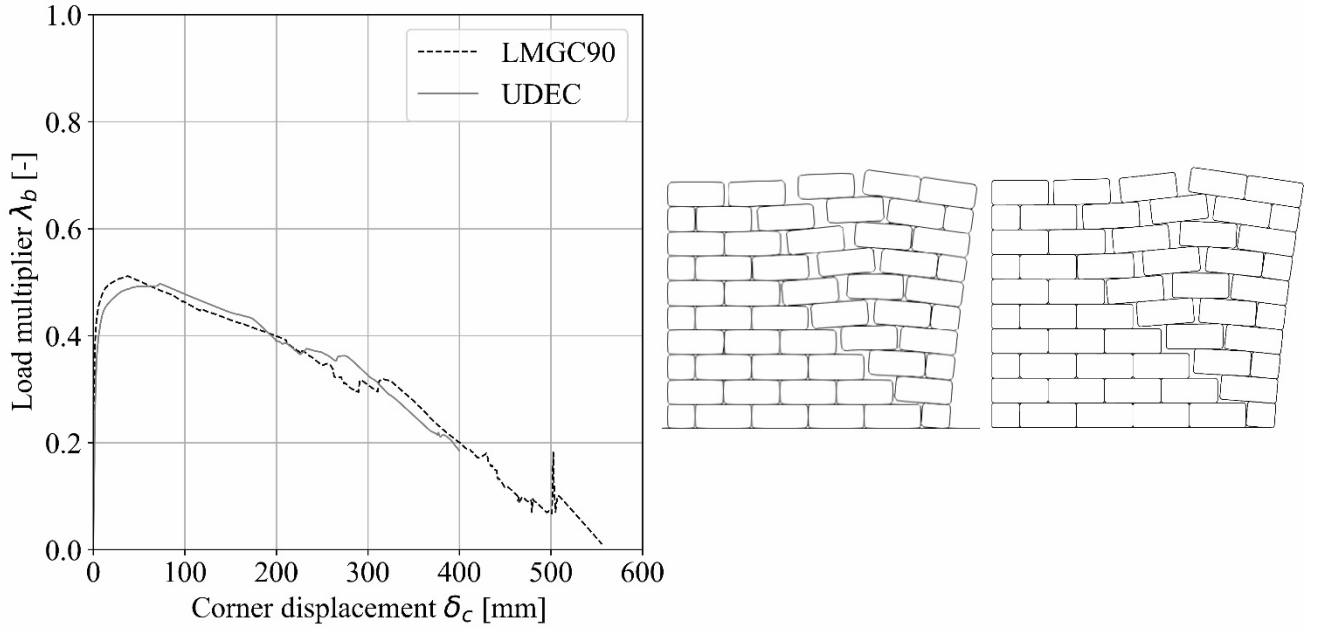


Fig. 4 – Wall 2: Pushover curves (left) and collapse mechanism at $\delta_c = 0.2m$ from LMGC90 (left) and UDEC (right)

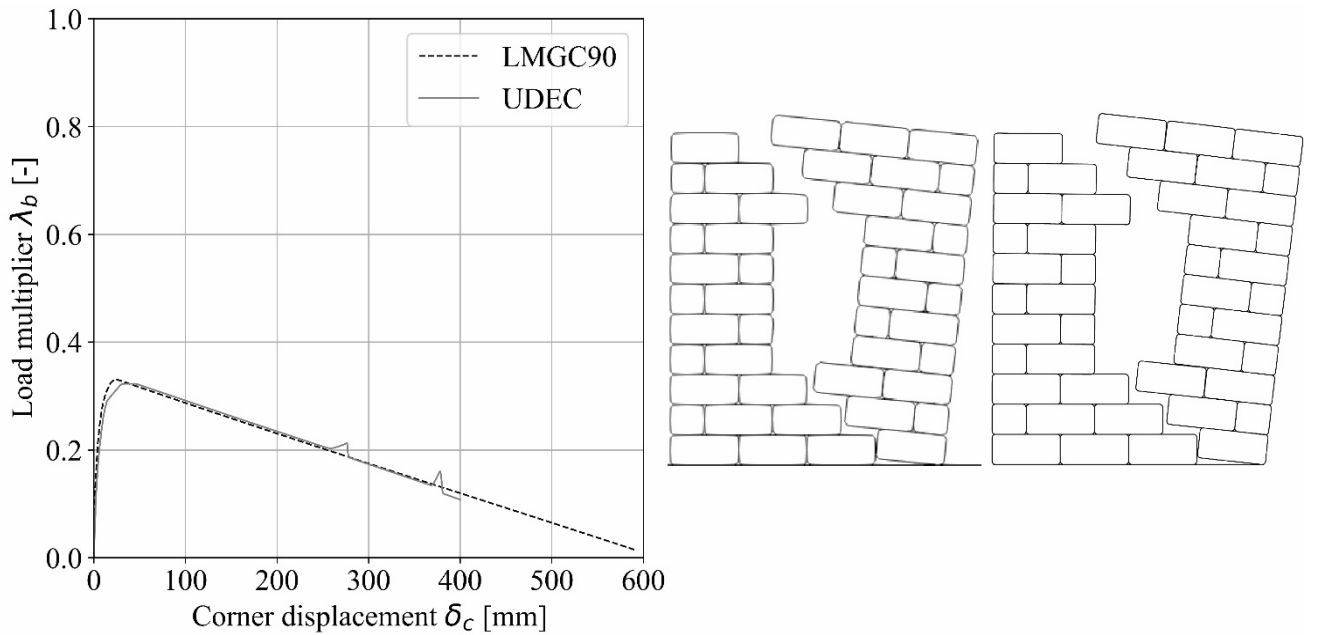


Fig. 5 – Wall 3: Pushover curves (left) and collapse mechanism at $\delta_c = 0.2m$ from LMGC90 (left) and UDEC (right)

Moreover, the results show that walls with a clear collapse mechanism (such as wall 3 or wall 1, where one single macroblock detaches from the rest of the structure) display a rather linear post-peak behavior that corresponds to the progressive tilting of the detaching block. On the contrary, walls for which the collapse mechanism displays multiple cracking lines (such as wall 2) display a less smooth post-peak behavior, with local peaks attributed to the sudden detachment of one block.

The correspondence between the results of both software is encouraging regarding the use of LMGC90 as an open-source alternative for practitioners. Through its Python-interface, LMGC90 offers the possibility to implement algorithms that are more complex than a classical force- or displacement-control strategy. The possibility to extract results at any time

step during the simulation and to use these results in the next time step, one of the attractive features of UDEC, is also feasible in LMGC90 and offers significant flexibility to the user. After the relevance of LMGC90 had been explored for running time-history analyses, computing collapse load multipliers and identifying collapse mechanisms (Bouckaert et al., 2021), the present paper contributes to increase confidence in the use of LMGC90 for running pushover analyses on masonry structures.

However, it is important to note that the computational time needed to run a quasi-static analysis in LMGC90 is large compared to UDEC when the time step is not chosen carefully. The generation of the full pushover curve in LMGC90 can take up to one or two days on a personal computer with reasonable computational power, versus a few hours in UDEC. Adaptations of the applied algorithm to overcome this problem will be investigated in future works. The algorithm can also be implemented in structural analysis software other than the two considered here.

Finally, the numerical strategy described in this paper can be applied to more complex structures than those herein presented. Its extension to 3D structures is a necessary step towards the application of the pushover curves for the seismic assessment of existing buildings.

Acknowledgements

Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Equipements de Calcul Intensif en Fédération Wallonie Bruxelles (CECI) funded by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region. The first author is thankful for the financial support given by UCLouvain.

References

- Ahrens, J., Geveci, B., & Law, C. (2005). *ParaView: An End-User Tool for Large Data Visualization*.
- Anthoine, A. (2006). A simple displacement control technique for pushover analyses. *Earthquake Engineering & Structural Dynamics*, 35(7), 851-866. <https://doi.org/10.1002/eqe.560>
- Baggio, C., & Trovalusci, P. (1998). Limit Analysis for No-Tension and Frictional Three-Dimensional Discrete Systems*. *Mechanics of Structures and Machines*, 26(3), 287-304. <https://doi.org/10.1080/08905459708945496>
- Bocciarelli, M., & Barbieri, G. (2017). A numerical procedure for the pushover analysis of masonry towers. *Soil Dynamics and Earthquake Engineering*, 93, 162-171. <https://doi.org/10.1016/j.soildyn.2016.07.022>
- Bouckaert, I., Godio, M., & Pacheco de Almeida, J. (2021). Large-displacement response of unreinforced masonry structures: comparison between analytical solutions and DEM models including open-source software. COMPDYN - 8th ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering, Athen, Greece.
- Cundall, P. A. (1971). A Computer Model for Simulating Progressive Large Scale Movements in Blocky Rock Systems. Symposium of the International Society for Rock Mechanics,
- Dubois, F., Acary, V., & Jean, M. (2018). The Contact Dynamics method: A nonsmooth story. *Comptes Rendus Mécanique*, 346(3), 247-262. <https://doi.org/10.1016/j.crme.2017.12.009>

- Ferrante, A., Loverdos, D., Clementi, F., Milani, G., Formisano, A., Lenci, S., & Sarhosis, V. (2021). Discontinuous approaches for nonlinear dynamic analyses of an ancient masonry tower. *Engineering Structures*, 230. <https://doi.org/10.1016/j.engstruct.2020.111626>
- Ferris, M. C., & Tin-Loi, F. (2001). Limit analysis of frictional block assemblies as a mathematical program with complementarity constraints. *International Journal of Mechanical Sciences*, 43(1), 209-224. [https://doi.org/10.1016/s0020-7403\(99\)00111-3](https://doi.org/10.1016/s0020-7403(99)00111-3)
- Gilbert, M., Casapulla, C., & Ahmed, H. M. (2006). Limit analysis of masonry block structures with non-associative frictional joints using linear programming. *Computers & Structures*, 84(13-14), 873-887. <https://doi.org/10.1016/j.compstruc.2006.02.005>
- Gobbin, F., de Felice, G., & Lemos, J. V. (2021). Numerical procedures for the analysis of collapse mechanisms of masonry structures using discrete element modelling. *Engineering Structures*, 246. <https://doi.org/10.1016/j.engstruct.2021.113047>
- Godio, M., & Beyer, K. (2019). Evaluation of force-based and displacement-based out-of-plane seismic assessment methods for unreinforced masonry walls through refined model simulations. *Earthquake Engineering & Structural Dynamics*, 48(4), 454-475. <https://doi.org/10.1002/eqe.3144>
- Isfeld, A., & Shrive, N. (2015). Discrete Element Modeling of Stone Masonry Walls With Varying Core Conditions: Prince of Wales Fort Case Study. *International Journal of Architectural Heritage*, 9(5), 564-580. <https://doi.org/10.1080/15583058.2013.819135>
- Itasca. (2014). *UDEC 6.0 User's Guide*. Itasca.
- Jean, M. (1999). The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177(3-4), 235-257. [https://doi.org/10.1016/s0045-7825\(98\)00383-1](https://doi.org/10.1016/s0045-7825(98)00383-1)
- Krawlinker, H., & Seneviratna, G. (1998). Pros and cons of a pushover analysis of seismic performance evaluation. *Engineering Structures*, 20, 452-464.
- Lemos, J. V. (2007). Discrete Elements Modeling of Masonry Structures. *International Journal of Architectural Heritage*.
- Mordanova, A., & de Felice, G. (2018). Seismic Assessment of Archaeological Heritage Using Discrete Element Method. *International Journal of Architectural Heritage*, 14(3), 345-357. <https://doi.org/10.1080/15583058.2018.1543482>
- Mozul, R. (2021). *LMGC90 project : user version*. https://git-xen.lmgc.univ-montp2.fr/lmgc90/lmgc90_user
- Portioli, F. P. A., Godio, M., Calderini, C., & Lourenço, P. B. (2021). A variational rigid - block modeling approach to nonlinear elastic and kinematic analysis of failure mechanisms in historic masonry structures subjected to lateral loads. *Earthquake Engineering & Structural Dynamics*, 50(12), 3332-3354. <https://doi.org/10.1002/eqe.3512>
- Rafiee, A., Vinches, M., & Bohatier, C. (2008). Modelling and analysis of the Nîmes arena and the Arles aqueduct subjected to a seismic loading, using the Non-Smooth Contact Dynamics method. *Engineering Structures*, 30(12), 3457-3467. <https://doi.org/https://doi.org/10.1016/j.engstruct.2008.05.018>
- Taforel, P. (2012). *Apport de la Méthode des Eléments Discrets à la Modélisation des Maçonneries en Contexte Sismique : Vers une Nouvelle Approche Numérique de la Vulnérabilité Sismique* Université Montpellier 2]. Montpellier.