



HAL
open science

Fast generative tool for masonry structures geometries

Nathanaël Savalle, Elham Mousavian, Carla Colombo, Paulo B Lourenco

► **To cite this version:**

Nathanaël Savalle, Elham Mousavian, Carla Colombo, Paulo B Lourenco. Fast generative tool for masonry structures geometries. Canadian Masonry Symposium, May 2021, Montréal (colloque en ligne), Canada. hal-04481922

HAL Id: hal-04481922

<https://hal.science/hal-04481922v1>

Submitted on 28 Feb 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.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/351783939>

Fast generative tool for masonry structures geometries

Conference Paper · May 2021

CITATIONS

4

READS

567

4 authors:



Nathanaël Savalle

Université Clermont Auvergne

25 PUBLICATIONS 119 CITATIONS

SEE PROFILE



Elham Mousavian

The University of Edinburgh

32 PUBLICATIONS 185 CITATIONS

SEE PROFILE



Carla Colombo

University of Minho

10 PUBLICATIONS 26 CITATIONS

SEE PROFILE



Paulo B. Lourenco

University of Minho

1,276 PUBLICATIONS 27,948 CITATIONS

SEE PROFILE



14TH CANADIAN MASONRY SYMPOSIUM
MONTREAL, CANADA
MAY 16TH – MAY 19TH, 2021



FAST GENERATIVE TOOL FOR MASONRY STRUCTURES GEOMETRIES

Savalle, Nathanaël¹; Mousavian, Elham²; Colombo, Carla³ and Lourenço, Paulo B.⁴

ABSTRACT

Modelling masonry bond pattern is still challenging for the scientific community. Though advanced Laser Scanning methods are available and allow to extract blocks sizes and shapes of actual masonry structures, they are up to now very time-consuming and complex to set up. Therefore, modelling masonry as an ideal and regular assemblage of regular units is still very common in the scientific field. This paper presents a generative algorithm for masonry specimens built with a single-leaf bond pattern. It is based on C# programming under the environment offered by Rhinoceros (+ Grasshopper). Five components have been constructed (wall, corner, T and cross-connections, and opening). They can be assembled, up to infinity, to build complex masonry specimens. Moreover, they are all parametrised to account for every wish of the modeller. The global methodology is found highly time-efficient, with the creation of an initial geometry composed of 5 – 10 components requiring around 10 minutes and, while the update due to a parameter variation is done in less than one second. The paper finally discusses the next developments of the promising generative algorithm.

KEYWORDS: *Masonry, Parametric modelling, C# coding, single-leaf, Grasshopper*

¹ PhD, University of Minho, ISEC, Department of Civil Engineering, Guimarães, Portugal, n.savalle@civil.uminho.pt

² PhD, Department of Structures for Engineering and Architecture, University of Naples Federico II, Italy, elham.mousavian@unina.it

³ PhD Student, University of Minho, ISEC, Department of Civil Engineering, Guimarães, Portugal, carla.colombo95@gmail.com

⁴ Full Professor, University of Minho, ISEC, Department of Civil Engineering, Guimarães, Portugal, pbl@civil.uminho.pt

INTRODUCTION

Masonry is one of the most used building material in the world. Masonry buildings also represent a significant part of our cultural heritage, being the physical memory of our civilisation's history. Because they are often old, these historical structures are often highly damaged (due to, e.g. foundation settlements [1], past earthquakes [2-4], insufficient initial design [5-6], or weathering). They need today maintenance operations. Their assessment and the identification of the maintenance or strengthening actions are carried out using either limit analysis [7-10] or numerical [1, 10-16] tools. Both categories can then be divided into micro-modelling [8-9, 11-12] or macro-modelling approaches [1, 7, 10, 13-16]. In micro-modelling approaches, each block is modelled separately and interact one with another at an interface. In macro-modelling approaches, masonry is represented by an homogeneous continuum with mechanical properties that depend on the one of the the masonry units and of the mortar. Both of them need to define a geometrical parameter: the size of the masonry units. This directly dictates the size of the blocks in micro-modelling [8-9, 11-12], the failure criteria in limit analysis macro approaches [7, 10] or lastly, the parameter of an equivalent homogenised masonry material [1, 13-16]. Indeed, blocks size has been found to influence the final load capacity of a masonry structure subjected to a foundation settlement [8].

However, given the variety of stones and methods to assemble them, no universal and unique description of the bond pattern is possible. In practice, it leads to very irregular masonry patterns, even when using rectangular blocks (Figure 1). Finally, when using regular and identical bricks as masonry units, one can still observe various bond patterns worldwide (Figure 2).



Figure 1: Masonry pattern using regular blocks for the historic perimeter wall of Guimarães, Portugal

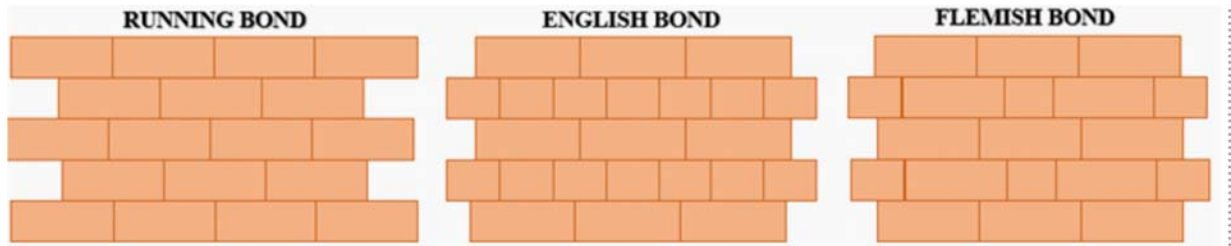


Figure 2: Different typical bond patterns

Some in-field studies have been developed to acquire the actual sizes and shapes of masonry units [16-19]. They are mainly based on Terrestrial or Mobile Laser Scanning technologies that create a 3D data Point Cloud of the monitored object. Though they all showed the high potential of Laser Scanning approaches, they also demonstrated that the extraction of the block sizes and shapes out of the created Point Cloud is not an easy task [17-18]. For instance, Valero et al. developed an automatic plugin for this extraction, but that only handles “straight” walls (i.e. without significant curvatures).

For this reason, masonry structures have been (and still today) often modelled as an assemblage of regular blocks, either from a computational point of view [8-9, 11] or from an experimental point of view [11-12, 20-22]. In this respect, it is evident that a micro-modelling strategy [8-9, 11] may become significantly time-consuming during the preparation of the model geometry because of the number of units and the definition of all possible interacting interfaces [8-9]. Computer-Aided-Design (CAD) can be of great help in this laborious task [23]. One of the noteworthy efforts in this regard is BIM-M (Building Information Modeling for Masonry) [24]. That initiative more specifically focuses on developing the construction-oriented data structure. Yet, providing the database collecting the structural aspects of the masonry assemblage is missing in the literature.

To tackle these challenges and improve the CAD already existing solutions, this work aims to present an innovative tool that generates masonry geometries in a very time-efficient way. Its main advantage lies in the parametrisation of all geometrical data, allowing geometrical updates very quickly.

GENERAL DESCRIPTION AND PURPOSES OF THE TOOL

The generative algorithm is implemented in the environment offered by Rhinoceros3D-Grasshopper to use the programming language C#. Its purposes can be gathered as follows:

1. Allowing the geometrical modelling of masonry structures, from simple shapes (e.g. U-shaped, Corners [21]) to more complex ones. The time required to build these models should be proportional to the complexity of the structure and comparable to other CAD software [23].
2. Covering all possible configurations and let the user choose his/her design. Hence, several geometrical parameters must be defined.
3. Allowing a fast update of the geometry when some parameters vary.

4. Enabling the easy exportation of the generated geometries to structural software for analysis (e.g., DIANA, Abaqus, 3DEC, LiABlock_3D, etc.).

The present tool allows modelling geometries with single-leaf masonry bond patterns. The first three objectives are already fulfilled for single-leaf structures. More specifically, the time requested to build a classical U-shaped masonry specimen is approximately ten minutes, while it updates the model in less than one second for each parameter variation. The following sections describe, in a more detailed way, the present generative algorithm.

PARAMETRISATION CHOSEN TO CREATE MASONRY GEOMETRIES

For simplicity, the generative algorithm uses five components corresponding to five different masonry elements. The first one corresponds to a masonry wall. The second to fourth ones correspond to masonry connections (corner, T-connection and cross-connection, respectively) that assemble two (or more) walls. The last component has been developed to create an opening in an already existing wall.

One should note that using these different typologies allows creating almost all possible masonry configurations. However, curved walls are not yet included, though the same methodology can be easily applied without significant work.

Wall component

The wall component is the main component of the generative algorithm. Figure 3 gives the perspective and plane views of it. One can note that the single-leaf bond pattern only considers two distinct courses (even and odd) that are repeated all along with the component's height. This characteristic is shared by all features described in the following.

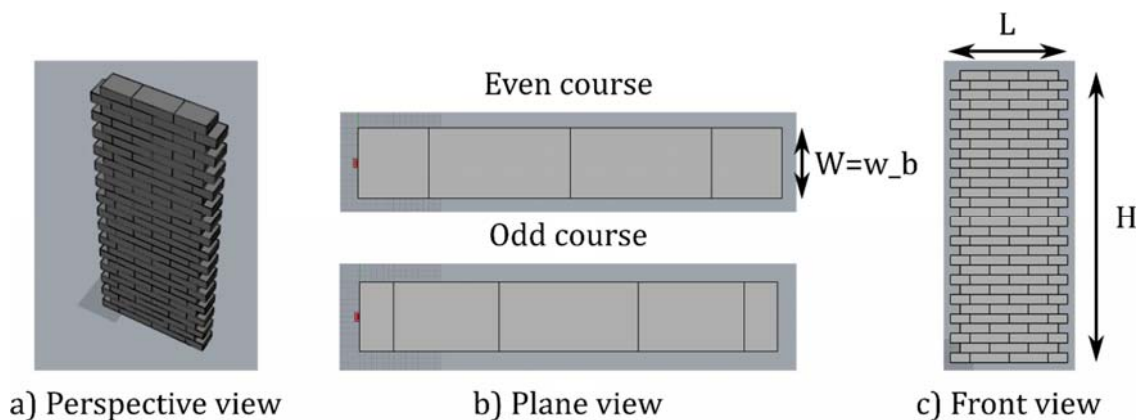


Figure 3: Wall component. a) Perspective, b) plane, and c) front views.

In total, a wall component has 19 parameters (Figure 4):

- Three geometrical parameters describe the masonry units: length l_b , width w_b and height h_b (Figures 3 & 5).
- Two geometrical parameters define the wall: its length L and height H (Figure 3).

- Two Boolean parameters ask if the length and height of the blocks need to be recomputed based on the wall length and height, respectively to avoid very small or thin blocks. If activated, they ensure that $L = n \times l_b$ and $H = n \times h_b$ with n to be an integer.
- A Boolean parameter asks whether the first course of the wall is an odd or an even type (Figure 3).
- For each side, three parameters handle the connection properties. Specifically, the first one is a Boolean asking if the side should be straight or indentated (Figure 6). If the side is indentated (Figure 6b), the second parameter determines the length of the indent l_1 and l_2 , respectively (Figure 5). The last parameter is also a Boolean specifying if the indentation is done on the even or odd courses.
- The horizontal distance between heading joints of even and odd layer is given by e (Figure 5).
- The minimum length that blocks have to be indeed created (Figure 6c).
- Three parameters that determine the coordinates of the local origin O of the wall, the orientation of its longitudinal dimension (V^1) and the orientation of its elevation dimension (V^2), see Figure 6a.

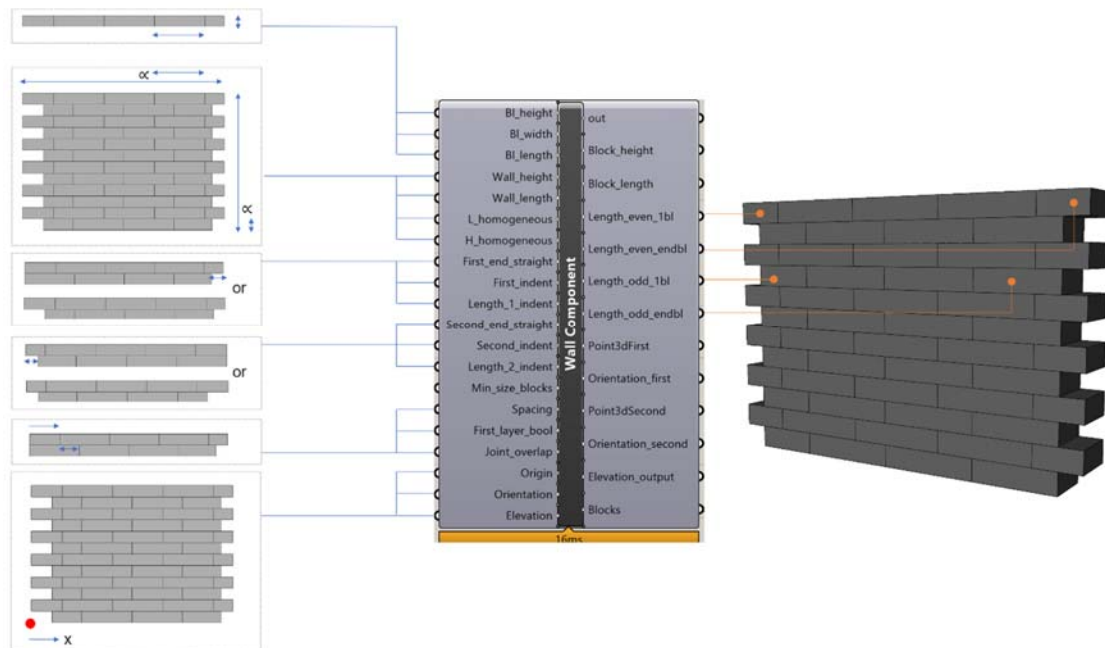


Figure 4: An overview of the GH component to model a wall with its inputs and outputs.

As for the output, all automatically generated blocks of a wall are stored in a DataTree (Rhinceros3D data structure), where each branch of the tree corresponds to a layer. It allows a much easier utilisation of the output later on.

The position O and orientation vectors of both side of the wall component is also outputted. Since all the other features use the same methodology to define their positions, it allows directly connecting the output position parameters of a given component to the input position parameters of another component to be connected with (Figure 7).

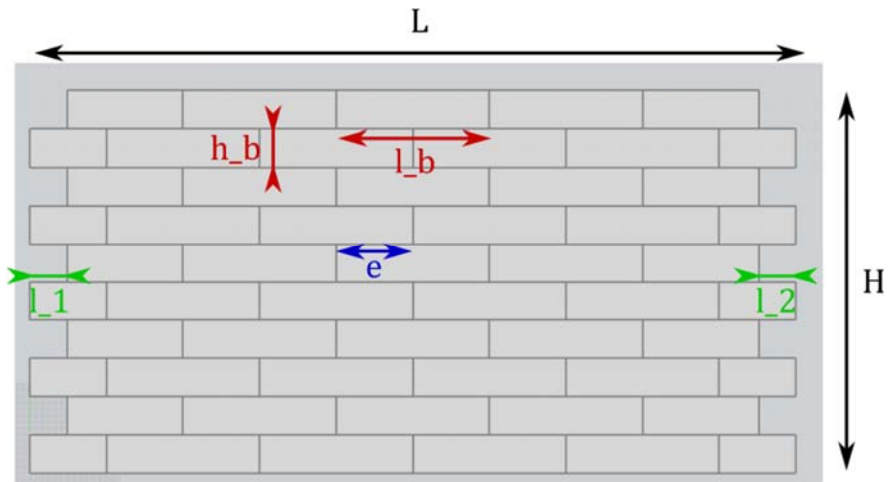


Figure 5: Parametrisation used for the wall component

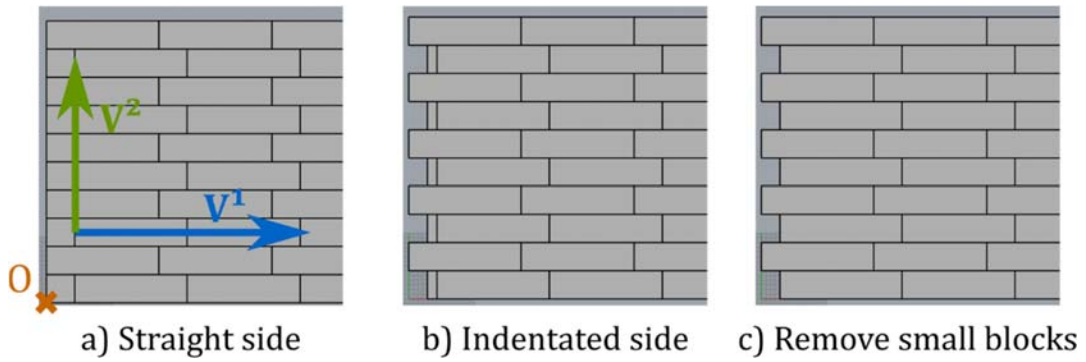


Figure 6: a) Straight and b) indentated connections. c) Effect of the minimum allowable block length

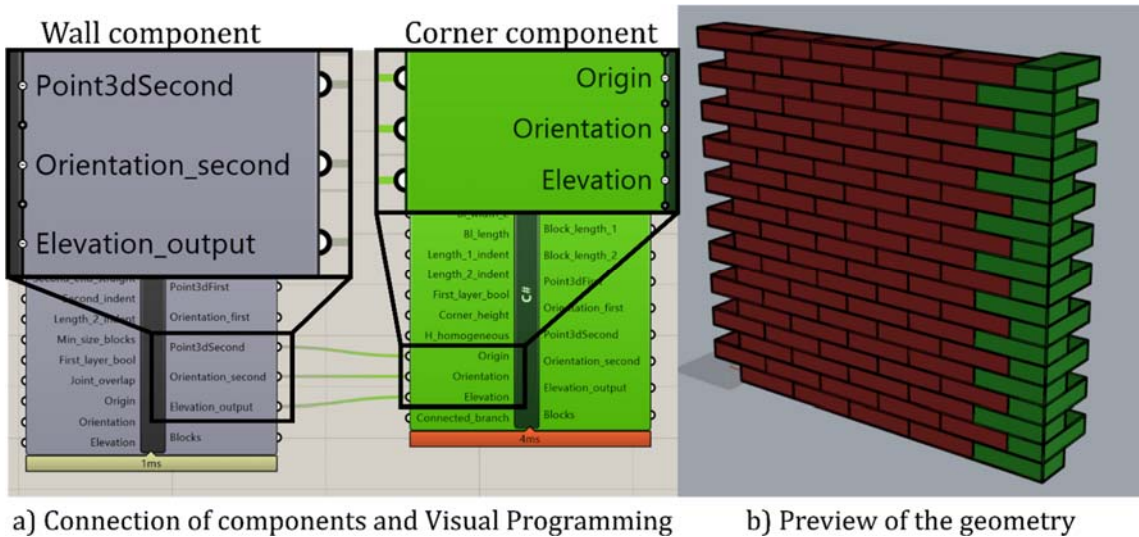


Figure 7: Connection of components to create the global geometry. a) Visual programming and b) preview

Corner component

A masonry corner component binds two different walls together connected orthogonally. It has 13 inputs (Figure 7), and most of them are similar to the inputs of a wall component. One of its particularities lies in the fact that the width of the two connected walls can be different (Figure 8). Another particularity is the possibility to connect either the 1st branch or the 2nd one, resulting in a “clockwise” or “counter-clockwise” connection (Figure 8).

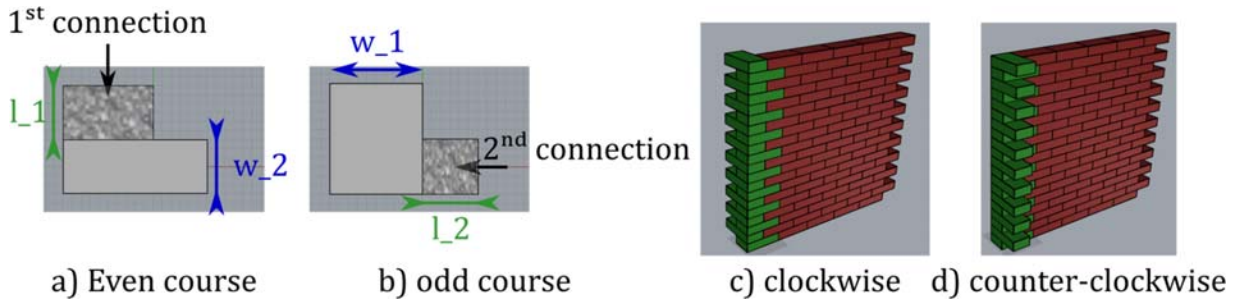


Figure 8: Corner component: a) and b) plane view and parametrisation; c) and d) previews with clockwise and counter-clockwise connections

T-connection and Cross-connection components

T-connection components connect three walls and have a total of 17 inputs. Most of them are shared with the previous features (wall and corner connection). Again, the width of the three branches to be connected can be different (Figure 9).

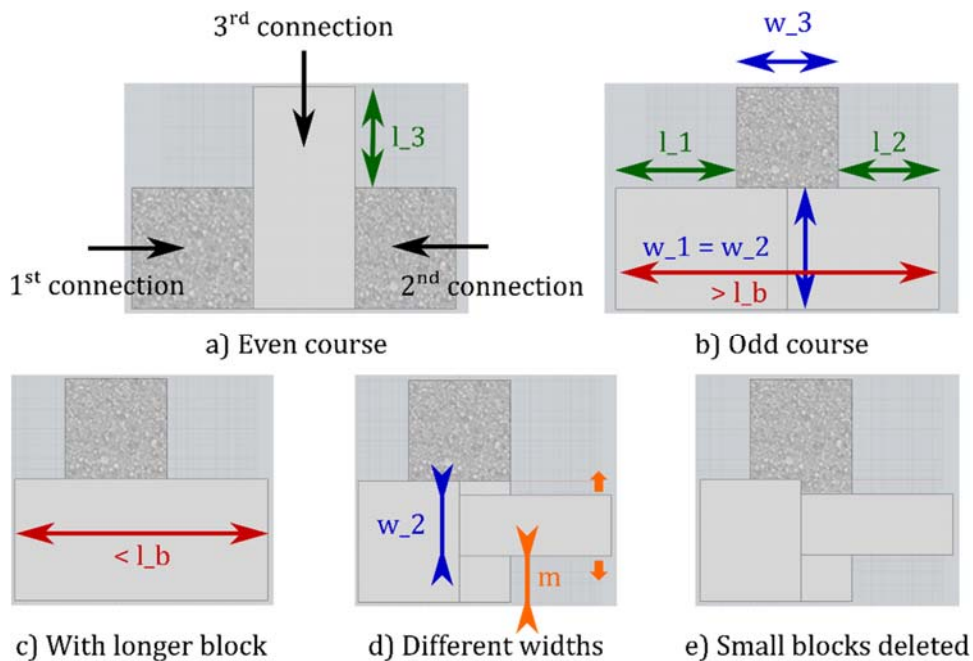


Figure 9: Parametrisation for a T-connection: a) even and b) odd layers; c) larger block length; d) misalignment of wall's façade and e) deleting of small blocks

Two specific features of T-connections are also presented on Figure 9. First, blocks can be merged if long enough blocks are available (Figure 9b-c). Moreover, in case the widths of the façade's walls are different, the user can determine the misalignment m (Figure 9d) so that the walls can either be aligned along their external ($m = 0\%$) or internal faces ($m = 100\%$). Finally, the last input parameter corresponds to the minimum length of blocks and avoid creating too small blocks (Figure 9d-e).

The cross-connection component has 19 inputs and works similarly to the T-connection features. The refinements discussed in Figure 9 apply in both x and y-directions, thus leading to more input parameters. Examples of T-wall and Cross-wall are shown in Figure 10.

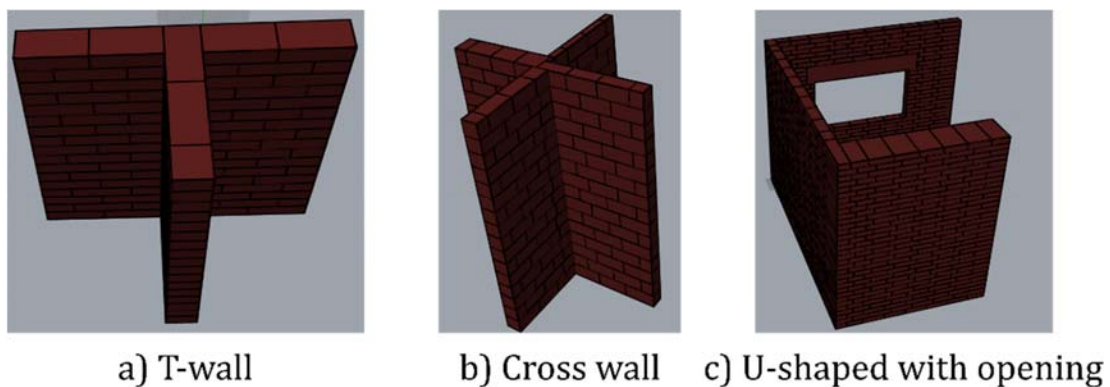


Figure 10: Final model geometries created using the five essential components.

Opening

The last feature regards the creation of an opening in an existing wall. The component has 23 input parameters, of which three corresponds to the wall DataTree (and eventually of two connections components) to cut. Then, six parameters define the relative and absolute position of the opening, while three other dimensions (l^{op} and h^{op}) control its size (Figure 11). Six parameters corresponding to the block and wall geometries are also needed. Finally, one Boolean parameter determines if a lintel should be created above the opening, and if yes, the lintel height (h^l) and width (w^l) as well the length of the supporting columns (b^l) must be assigned.

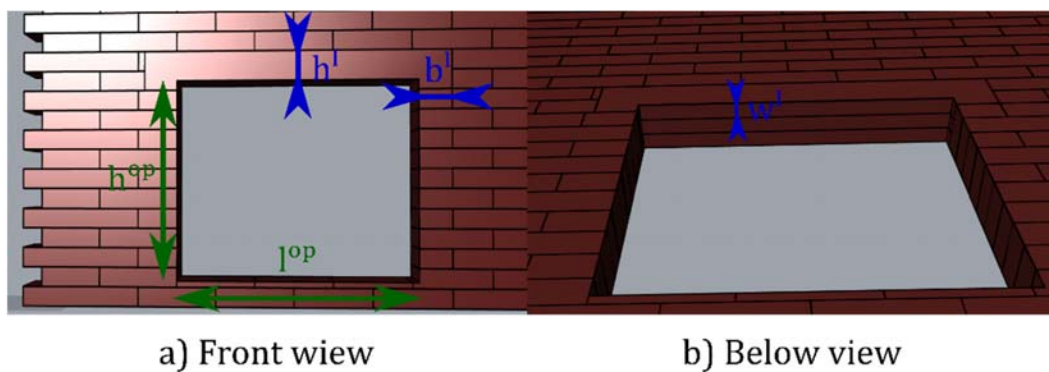


Figure 11: Parametrisation of the opening and the lintel above it

DISCUSSION AND CONCLUSIONS

The five parametrised components presented above are enough to create diverse masonry geometries (Figure 10). A simple structure like those shown in Figure 10c does not require more than 10 minutes to be built. Besides, as Figures 4 & 7 demonstrate, the time to update the geometry is less than one second. The proposed generative algorithm, therefore, reaches high performance in terms of real-time modeling. On the one hand, it can find application as an input of structural analysis software that aims at doing fast calculations (e.g. micro-modelling Limit Analysis tools [8-9, 23]). Indeed, the construction of complex geometrical models of masonry structures can be a laborious task that can largely overpass the structural analysis time, reducing the time-efficiency of these practical tools. Moreover, this tool is particularly suitable for the parameterically shape exploration to find e.g., the optimal geometry manually..

To conclude, the presented tool shows auspicious results and need to be further developed to account for multi-leaf bond patterns, gable walls, curved and inclined walls as well as non-homogeneous block size, which are all envisioned as next development steps. Furthermore, the automatisisation of the exportation of the generated masonry geometries to classical structural analysis software is currently being investigated.

ACKNOWLEDGEMENTS

This study has been funded by the STAND4HERITAGE project (new standards for seismic assessment of built cultural heritage) that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 833123), as an Advanced Grant. Its support is gratefully acknowledged. The opinions and conclusions presented in this paper are those of the authors and do not necessarily reflect the views of the sponsoring organisation.

REFERENCES

- [1] Alessandri, C., Garutti, M., Mallardo, V., & Milani, G. (2015). Crack patterns induced by foundation settlements: Integrated analysis on a renaissance masonry palace in Italy. *International Journal of Architectural Heritage*, 9(2), 111-129.
- [2] Vlachakis, G., Vlachaki, E., & Lourenço, P. B. (2020). "Learning from failure: Damage and failure of masonry structures, after the 2017 Lesvos earthquake (Greece)." *Engineering Failure Analysis*, 117, 104803.
- [3] Bruneau, M. (1994). "Seismic evaluation of unreinforced masonry buildings—A state-of-the-art report." *Canadian Journal of Civil Engineering*, 21(3), 512-539.
- [4] Leite, J., Lourenco, P. B., & Ingham, J. M. (2013). Statistical assessment of damage to churches affected by the 2010–2011 Canterbury (New Zealand) earthquake sequence. *Journal of Earthquake Engineering*, 17(1), 73-97.
- [5] Lourenço, P. B., Mendes, N., Ramos, L. F., & Oliveira, D. V. (2011). "Analysis of masonry structures without box behavior." *International Journal of Architectural Heritage*, 5(4-5), 369-382.

- [6] de Felice, G., De Santis, S., Lourenço, P. B., & Mendes, N. (2017). "Methods and challenges for the seismic assessment of historic masonry structures." *International Journal of Architectural Heritage*, 11(1), 143-160.
- [7] D'Ayala, D., & Speranza, E. (2003). "Definition of collapse mechanisms and seismic vulnerability of historic masonry buildings." *Earthquake Spectra*, 19(3), 479-509.
- [8] Gagliardo, R., Portioli, F. P. A., Cascini, L., Landolfo, R., & Lourenço, P. B. (2021). A rigid block model with no-tension elastic contacts for displacement-based assessment of historic masonry structures subjected to settlements. *Engineering Structures*, 229, 111609.
- [9] Portioli, F., Casapulla, C., Gilbert, M., & Cascini, L. (2014). Limit analysis of 3D masonry block structures with non-associative frictional joints using cone programming. *Computers & Structures*, 143, 108-121.
- [10] Funari, M. F., Spadea, S., Lonetti, P., Fabbrocino, F., & Luciano, R. (2020). Visual programming for structural assessment of out-of-plane mechanisms in historic masonry structures. *Journal of Building Engineering*, 31, 101425.
- [11] Bui, T. T., Limam, A., Sarhosis, V., & Hjiiaj, M. (2017). Discrete element modelling of the in-plane and out-of-plane behaviour of dry-joint masonry wall constructions. *Engineering Structures*, 136, 277-294.
- [12] Meriggi, P., de Felice, G., De Santis, S., Gobbin, F., Mordanova, A., & Pantò, B. (2019). Distinct element modelling of masonry walls under out-of-plane seismic loading. *International Journal of Architectural Heritage*, 13(7), 1110-1123.
- [13] Valente, M., Milani, G., Grande, E., & Formisano, A. (2019). Historical masonry building aggregates: advanced numerical insight for an effective seismic assessment on two row housing compounds. *Engineering Structures*, 190, 360-379.
- [14] Gams, M., Anžlin, A., & Kramar, M. (2017). Simulation of shake table tests on out-of-plane masonry buildings. Part (III): two-step FEM approach. *International Journal of Architectural Heritage*, 11(1), 94-102.
- [15] Landolfo, R., Gagliardo, R., Cascini, L., Portioli, F., Malena, M., Tomaselli, G., & de Felice, G. (2020). Rigid block and finite element analysis of settlement-induced failure mechanisms in historic masonry walls. *Frattura ed Integrità Strutturale*, 14(51), 517-533.
- [16] Milani, G., Esquivel, Y. W., Lourenço, P. B., Riveiro, B., & Oliveira, D. V. (2013). Characterisation of the response of quasi-periodic masonry: Geometrical investigation, homogenisation and application to the Guimarães castle, Portugal. *Engineering structures*, 56, 621-641.
- [17] Riveiro, B., Lourenço, P. B., Oliveira, D. V., González-Jorge, H., & Arias, P. (2016). Automatic morphologic analysis of quasi-periodic masonry walls from LiDAR. *Computer-Aided Civil and Infrastructure Engineering*, 31(4), 305-319.
- [18] Valero, E., Bosché, F., Forster, A., & Hyslop, E. (2019). Historic digital survey: reality capture and automatic data processing for the interpretation and analysis of historic architectural rubble masonry. In *Structural Analysis of Historical Constructions* (pp. 388-396). Springer, Cham.
- [19] Park, H. S., Lee, H. M., Adeli, H., & Lee, I. (2007). A new approach for health monitoring of structures: terrestrial laser scanning. *Computer-Aided Civil and Infrastructure Engineering*, 22(1), 19-30.
- [20] Savalle, N., Vincens, E., & Hans, S. (2018). Pseudo-static scaled-down experiments on dry stone retaining walls: Preliminary implications for the seismic design. *Engineering Structures*, 171, 336-347.

- [21] Restrepo Vélez, L. F., Magenes, G., & Griffith, M. C. (2014). Dry stone masonry walls in bending—Part I: Static tests. *International Journal of Architectural Heritage*, 8(1), 1-28.
- [22] Graziotti, F., Tomassetti, U., Penna, A., & Magenes, G. (2016). Out-of-plane shaking table tests on URM single leaf and cavity walls. *Engineering Structures*, 125, 455-470.
- [23] Cascini, L., Gagliardo, R., & Portioli, F. (2018). LiABlock_3D: a software tool for collapse mechanism analysis of historic masonry structures. *International Journal of Architectural Heritage*, 14(1), 75-94.
- [24] Gentry, T. R., Eastman, C., Haymaker, J., Lee B., Development and Documentation of Case Studies of Masonry Construction Projects, technical report, 2016, [project-ii-benchmark-project-final-report-from-georgia-tech-january-16-2016.pdf](#) (bimformasonry.org)