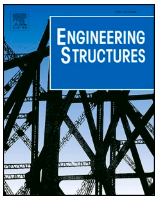
Engineering Structures 342 (2025) 120916

Contents lists available at ScienceDirect

Engineering Structures

journal homepage: www.elsevier.com/locate/engstruct



Breaking boundaries: discontinuum failure analysis of dry-joint masonry using physics engine models

A. Wang [a,*], B. Pulatsu [b], S. Andrews [c], D. Malomo [a]

[a] Department of Civil Engineering, McGill University, 817 Sherbrooke Street West, Montreal, QC H3A 0C3, Canada
[b] Department of Civil and Environmental Engineering, Carleton University, 1125 Colonel By Drive, Ottawa, ON K1S 5B6, Canada
[c] Department of Software and IT Engineering, École de Technologie Supérieure (ÉTS), 1100 Notre-Dame St W, Montreal, QC H3C 1K3, Canada

ARTICLE INFO

Keywords:
Physics engine
Discontinuum analysis
Discrete models
Unreinforced masonry
Dry-joint
In-plane
Out-of-plane
PyBullet
Bullet Physics

ABSTRACT

Discontinuum approaches, including the Distinct Element Method (DEM), are well-established for simulating the failure of dry-joint unreinforced masonry (URM) structures, particularly under in-plane (IP) shear-compression and out-of-plane (OOP) loading. However, they may be computationally intensive, with building-scale DEM analyses requiring up to 3 days for 15 s of seismic loading. This paper breaks boundaries between structural engineering and computer science by presenting the first systematic evaluation of PyBullet, an open-source physics engine based on Bullet Physics – originally conceived for visually plausible virtual animations – for simulating the mechanical response and collapse of dry-joint URM assemblies at different scales. Leveraging PyBullet's rigid body algorithms, contact models, and efficient constraint solvers, 3D simulations were performed for IP shear-compression walls, settlement-induced damage in interlocking panels, and OOP tilting of URM. Results were benchmarked against experimental data and established discontinuum models – PyBullet predicted peak loads within + 16 % of DEM for IP shear-compression walls. Numerical stability was maintained with time steps in the order of 0.001 s, and full simulations completed within 5 min – up to 6 times faster than DEM. OOP tilting analyses reproduced expected collapse modes (diagonal cracking, overturning) with critical collapse angles underestimated by up to 32 %, largely due to premature block slippage linked to contact stiffness and friction force coupling. Settlement-induced failure in interlocking panels showed good agreement with experimentally observed failure patterns, with ultimate displacements within ±3 % for non-interlocking cases. The study demonstrates that PyBullet offers a computationally efficient alternative for dry-joint URM analysis, providing reduced runtimes and acceptable predictive accuracy, especially for preliminary or large-scale probabilistic assessments. Further refinement of contact stiffness calibration strategies would enhance predictive consistency, supporting the adoption of physics engines as viable alternatives to conventional discontinuum methods for rapid masonry collapse and debris simulations.

1. Introduction

Numerical modelling has become the reference method for both practitioners and applied researchers to perform building-scale structural evaluations [83] – complex tasks that would otherwise often be "overambitious" to accomplish using simpler approaches [73]. Discontinuum analysis strategies, examples of which are the Distinct Element Method (DEM) [28], the Applied Element Method (AEM) [128], and Non-Smooth Contact Dynamics (NSCD) [70], originally formulated for simulating the mechanical behaviour of particulate systems [31], are now recognized as among the leading methods for assessing complex unreinforced masonry (URM) structures [32] – especially until collapse. These computational techniques can perform automatic detection of contact points between colliding bodies while simultaneously yielding large displacements [30,93] and are thus being increasingly employed beyond URM in structural, forensic and bridge engineering [116,58]. Although less scrutinized for URM modelling compared to traditional limit analysis [131,48,65] and continuum (e.g., equivalent frame method (EFM) [103,111,114,19,21], Finite Element Method (FEM) [124,18,24,97] or storey-mechanism models [130], discontinuum methods are distinguished in their ability to capture re-contact after separation between individual units, enabling a

[*] Corresponding author.
E-mail address: anna.h.wang@mail.mcgill.ca (A. Wang).

https://doi.org/10.1016/j.engstruct.2025.120916
Received 3 April 2025; Received in revised form 19 June 2025; Accepted 6 July 2025
Available online 17 July 2025

combined assessment of in-plane (IP) and out-of-plane (OOP) structural behaviour up-to and beyond near-collapse [118,91]. This explicit representation of progressive cracking and separation processes makes discontinuum methods exceptionally suited for debris prediction (i.e., debris area, volume, and distribution) [16,36,38], a product of collapse behaviour. Comprehensive analysis of (especially OOP) collapse mechanisms and post-failure consequences is vital for appropriately informing loss assessment due to natural hazards (e.g., earthquakes floods, landslides, see [1]), and disaster preparedness (e.g., assessing road accessibility for emergency vehicles) [115,126,47]. The aforementioned researchers have demonstrated the capabilities of discontinuum models in assisting field-informed and survey-based damage recognition efforts.

Nonetheless, and despite their unique capabilities, discontinuum micro- and meso-modelling – see Malomo and Pulatsu [93] for definitions of modelling abstraction levels used herein – requires prohibitive analysis times and technical skills not readily compatible with those of practicing engineers and applied researchers. DEM seismic modelling of two adjacent URM structures by Galvez et al. [50] required reducing time-step size for numerical stability, leading to substantially long computational times (average of 99 h for 15 s of dynamic analysis). In this case, the original earthquake time-history duration was even truncated by 61 % to achieve reasonable runtimes – similarly completed by Mehrotra and DeJong [95]. Recent strategic geometrical discretization techniques have improved discontinuum computational efficiency, by artificially increasing block size or proposing the use of component-scale mesh layouts. Ferrante et al. [44] reduced computational time for DEM and NSCD seismic analyses of a small (~136 m^2) URM church from 1600 to 100 h using equivalent macro-blocks. Malomo and DeJong [89] developed a macro-block-based approach (M-DEM), which defined a priori deformable macro-block discretizations informed by expected fracture patterns. Their methodology resulted in improved performance (up to 100 times faster than the corresponding meso-model), at the cost of a simplified representation of URM damage. This has been experimentally validated for predicting the IP response of retrofitted URM by Damiani et al. [34] - see Liu et al. [79] and Damiani et al. [35] for additional recent applications of DEM for retrofitted URM. Also using DEM, Zhang et al. [137] proposed a semi-automated EFM-inspired macro-discretization that yielded computational times up to 150 times faster than standard discontinuum meso-models (albeit still only offering idealized failure modelling) by combining larger rigid blocks and energy-based interface constitutive laws [107]. Building-scale seismic DEM and NSCD comparative meso-modelling of an existing URM church required 161 h and 744 h of computational time, respectively, despite using oversized rigid blocks [117]. AEM approaches have also leveraged the incremental ground acceleration (an alternative loading scheme where a mass-proportional lateral force is applied to each unit) to evade computational time typically used for determining the appropriate lateral load pattern [113,120]. For further discussion of discontinuum methods, including their formulations, strengths, limitations, and applications for URM structures at varying scales, the reader is referred to Malomo and Pulatsu [93].

An underexplored alternative to trusted URM discontinuum modelling is the use of physics engines – user-friendly simulation platforms conceived for reproducing visually plausible physical phenomena primarily for 3D virtual animations in videogames and movies. Physics engines present surprising conceptual similarities with discontinuum modelling: both employ interfaces, block interaction principles, and point-contact models embedded in a time-stepping algorithm. However, they feature an unmatched computational speed when simulating interactions between rigid body elements [23,67,99] due to their simplified and more efficient formulation – as further discussed in the following sections. The typical physics engine contact simulation algorithm consists of three consecutive processes performed at each time-step: (1) collision detection algorithms run to identify adjacent and penetrating bodies [100], (2) contact forces are calculated based on established constraints (e.g., boundary conditions) and input contact laws, and (3) displacements are updated using the chosen solver to integrate the equations of motion [3,63]. Block interfaces can take the form of hard contact models (i.e., those with discontinuities and strict enforcement of non-penetration requirements) which are analogous to those from NSCD, soft contact models (i.e., those that allow certain levels of interpenetration between bodies), and various friction models (e.g., Mohr-Coulomb, bristle, anisotropic matchstick) [122,41,59,62, 78]. While this algorithm can lead to computational challenges (e.g., numerical stability, convergence, efficiency) if implemented in typical discontinuum frameworks, physics engines overcome this through strategies such as leveraging GPU solvers [121,2] (studies have also begun exploring GPU architectures for traditional DEM too, see e.g. [55]), utilizing more efficient numerical methods (e.g., direct solvers [5, 80], semi-implicit Euler time discretization [3]), and allowing flexibility (i.e. simplifications) with kinematic and geometrical constraints (e.g., Baumgarte stabilization, error reduction parameter (ERP), constraint force mixer (CFM)) [20,6]. The effect of these creative numerical solutions, however, have yet to be widely validated for rigorous mechanical simulations of structural engineering systems through comparisons with proven discontinuum methods.

Physics engines, while established in fields such as robotics [25,26, 71], biomedical engineering [42,87], fracture mechanics [119,129,52, 99] and geotechnical engineering [63,64], have limited presence in structural engineering applications and lean on graphical references (e.g., photos and videos of damage and debris) for visual realism [76,84] and/or established computational methods for numerical legitimacy (especially for pre-collapse behaviour). Hybrid continuum-physics engine frameworks have been explored for reinforced concrete (RC) collapse, but relied on FEM for complex behaviour such as nonlinear small deformations [77] and punching failure mechanisms [85]. Similarly, the occasional use of physics engines for structural analysis of URM can be found in literature, albeit with varying degrees of numerical rigor and typically limited quantitative validation exercises. A NSCD contact model implemented into the physics engine Project Chrono [94] produced visually comparable results with the graphical thrust-line method when analyzing dry-joint arches under static and dynamic loads. Real-time simulations, regularization (e.g., introducing compliance in contact models to avoid indeterminant systems), and stability with larger time-steps demonstrate the overall appeal of physics engines [7]. Whiting et al. [135] used Bullet Physics to dynamically perturb URM structures developed using procedural modelling methods, albeit without quantitative validation of seismic responses. Ma et al. [86] validated the response of a single block under free and forced rocking through comparing rotational displacements with those obtained from Housner's analytical mechanics equations [66], obtaining reasonable agreement. However, only limited and oversimplified configurations (i.e., a single macro-block with varying aspect ratio) were investigated for verifying URM wall rocking under base excitation in proof of concept. Seismic simulations of old URM structures using Bullet Physics were compared with results from SAP2000 (FEM) by Fita et al. [45] – URM unit displacement was agreeable between the two models, but was underestimated (up to −70 %) in Bullet and did not exhibit nonlinear behaviour. The Bullet Constraint Builder, an add-on in Blender conceptually based on DEM that allows the user to create custom constraints (e.g., mortar joints), has also been used for seismic URM collapse visualization, exhibiting realistic damage patterns and debris distribution. Nonetheless, without any output of predicted forces [9].

In this paper, we break boundaries between computer science and structural engineering by unprecedently exploring the capabilities of PyBullet, a Python module operating the well-known, open-source Bullet Physics engine [26], in replicating the experimentally observed IP and OOP failures of dry-joint URM at different scales. Dry-joint URM structures are representative of a large portion of today's existing building stock, whether an older structure built without mortar or one experiencing mortar degradation [136]. The shear-compression behaviour of dry-joint URM assemblies is typically governed by

Mohr-Coulomb assumptions of pure friction – simpler constitutive models compared to mortar counterparts. This lends dry-joint URM as an ideal candidate as a benchmark for preliminary validations, as the exploratory ones presented herein. For appropriate comparisons with our physics engine models, further discussion of discontinuum dry-joint analysis is presented in Section 2. PyBullet was selected as the simulation program for its accessibility amongst engineers – Python is an intuitive coding language, open-access, and has extensive documentation. Although PyBullet is not yet equipped with GPU computing resources, the Bullet Physics Software Development Kit (SDK) has an OpenCL GPU backend with potential for implementation in PyBullet. The contact mechanics (e.g., rigid body kinematics, time-stepping schemes, point contact models) of physics engines are thoroughly examined in Section 3. Contact forces output from IP shear-compression, IP settlements (Section 4), and OOP tilting (Section 5) simulations are quantitatively evaluated against experimental outcomes and discontinuum predictions, with emphasis on differences in the prediction of force capacities, displacement rates and failure mechanisms. Results from this novel adaptation and subsequent application to URM modelling support the feasibility of PyBullet as an acceptable, rigorously scrutinized discontinuum option that can overcome the limitations of already established discontinuum techniques.

2. Discontinuum modelling of dry-joint masonry

The global structural behaviour of dry-joint URM assemblies depends on the mechanical interaction between individual masonry units through shear-compression and crack opening/closing due to tension/flexure [134]. As such, dry-joint URM is often modelled in discontinuum domains using meso-scale blocks with Mohr-Coulomb no-tension/infinite compression relationships at the contact points [106]. While not classified as a discontinuum approach, it is noted that limit analysis formulations proposed by Portioli et al. [104] and Casapulla and Portioli [17] have also proven effective for analysis of dry-joint URM structures. This section discusses existing applications of DEM and NSCD (to the authors' knowledge, no implementations of AEM for exclusively dry-joint URM are found in literature) for dry-joint URM components (e.g. isolated walls) and assemblies (multiple interconnected components, e.g. façades, vaults).

Extensive validations for IP and OOP dry-joint URM simulations using DEM are present in literature. Bui et al. [15] compared numerical results from the commercial three-dimensional DEM software, 3DEC [68], with an existing suite of experimental tests on dry-joint URM walls [112,81]. PyBullet comparisons with select simulations from this study are presented in the next sections. The authors above used an elastic-perfectly plastic Coulomb slip joint model with experimentally measured friction angles and contact stiffnesses calculated based on equations from Lourenço et al. [81] (cohesion, tensile strength, and angle of dilatancy were set to 0). Small-scale IP shear-compression models of walls illustrated de-bonding of the top beam and the diagonal stepped crack pattern, similarly observed in experimental tests (with the exception of those that failed in crushing). The inability to capture individual unit compressive failure yielded an overestimation of force capacity for high compression loads (up to 28 %). Hybrid FEM-DEM strategies have overcome this limitation using crack models to simulate intra-unit fracture and fragmentation [125,22]. Quasi-static OOP tilting for C-walls and a two-storey URM construction exhibited expected overturning failure mechanisms (e.g., named A, B2, G by D'Ayala and Speranza [33]). Collapse load factors from DEM were also agreeable (-16–17 %) with experimental results. Nonetheless, predicted maximum accelerations for harmonic seismic loading for rectangular URM structures overestimated (~50 %) those from scaled (1:10) experimental results. The authors associate this discrepancy to friction uncertainties (e.g., heterogeneous contact surfaces, inconsistencies with normal stress distribution) with the scaled model that were not captured with Coulomb friction – similarly concluded by Foti et al. [46] for IP settlement models of dry-joint cross vaults. More sophisticated, velocity-dependent friction models (e.g., Stribeck friction) are proposed by Bui et al. [15] for future studies – capabilities currently available in physics engines [3]. To address these uncertainties at unit-joint interfaces, Pulatsu et al. [109] presented a stochastic discontinuum analysis approach with both spatial and non-spatial distribution of contact properties. Spatial analyses produced reduced variability in IP shear-compression force-displacement curves for mixed-mode failure mechanisms (i.e., diagonal shear cracking) – the increased predictability is attributable to the explicit representation of masonry's heterogeneous features. DEM simulations of dry-joint URM are also particularly sensitive to contact stiffness – higher contact stiffnesses require smaller critical time-steps and can result in overestimations of force capacity, as demonstrated by Pulatsu et al. [108] for IP shear-compression for walls. Nevertheless, sensitivity studies reveal that all simulations result in the same failure mechanism regardless of contact stiffness [110]. Seismic models of dry-joint groin vaults by Bianchini et al. [11] used calibrated contact stiffness based on the estimated first natural frequency. Notably, their DEM models with damping required exceedingly small time-steps (10^{-9} s), requiring more than 2 months of simulation time for a single earthquake record. Thus, null damping was exceptionally used for their DEM models (yielding a time-step of 10^{-6} s and 12 h of computational time for one record), allowing the authors to conduct a sensitivity analysis with a series of 40 earthquake records – a study that would be unattainable otherwise.

While not as widely and rigorously substantiated using experimental results for dry-joint applications, NSCD models nonetheless exhibit expected failure mechanisms and quantitative consistencies with DEM and analytical results. Comparisons with LMGC90 (NSCD code) [39] and UDEC (2-D counterpart to 3DEC) [69] for IP shear-compression tests of three dry-joint URM panels with varying geometry reveal reasonable agreement for force-displacement curves (especially for the linear softening post-peak branch), maximum load multiplier (+2.93 % max error), and failure mechanisms (diagonal stepped cracking) [14]. The authors note that a full pushover analysis in LMGC90 can demand up to 2 days of processing time compared to a few hours using UDEC. Lancioni et al. [74] also used LMGC90 to identify seismic vulnerabilities of existing URM churches (i.e., ones experiencing mortar degradation and thus can be approximated as dry-joints since there is minimal cohesion). Their study highlights the extreme sensitivity to input parameters (e.g., friction) for non-smooth systems. The authors note that this can be a silver lining for models of existing structures as material properties are often uncertain and variable, similarly corroborated by [13]. For NSCD models with high friction coefficients, "rotation-like" and OOP collapse mechanisms were observed, while models with lower friction coefficients yielded "translation-like" mechanisms (i.e., sliding) – consistent with expected "tipping" versus "slipping" behaviour for Coulomb friction. NSCD simulations of an old URM tower by Ferrante et al. [43], while exhibiting excessive computational times (up to ~2222 h), demonstrated agreement with analogous DEM models with respect to displacements of control points and activated failure mechanisms (e.g., diagonal cracking, rocking, vertical splitting of walls). Beatini et al. [8] introduced a pseudo-non-associative friction law into their NSCD model for dry-joint segmental and multi-ring URM arches to better illustrate shear sliding phenomena (observed experimentally and in DEM by Dimitri and Tornabene [37] and Misseri et al. [98]). This formulation relies on associative friction (i.e., direction of sliding is related to the direction of normal stress) at each time-step and implements an ad hoc stabilization procedure to negate effects of dilatancy, rendering the overall procedure non-associative. For numerical efficiency and stability, a spectral projected gradient iteration solver (which uses gradients to guide convergence) [12] is used in lieu of a fixed point iterative solver (which enforces the contact model's cone complementarity constraints individually, e.g. Projected Gauss-Seidel) [123] typically used in NSCD. Although no force-displacement results from this framework were numerically scrutinized, collapse capacity and modes aligned with those from conventional thrust line models. The numerical methods (i.e., alternative solvers and friction models) used by Beatini et al. [8] evoke similarities with those that characterize physics engines [3] – albeit, both analyses require further understanding of parameter sensitivities and numerical validations with experimental and discontinuum techniques referenced herein.

3. Physics engines: from visually plausible to rigorous sims

This section presents a high-level overview of the contact mechanics formulation in physics engines, from a discontinuum numerical modelling perspective. This effort of "translating" established terms and definitions in computer science to structural engineering readers and vice versa is crucial and part of "breaking boundaries". As shown in this section, the two disciplines often refer to the same thing in distinct ways (e.g. structural engineering "simulations" becomes "sims" in computer science), while also providing different solutions to solve analogous issues. Discussion will focus on the underlying principles that dictate the mechanics of rigid blocks (e.g., equations of motion, dry contact constitutive laws), highlighting numerical methods and input parameters unique to physics engines. Comprehensive examination of contact mechanics in DEM, AEM, and NSCD can be found in Hart et al. [61], Tagel-Din [128], and Dubois and Jean [39], respectively – readers are referred therein to draw more detailed comparisons with physics engines contact models. Following notations (e.g., bolded variables denote vectors and matrices) and equations are adopted (and "translated") from Andrews et al. [3].

Bullet Physics uses a semi-implicit time integration scheme [40] (in contrast to explicit schemes used by e.g. DEM) [29,75] as a compromise between numerical stability, computational time, and mechanical accuracy. Time discretization is often quantified as frames per second, *FPS*, the rate at which the system can advance to the next frame. The desired visual fidelity of the simulation often dictates the chosen frame rate – e.g., a higher frame rate (60–120 fps) for real-time and/or first-person videogames and lower (<30 fps) for 3D animations [10]. To increase numerical stability, each frame may be divided into n_s sub-steps – contact mechanics are calculated at each sub-step [88] but the updated rigid body kinematics at sub-steps are not always visually output. The actual time-step, Δt, used in the time integration scheme can be related to *FPS* and n_s using Eq. (1).

$$\Delta t = \frac{1}{(n_s + 1) \bullet FPS} \tag{1}$$

Collision detection processes – present in other discontinuum methods for simulating separation and re-contact – are run at the start of each time-step, Δt, to identify bodies that are in "contact". "Contact" in physics engines is defined as the intersection of two bodies' collision shapes (i.e., approximations of the rigid body's geometry). For blocky assemblies, boxes or convex hulls (i.e., convex polygon that encloses a shape) are preferred – the latter is more effective at representing detailed geometries, though it comes with increased computational expense (i.e., additional combinations of faces, vertices, and edges to check for intersections). Collision shapes can also be inflated using a "collision margin", which adds a buffer around the original shape and helps with catching edge cases (e.g., edge-edge intersections). The collision detection pipeline generally occurs in two phases: (i) broad-phase (uses a simple shape, e.g., spheres or boxes, to discard combinations of bodies that cannot be in contact for computationally efficient collision tests) and (ii) narrow-phase (checks the remaining bodies for contact using high-level fidelity of the object geometry). Three key outputs are produced from the collision detection process: (i) discrete contact points and their position in global space, $\boldsymbol{u}$, (ii) a contact normal direction to inform how bodies should move to avoid further interpenetration, and (iii) a penetration (i.e., gap) measure, $\varphi(\boldsymbol{u})$, describing the distance between bodies (or distance that two bodies are intersecting). The kinematics of the identified contact points can be characterized using Newton-Euler equations of motion [53], which take the form of a second-order ordinary differential equation: $\boldsymbol{M}\ddot{\boldsymbol{u}} = \boldsymbol{f}$, where $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ is the system masses, $\ddot{\boldsymbol{u}} \in \mathbb{R}^n$ is the respective accelerations, and $\boldsymbol{f}$ is the function that defines the applied forces on the system for each degree of freedom n. The equations of motion are typically solved at the velocity level for computational efficiency and to ensure stable response to collision forces and impulses. Substituting the first-order Taylor series approximation of the implicit rigid body velocities, $\dot{\boldsymbol{u}}^+ \approx \dot{\boldsymbol{u}} + \Delta t \ddot{\boldsymbol{u}}$, (the superscript + indicates implicit quantities) into the equations of motion yields a linear relationship between the system velocities and the time-step (Eq. (2)).

$$\boldsymbol{M}\dot{\boldsymbol{u}}^+ = \boldsymbol{M}\dot{\boldsymbol{u}} + \Delta t \boldsymbol{f} \tag{2}$$

Most physics engines use constraint-based methods to enforce boundary conditions and other mechanical requirements (e.g., interface constitutive laws). The "constraint manifold" (i.e., set of all possible rigid body motions that satisfy the given constraints at the given instant) is expressed using the gap function, $\varphi(\boldsymbol{u}) \in \mathbb{R}^j$, where j is the number of constraints. Two types of constraints are present in physics engines: (i) bilateral constraints, $\varphi(\boldsymbol{u}) = 0$ (e.g., hinges, ball-and-socket joints), and (ii) unilateral (i.e., contact) constraints, $\varphi(\boldsymbol{u}) \geq 0$ (e.g., objects cannot penetrate each other). The contact impulses, $\boldsymbol{\lambda}^+$, needed to enforce these constraints are applied in a direction determined by the constraint gradient, $\boldsymbol{J} \in \mathbb{R}^{j \times n}$ (Eq. (3)) (assumed to be constant throughout the time-step).

$$\boldsymbol{J} = \frac{\partial \varphi(\boldsymbol{u})}{\partial \boldsymbol{u}} \tag{3}$$

The contact normal impulse, $\lambda_{\hat{\boldsymbol{n}}}{}^+$, is defined by with a "push-only" spring-dashpot (regenerated at each time-step) at each contact point (Fig. 1). Note that italicized notation, λ^+, represents impulse, whereas the not italicized notation, $\mathrm{\lambda}^+$, represents the equivalent force such that $\lambda^+ = \mathrm{\lambda}^+ \Delta t$. Although the spring adds potential energy to the system, it is paired with a damper to dissipate the added energy/energy lost during impacts and to prevent unrealistic collisions (e.g., high impacts) [6]. The magnitude of the contact normal force is dependent on the relative velocity, $v_{\hat{n}}$, and displacement, $\Delta u_n{}^+$, in the normal direction and can be calculated as follows:

$$\mathrm{\lambda}_{\hat{\mathrm{n}}}{}^+ = -k_n \Delta u_n{}^+ - c v_{\hat{n}}{}^+ \tag{4}$$

Parameters k_n and c are the contact stiffness and damping coefficients, and the relative normal displacement is equivalent to the first order Taylor expansion approximated constraint error term, $\Delta u_n{}^+ = \varphi^+ = \varphi + \Delta t v_{\hat{n}}{}^+$. Unlike contact stiffness in DEM (which can be calculated using dry-joint material properties, see [81,15,134]), contact stiffness and damping in physics engines are chosen based on the required level of constraint stabilization. Their relationship will be examined later in this section. The interface constitutive relationship resulting from Eq. (4) dictates that a greater contact normal impulse is required to prevent further overlap if the bodies are expected to continue penetrating in the next time step (i.e., $v_{\hat{n}}{}^+ < 0$), and a smaller or zero-magnitude impulse is applied if the bodies are in constant contact (i.e., $v_{\hat{n}}{}^+ = 0$) or moving away from each other (i.e., $v_{\hat{n}}{}^+ > 0$). No impulse is applied if the bodies are not in contact (i.e., zero tensile strength, $f_t = 0$). For comparison with dry-joint contact models in discontinuum methods, the physics engine contact model in the stress-displacement domain is shown in Fig. 1.

Physics engines simulate friction phenomena through the application of a contact impulse in the shear (i.e., contact frame tangential) direction, $\lambda_{\hat{\boldsymbol{t}}}$. For isotropic Coulomb friction models, the shear impulse is coupled with the normal impulse such that $\|\lambda_{\hat{\boldsymbol{t}}}\| \leq \mu \|\lambda_{\hat{\boldsymbol{n}}}\|$, where μ is the coefficient of friction – this relationship can be visually represented using a quadratic "friction cone" (Fig. 1). The relative tangential velocity, $\mathbf{v}_{\hat{\mathbf{t}}}$, dictates whether the bodies are sticking (i.e., $\|\mathbf{v}_{\hat{\boldsymbol{t}}}\| = 0$) or slipping (i.e., $\|\mathbf{v}_{\hat{\boldsymbol{t}}}\| > 0$). If sticking, the friction impulse must lie within

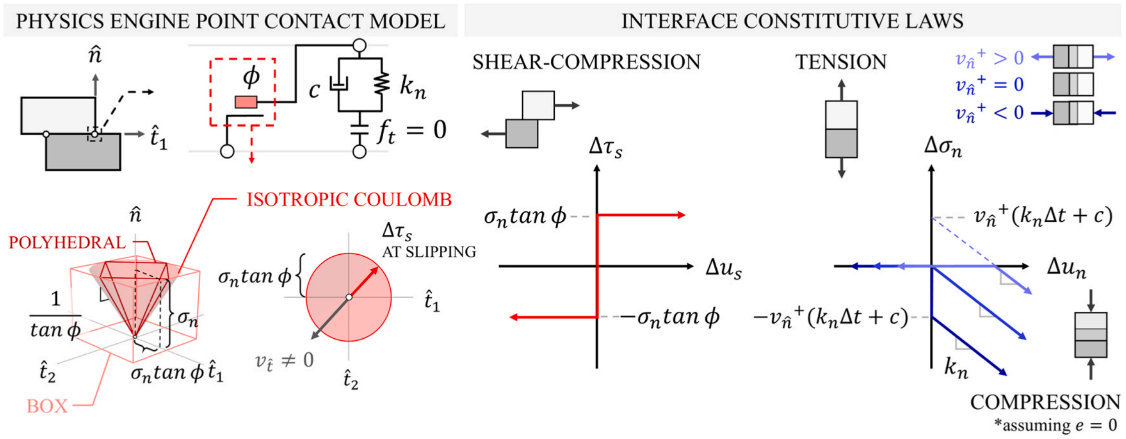


Fig. 1. Physics engine point contact model and interface constitutive laws in shear-compression, tension, and compression expressed in the stress-displacement domain – adapted from Andrews et al. [3].

the friction cone (i.e., $\|\lambda_{\hat{t}}\| < \mu\|\lambda_{\hat{n}}\|$); however, there can be multiple solutions to satisfy the sticking requirement. If slipping, the friction impulse lies on the surface of the friction cone (i.e., $\|\lambda_{\hat{t}}\| = \mu\|\lambda_{\hat{n}}\|$) and occurs in the direction that ensures maximum energy dissipation. Friction impulses in physics engines can be under or overestimated due to their coupling with the contact normal impulse (which is influenced by choice of contact stiffness and damping) and/or if the friction problem is linearized for numerical efficiency – e.g., polyhedral cone approximation [127,4], box approximation [72], which simplify the envelope of possible friction forces into a discrete number of unit vectors defined by the approximated shape of the friction cone.

The implicit velocities, $\dot{\boldsymbol{u}}^+$, and impulses, $\boldsymbol{\lambda}^+$, are solved using the linear system in Eq. (5), which is assembled using the equations of motion (first row of Eq. (5)) and the constraint manifold (second row of Eq. (5)). The constraint stabilization parameters ϵ_j (Eq. (6)) and υ_j (Eq. (7)) – the constraint force mixing (CFM) and error reduction parameters (ERP) for each constraint j, respectively – are derived from the non-interpenetrative contact normal impulse, $\lambda_{\hat{n}}{}^+$, and therefore can be tuned using k_n and c. The ERP describes the fraction of the constraint error, $\varphi^+ = \Delta u_n{}^+$, that is resolved in the next time-step, while the CFM introduces artificial constraint relaxation to the system (i.e., allowing a tolerance for constraint violation). The coefficient of restitution, e_j, (typically 0) (Eq. (8)) is also introduced into the right-hand side of Eq. (5) when setting up the constraint manifold and defines the fraction of the initial velocity ($\dot{u}$) that lost due to impact.

$$\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{J}^T \\ \boldsymbol{J} & \boldsymbol{\Sigma} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}^+ \\ \boldsymbol{\lambda}^+ \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}\dot{\boldsymbol{u}} + \Delta t\boldsymbol{f} \\ -\boldsymbol{\Upsilon}\dfrac{\boldsymbol{\varphi}}{\Delta t} - \mathbf{E}\boldsymbol{J}\dot{\boldsymbol{u}} \end{bmatrix} \tag{5}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \epsilon_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \epsilon_j \end{bmatrix} \tag{6}$$

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \upsilon_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \upsilon_j \end{bmatrix} \tag{7}$$

$$\mathbf{E} = \begin{bmatrix} e_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e_j \end{bmatrix} \tag{8}$$

To solve Eq. (5), physics engines will use a constraint-based solver – e.g., pivoting methods (which find an exact solution using a direct solver and are computationally efficient) [80], iterative methods (which find an approximate solution using iterations) [123,96]. The choice of solver is guided by the nature (e.g., non-linear, linear, mixed) of the complementarity conditions accompanying Eq. (5) (not shown herein – e.g., the friction forces must lie within the "friction cone", normal impulses must be positive). The Projected Gauss-Seidel (PGS) [123] iterative method is the default solver used in Bullet Physics. Although iterative methods – which are widely used in DEM (e.g., Newton-Raphson) – are computationally demanding, physics engines will often introduce a cap on the number of iterations rather than a tolerance for convergence (i.e., the "solver iterations" parameter). However, too few solver iterations can possibly contribute to underestimating contact normal impulses. The obtained velocities are then used to update the rigid body positions, $\boldsymbol{u}^+$, using the implicit approximation, $\boldsymbol{u}^+ = \boldsymbol{u} + \Delta t\boldsymbol{S}\dot{\boldsymbol{u}}^+$, where $\boldsymbol{S}$ maps the angular velocities in $\dot{\boldsymbol{u}}^+$ to their respective rigid body quaternion orientations.

4. In-plane failure analysis

The 3D PyBullet simulations presented in Sections 4 and 5 were run using Python version 3.12 and PyBullet version 3.2.6 [26] on a Dell Precision 7865 Tower PC equipped with an AMD Ryzen Threadripper 12-core 24-thread CPU. The PyBullet *p.stepSimulation* command was used to manually step to the next frame to maintain control determinism of the simulation (i.e., allowing the user precise control over the simulation's progression and ensure reproducible results). URM was modelled using a meso-scale discretization with box collision shapes representing units, and zero-thickness contact interfaces for modelling unit-mortar interactions (*p.GEOM_BOX* flag in PyBullet) – typically 4 contact points are generated between two boxes stacked on top of each other. The contact friction model used in all PyBullet simulations follows the non-linear complementarity problem defined by the isotropic friction cone (the default model in PyBullet) with the friction anchor (i.e., friction drift correction designated by the user-input ERP) enabled. Block interfaces were assigned zero cohesion, tensile strength (f_t), and dilatancy – similarly designated in DEM dry-joint URM models discussed in Section 2 and presented in this study. Prior to applying quasi-static loading, the simulations were run under gravity loads (set to 9.8 m/s^2) for at least 5 s, to stabilize the system. While PyBullet has a built-in graphical user interface (GUI), displaced shapes (i.e., rigid body origin translation and Euler rotation in the global frame) were queried using the *p.getBasePositionAndOrientation* command, output into a text file, and imported into Rhino for the purposes of creating figures herein. Reported PyBullet execution times in Sections 4 and 5 are for the number of time-steps simulated. DEM simulations in this study were run on the same machine using 3DEC version 7.00 [68]. All DEM models herein used the Mohr-Coulomb joint model for all block interfaces and the default quasi-static local damping scheme (damping value of 0.8) – see Itasca Consulting Group [68] for complete discussion of contact mechanics used in 3DEC.

4.1. Walls under in-plane shear-compression

Two sets of dry-joint URM walls with different aspect ratios (AR) under monotonic, quasi-static IP shear-compression were simulated in PyBullet and compared with previous numerical and/or experimental results. Physics engine parameters specific to IP shear-compression models are shown in Table 1. The specified ERP in Table 1 applies to constraint and friction errors – the CFM parameter is not exposed in the PyBullet application programming interface (API). The *FPS*, solver iterations, contact stiffness, and contact damping values used herein were selected to provide the best numerical stability and ensure mechanical fidelity (e.g., agreeable force-displacement outputs and failure mechanisms as discussed hereafter). The chosen contact stiffness is comparable to those used in dry-joint DEM simulations (see, e.g. [90]) and the ones obtained through experimental testing on unloaded dry-joint masonry assemblies – see Vlachakis et al. [134]. Coefficient of restitution was set to zero (energy dissipation was captured through friction assumptions and the addition of contact damping).

The first set of walls (AR = 1.0) simulated in PyBullet were experimentally tested and modelled in FEM by Lourenço et al. [81], as well as analyzed in DEM by Pulatsu et al. [109] and Bui et al. [15] – herein referred to as DEM-1 and DEM-2, respectively. All walls were 1000 mm × 200 mm × 1000 mm (length × width × height) and constructed using dry-joint masonry stone blocks with dimensions 200 mm × 200 mm × 100 mm and specific weight $\gamma = 25$ kN/m^3 [102]. Four different vertical pre-compression loads 30 kN, 100 kN, 200 kN, and 250 kN (compressive normal stresses of $\sigma = 0.15$, 0.50, 1.00, and 1.25 N/mm^2, respectively) were applied to a RC beam that was placed on top of the wall – naming conventions adopted from Lourenço et al. [81] correspond to the wall's applied loads (e.g., "SW30" for the AR = 1.0 wall with a 30 kN compressive load). An additional 1 kN load was applied to account for the steel beam on top of the RC beam in the experiment. The compressive loads were applied in PyBullet by assigning an equivalent density to an RC beam (dimensions 1100 mm × 350 mm × 200 mm) that was restrained OOP using fixed guide rails (i.e., translations and rotations were only permitted IP). To model the high strength mortar adhering the top course of blocks to the RC beam, the blocks were fixed with respect to the RC beam using the *p.createConstraint* command. A block with a height of 50 mm was fixed at the foot of the wall to prevent the bottom course from sliding. Masonry-masonry interfaces were assigned a friction angle of $\phi = 32^\circ$ (i.e., a friction coefficient of $\mu = \tan\phi = 0.624$, as experimentally obtained by [82]). To apply the horizontal load at the RC beam, a multi-body actuator consisting of two frictionless cubes with lengths of 25 mm – hereinafter referred to as the "reference" and "driver" blocks – was created using the *p.createMultiBody* function. The driver block had a mass of 10 kg and was constrained to the fixed reference block using a prismatic joint (i.e., only translation in the horizontal direction was allowed). The horizontal displacement of the driver block was controlled using the *p.setJointMotorControlArray2* function (*positionGain* and *velocityGain* flags were set to 1 to ensure minimization of the positional constraint error) which defines the target position of block at each time-step – this is treated as a bilateral constraint in the physics engine contact formulation. A quasi-static displacement rate of 0.5 mm/s was used for the SW (AR = 1.0) walls. To obtain the force-displacement curves, the sum of the contact normal forces between the RC beam and the driver block were output (using *p.getContactPoints*) as the "horizontal loads" and plotted versus the horizontal displacement of the RC beam (output using *p.getBasePositionAndOrientation*). Force-displacement data was smoothed using a moving average to eliminate noise from numerical stabilization.

Comparisons of force-displacement curves between PyBullet, DEM-1, DEM-2, and FEM show overall agreement between numerical models (Fig. 2). However, PyBullet predicted marginally higher force capacities compared to those obtained using DEM-1 and DEM-2 (max of +9.1 %, 8.1 %, 12.7 %, and 16.2 %, respectively for the four compression loads). In the absence of a clear post-peak strength degradation, walls under IP shear-compression exhibit a rocking dominated failure hinging at the toe. This allows for increased ductility and sustained maximum force capacity [109] – quantified herein as the force in the plateau region of the force-displacement curve. Wall stiffnesses (although not iteratively calibrated in PyBullet to match experimental results but rather selected for enhancing numerical stability) were also well aligned between both DEM models and PyBullet (-9.9 % to +11.5 %) for higher compression loads (i.e., SW200 and SW250). Although, PyBullet output a higher stiffness than both DEM models for the SW30 and SW100 walls (up to +111.2 % and +46.1 %, respectively). Experimental force capacities were overestimated by PyBullet and DEM-2 for the SW100, SW200, and SW250 walls. This can be attributed to their inability to capture individual unit compression failures (e.g., tensile splitting, crushing), leading to lack of strength degradation post-peak and possible underestimations of energy dissipation (as discussed by Graziotti et al. [57]). Similarly, the lack of residual friction in the constitutive model – the effects of which are typically neglected in discontinuum modelling [101,51] – can also fail to capture reduction in lateral strength. Experimental testing for saw cut dry-joint URM (modelled herein) also shows that there is negligible difference between initial and residual friction [54,82]. DEM-1, in contrast, reproduced reduction of force capacity due to post-peak unit cracking and shear failure using the author's coupled fracture energy-based contact constitutive model. From experimental tests, the walls demonstrated three primary behaviours: (i) de-bonding with the RC beam and (ii) diagonal stepped cracking originating from the top right corner which causes (iii) crushing of the masonry units at the wall toe. Lourenço et al. [81] also conclude that IP shear-compression failure for dry-joint URM walls involves primarily diagonal cracking with additional bend joint slippage – this mechanism is well reproduced in all numerical models (Fig. 3). Comparison of PyBullet, DEM-1 [109], DEM-2 [15], FEM [81], and experimental [81] deformed shapes for SW walls (AR = 1.0) (Fig. 3). PyBullet simulations were also limited in representing de-bonding with the RC beam (which was captured in DEM-2) due to the lack of tensile strength in the contact model (a fixed boundary condition represented the mortar layer between the top course and the RC beam). Displaced shapes in PyBullet also show OOP rotations for some blocks near the wall perimeter (e.g., see top left of SW250 displaced shape). These spurious torsional movements are believed to be due to the summation of numerical errors from constraint enforcement/relaxation and/or underprediction of friction forces (due to its coupling in the contact normal direction).

The second set of walls ("WS" specimens) simulated in PyBullet were experimentally tested by Vasconcelos [132]. All walls were 1000 mm × 200 mm × 1200 mm (AR = 1.2) and constructed using dry-joint granite blocks with dimensions 200 mm × 200 mm × 150 mm and density $\rho = 2600$ kg/m^3. Vertical pre-compression loads 40 kN, 100 kN,

Table 1
PyBullet simulation parameters for IP shear-compression models.

*FPS**	*Sub-steps*	*Time-step**	*Solver iterations*	*Contact stiffness*	*Contact damping*	*ERP*	*Coefficient of restitution*
[*fps*]	[–]	[*s*]	[–]	[*N/m*]	[*Ns/m*]	[–]	[–]
1000 or 2000	0	0.0005 or 0.001	10	10^9	10	0.95	0

* SW walls (AR = 1.0) used 2000 fps (i.e., $\Delta t = 0.0005s$) and WS walls (AR = 1.2) used 1000 fps (i.e., $\Delta t = 0.001s$)

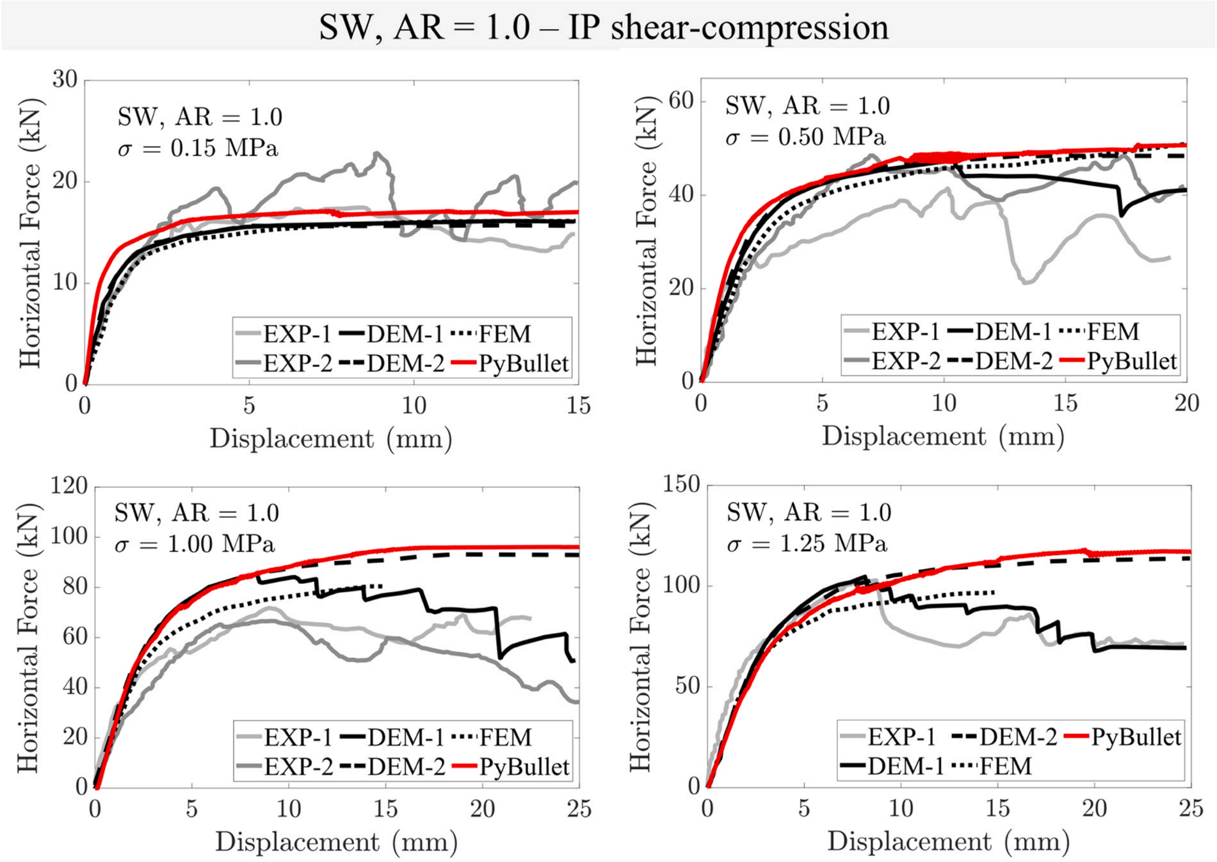


Fig. 2. Comparison of PyBullet, DEM-1 [109], DEM-2 [15], FEM [81], and experimental [81] force-displacement outputs for SW walls (AR = 1.0).

175 kN, and 250 kN (compressive normal stresses of $\sigma = 0.2$, 0.5, 0.875, and 1.25 MPa, respectively) were applied to a steel beam placed on top of the wall. As for the previous walls, the overburden pressure was implemented in PyBullet by assigning an equivalent density to a steel beam with dimensions 1100 mm × 250 mm × 150 mm, which was similarly restrained OOP using fixed guide rails. In the experimental setup, a layer of masonry units lined the bottom and top of the wall and were fixed with respect to the base and the steel beam, respectively, using steel clamps. This was replicated in PyBullet using two 1000 mm × 200 mm × 75 mm masonry blocks above and below the wall. A fixed constraint (using *p.createConstraint*) was applied between the steel beam and the top block. The bottom block was assigned zero mass, essentially rendering the block fixed (this provided more numerical stability than using an additional constraint). All masonry-masonry interfaces had a friction coefficient of μ = 0.65 (obtained from single dry-joint shear--compression experimental tests) and zero cohesion, tensile strength (f_t), and dilatancy. The horizontal load was similarly applied using a two-block rigid body actuator at a quasi-static displacement rate of 1 mm/s.

For a comparable discontinuum numerical assessment, the authors conducted DEM simulations of the WS walls with normal and shear stiffnesses of $k_n =$ 5 MPa/mm and $k_s =$ 2 MPa/mm – all other interface properties aligned with those used in PyBullet (residual friction was not considered). The pre-compression load was applied as a stress on the top face of the RC beam, which was displacement controlled using an applied velocity of 0.5 mm/s. Blocks were triangulated using a maximum edge length of 25 mm. The small-strain numerical mode was enabled, and the convergence ratio criteria was set to 1e-6. Computational time was evaluated using the *time.kernel* FISH function.

While force capacities for all four walls are aligned with those from experimental tests (+0.33 to +2.79 %), PyBullet simulations underpredicted experimental wall stiffnesses for 100 kN, 175 kN, and 250 kN compression loads (up to −61 %) (Fig. 4) – likewise using DEM (up to −45.8 %). The prediction of constant initial stiffness with increased vertical pressure by both numerical models can be attributed to rigid block assumptions (lack masonry unit deformations and associated redistribution of internal stresses and increased confinement). The coupling of shear and normal contact forces in the PyBullet friction model is also believed to contribute to underestimations of shear capacity (and thus initial stiffness). PyBullet also does not have a shear spring in its contact model – resulting in a friction dominated shear behaviour. This friction (slip/no-slip) model is very similar to that of NSCD and primarily dictates shear strength (i.e., friction impulse magnitude), leading to premature joint slipping. Similar phenomena were also observed in AEM models (albeit of mortared URM) by Malomo et al. [92]. Normal contact stiffness was chosen to align with those used in the SW wall tests and intentionally not calibrated iteratively. Although this causes bed joints to slip pre-emptively – resulting in early onset of diagonal stepped cracking propagating from the top right joint – force capacities are aligned between PyBullet, DEM, and experimental results (maximum of +9.1 % difference with experimental). This is attributed to diagonal cracking leading to rocking failure hinging at the wall toe – the resultant moment of which is primarily governed by the applied horizontal and pre-compression loads, not the contact normal forces between the joints. This is in agreement with failure mechanisms observed in the experimental tests with the exception of WS175 which experienced toe crushing. Individual unit compressive failures are unaccounted for in PyBullet and DEM models due to rigid body assumptions as previously addressed. Like in the SW walls, some units exhibit OOP slipping (for interior blocks) or rotations (for boundary blocks). While this is unexpected for idealized numerical models using perfectly isotropic Coulomb friction, it is representative of observed experimental behaviours (e.g., irregularities in the stone causing them to become “loose” and displace OOP during testing) [133].

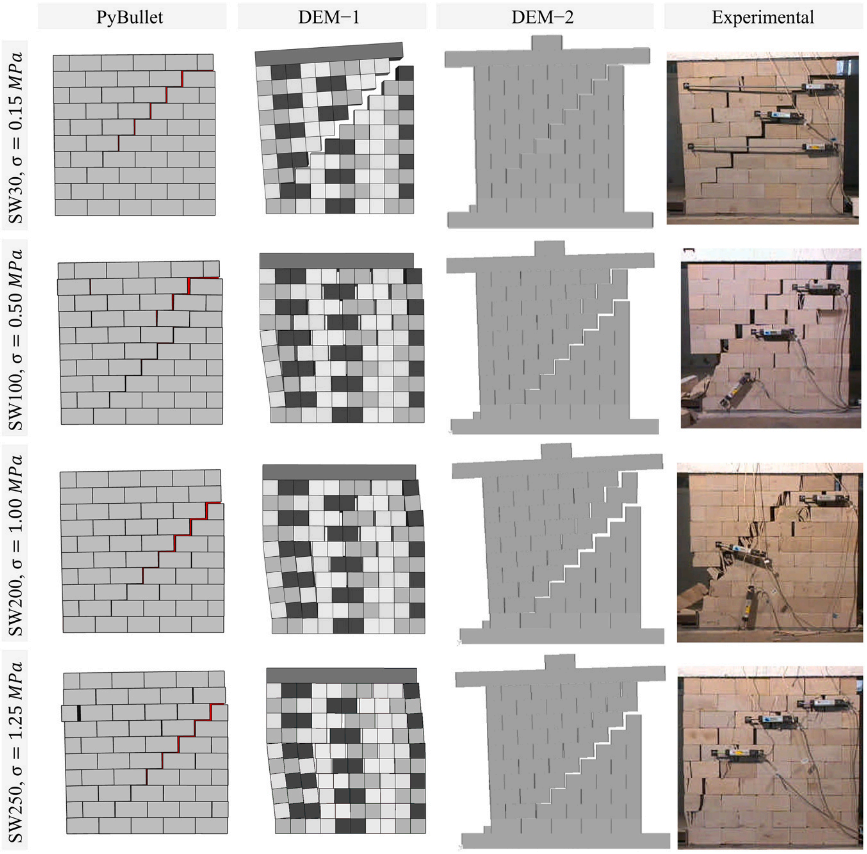


Fig. 3. Comparison of PyBullet, DEM-1 [109], DEM-2 [15], FEM [81], and experimental [81] deformed shapes for SW walls (AR = 1.0).

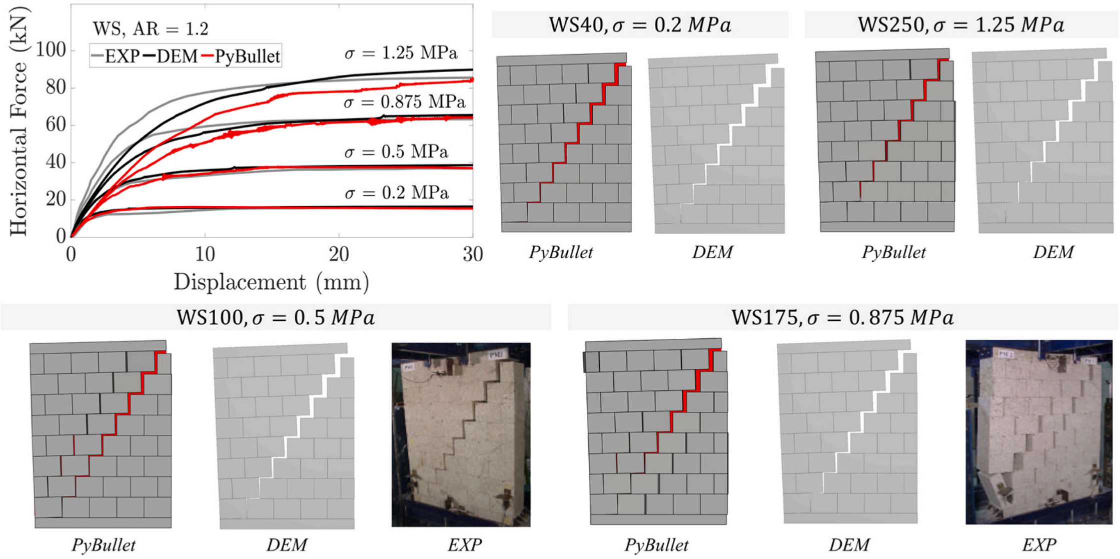


Fig. 4. Comparison of PyBullet, DEM, and experimental [132] force-displacement outputs and deformed shapes for WS walls (AR = 1.2).

Table 2
Comparison of computational times for IP shear-compression simulations from DEM (present study) and PyBullet.

Specimen		[–]	*SW30*	*SW100*	*SW200*	*SW250*	*WS40*	*WS100*	*WS175*	*WS250*
No. rigid bodies		[–]	61	61	61	61	48	48	48	48
Execution time	*PyBullet*	[*min*]	2.49	5.05	4.57	5.11	2.63	2.67	2.62	2.68
	DEM	[*min*]	-	-	-	-	5.56	5.67	5.14	5.52

The SW and WS IP shear-compression PyBullet models required at most ~5 min of execution time (inclusive of simulation setup and data output) (Table 2). Respective times for DEM simulations in this study are also reported to give the reader a general idea of computational demand for comparisons with PyBullet (chosen parameters such as displacement rate and number of sub-contacts will influence DEM efficiency). As reasonably expected, PyBullet execution times appear directly proportional to the total number of time-steps (order of magnitude of 10^3 for IP shear-compression simulations) and quantity of rigid bodies [60] – demonstrated by the SW100, SW200, and SW250 walls. The number of contact points – which can rapidly accumulate with stacked assemblies/structures (e.g., walls) – will also contribute to computational demand (i.e., more rows in Eq. (5) possibly leading to ill-conditioned matrices). Excessively stiff systems (e.g., high compressive loads and/or contact stiffness) also require reducing time-step size (and consequently increasing computations) for numerical stability (likewise in DEM when using the Courant-Friedrichs-Lewy, or CFL, condition) [27]. An overly large time-step may also lead to numerical issues in stiff systems – some of which can be visualized in the GUI (e.g., block “jitters”, interpenetrations). Further investigations are warranted to better inform selection of time-step size and their effect on numerical outputs for stiff rigid body systems present in URM structural analysis (e.g., see [88] for such studies on soft bodies and fluids).

4.2. Walls under in-plane settlements

Two tuff masonry T-panels (with and without interlocking – herein referred to as the S1 and S2 panels, respectively) undergoing IP settlements were simulated in PyBullet and compared to experimental tests [49] and two 2D numerical Limit Analysis rigid block models proposed by Portioli and Cascini [105] and Gagliardo et al. [49]. Selected physics engine parameters shown in Table 3 were chosen to align with settings from the IP shear-compression simulations. Each T-panel was constructed using blocks of dimensions 100 mm × 100 mm × 50 mm with a volumetric weight of 12.5 kN/mm^3. The IP face (i.e., web) of the panel was 800 mm × 100 mm × 300 mm (i.e., 8 × 1 × 7 blocks), and the orthogonal face (i.e., flange) of the panel was 300 mm in depth (i.e., 1 × 3 × 7 blocks). The right half of the panel rested on a fixed slab of timber, while the left half rested on a support layer of masonry and timber that was fixed with respect to the lifting table. The lifting table was replicated in PyBullet analogous to the actuator multi-body from IP shear-compression simulations – the “driver” plate had dimensions of 500 mm × 200 mm × 200 mm, a mass of 1 kg, and displaced downwards at a rate of 0.5 mm/s. Block interfaces were assigned a friction coefficient μ =0.72 – similarly chosen by the rigid block models. The vertical support force was calculated as the sum of the global Z-components of the contact normal forces between the wall and the support layer of masonry and timber.

DEM joint models for IP settlement tests used normal and shear stiffnesses of $k_n =$ 1 MPa/mm and $k_s =$ 0.4 MPa/mm – all other interface properties were consistent with those used in PyBullet (residual friction was not considered). Blocks used a local damping constant of 0.8 and were triangulated using a maximum edge length of 50 mm. Large-strain numerical mode was enable with the convergence ratio criteria was set to 1e-10. The lifting table displaced downwards at a velocity of 0.5 mm/s.

Similar failure mechanisms (Fig. 5) for both walls are exhibited by PyBullet, DEM, the 2D limit analysis model proposed by Gagliardo et al. [49], and the experimental tests (also [49]). In PyBullet and DEM, however, rotational hinges appear in slightly different locations for the S2 wall (occurring at lower courses), which can be attributed to the 3D effects of interlocking. A secondary rotational hinge also develops at the interface between the flange and web for the S1 wall in experimental testing, DEM, and PyBullet simulations. Furthermore, diagonal stepped cracking patterns from PyBullet are more closely aligned with experimental tests compared to those from 2D limit analysis. DEM models also appropriately replicate these cracking patterns, albeit with unexpected horizontal separation between blocks. Experimentally observed OOP rotations at the top of the wall were also notably present in PyBullet and DEM simulations but not limit analysis (the latter restricted by 2D geometry assumptions). Force-displacement outputs from all numerical models and experimental tests predict the same overall response (see Fig. 6): the support reaction initially decreases until complete formation of the failure mechanism around $\delta = 10$ mm and is followed by gradually increasing support forces (attributed to large displacement effects) [49] until collapse. While ultimate displacement from PyBullet is in agreement with those from the limit analysis model (+2.0 %), DEM (-8.7 %), and the three experimental tests (+3.2 %) for the S1 wall, PyBullet underestimated experimental displacement capacity for the S2 wall (-21.40 %) (Table 4). Alternatively, DEM overestimated experimental displacement capacity for both walls (+12.2 % and +8.6 %, for S1 and S2, respectively). Differences can be associated to interlocking mechanisms – premature slipping (possibly due to errors with contact normal and friction forces, which are influenced by the chosen contact stiffness) at the interface between the flange and web can induce overturning collapse in the orthogonal wall. Additionally, PyBullet and DEM computational times significantly exceed those of limit analysis, although this can potentially be linked to additional rigid bodies, 2D vs. 3D contact formulations (the latter requiring more computations), use of different PC’s (in the case of limit analysis), and incremental methods (i.e., time vs. displacement-based discretization schemes and number of increments simulated – e.g., ~300,000 time-steps simulated in both PyBullet and DEM). PyBullet also required ~20 additional minutes of computational time compared to DEM despite simulating a comparable number of time-steps. The increased computational time is believed to be due to the implementation of the boundary conditions for the support layer of masonry and timber and the lifting table. In DEM, this is implemented by joining the blocks together (*block join* command). In PyBullet, each block is individually assigned a fixed constraint to the lifting table – resulting in additional constraint equations that may slow down the solver.

5. Out-of-plane failure analysis

Selected URM specimens from scaled (1:5) experimental OOP tilting tests by Restrepo Vélez et al. [112] were replicated using PyBullet and compared with DEM simulations by the authors and from Bui et al. [15] – hereinafter referred to as DEM-1 and DEM-2, respectively. Detailed configurations, dimensions, and naming conventions for constructions used herein can be found in Restrepo-Vélez et al. (2014). All wall assemblies were composed of masonry blocks with dimensions 80 mm × 40 mm × 30 mm and density 2680 kg/m^3. Wood components (when present) had a density of 160 kg/m^3. Block interfaces used a friction coefficient of $\mu =$ 0.67 (the lower bound from experimental results similarly used by [15]), while masonry-wood interfaces $\mu =$ 0.40 (assumed). The same physics engine parameters from the IP tests were selected for OOP simulations for consistency (Table 5) – exceptionally, specimen S42 (two-storey building) required 3 sub-steps for numerical

Table 3
PyBullet simulation parameters for IP settlement models.

FPS	*Sub-steps*	*Time-step*	*Solver iterations*	*Contact stiffness*	*Contact damping*	*ERP*	*Coefficient of restitution*
[*fps*]	[–]	[*s*]	[–]	[*N/m*]	[*Ns/m*]	[–]	[–]
1000	0	0.001	10	10^9	10	0.95	0

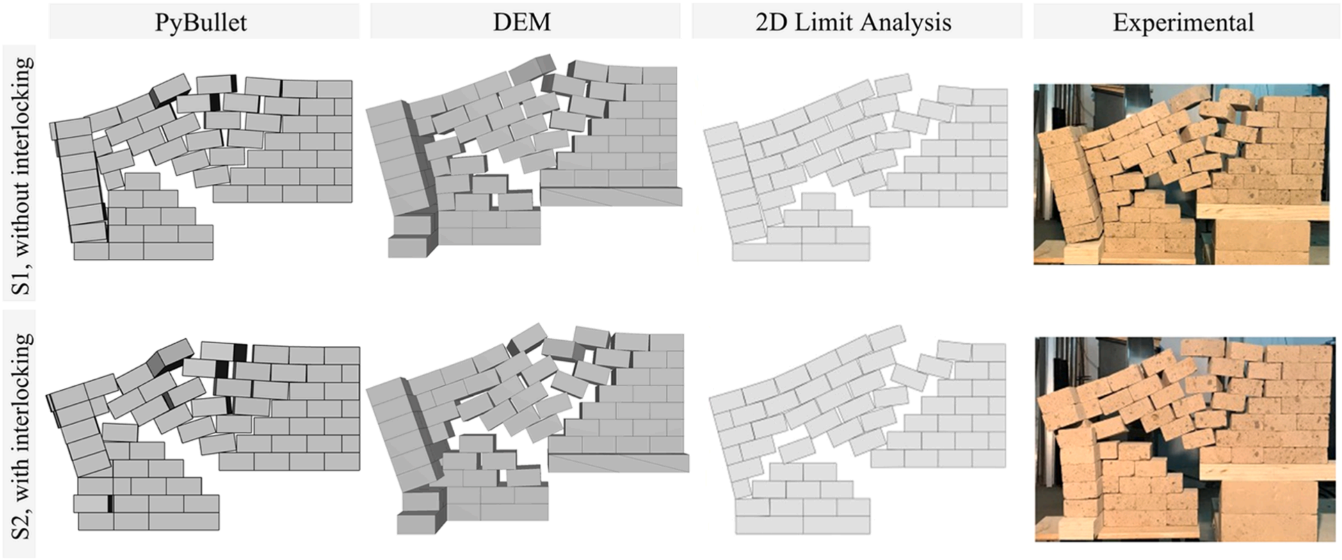


Fig. 5. Comparison of IP settlement displaced shapes from PyBullet, DEM, 2D Limit Analysis model [49], and experimental tests [49].

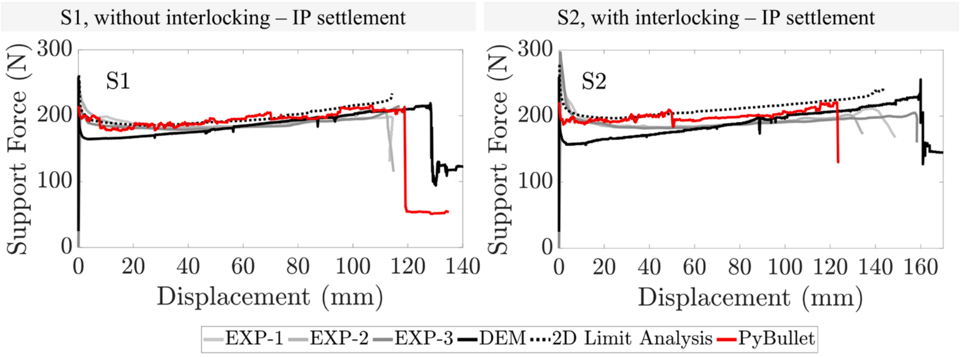


Fig. 6. Comparison of IP settlement force-displacement outputs from PyBullet, DEM, 2D Limit Analysis [49], and experimental tests [49].

Table 4
Comparison of numerical and experimental IP settlement results output from PyBullet, Limit Analysis [49], and experimental tests [49].

		PyBullet		*DEM*		*2D Limit Analysis* [49]		*Experimental** *Gagliardo et al.,* [49]	
Specimen	[–]	*S1*	*S2*	*S1*	*S2*	*S1*	*S2*	*S1*	*S2*
No. rigid bodies	[–]	78	78	78	78	61	61	-	-
Support Force at $\delta = 0$ mm	[*N*]	210.9	221.1	258.3	258.3	260.8	277.4	231.3	271.8
Support Force at $\delta = 10$ mm	[*N*]	196.7	190.0	165.8	158.5	189.5	200.3	192.5	194.8
Ultimate Displacement, δ_u	[*mm*]	118.8	123.0	129.1	161.0	116.5	145.0	115.1	148.2
Execution time	[*min*]	76.5	107.2	63.9	84.8	1.92	2.67	-	-

[*] Experimental values are averaged across all three tests

Table 5
PyBullet simulation parameters for OOP tilting models.

FPS	*Sub-steps**	*Time-step**	*Solver iterations*	*Contact stiffness*	*Contact damping*	*ERP*	*Coefficient of restitution*
[*fps*]	[–]	[*s*]	[–]	[*N/m*]	[*Ns/m*]	[–]	[–]
1000	0 or 3	0.00025 or 0.001	10	10^9	10	0.95	0

[*] Specimen S42 used 3 sub-steps for stability (i.e., $\Delta t = 0.00025s$)

stability. The tilting apparatus used for tests was replicated in PyBullet using the *p.createMultiBody* function to join two rigid plates (with mass of 1 kg and thickness of 10 mm) at the edge with a hinge constraint (i.e., rotation OOP is only degree of freedom). A narrow support 15 mm in height spanned the length of the tilting apparatus and was fixed with respect to the top tilting plate to prevent the structure from sliding off. For computational efficiency, the apparatus was rotated OOP at a rate of 1 degree/s until near collapse (1–2 degrees prior to expected collapse), where the rotation rate was slowed to the experimental rate of 0.045 degree/s. Critical angle at collapse (θ_{cr}) – herein defined as a local or global severe damage limit state or post-failure condition when (i) components lose their ability to withstand vertical load and (ii) units or components/assemblies dislocate and start interacting dynamically [56] – was determined through visual observation and obtained using the *p.getBasePositionandOrientation* command to query the tilting angle of the apparatus.

Analogous to simulations in Section 4, DEM-1 OOP tilting simulations used Mohr-Coulomb joint models with normal and shear stiffnesses of $k_n = \quad 1$ MPa/mm and $k_s = \quad 0.4$ MPa/mm, a friction coefficient of $\mu = 0.67$, and zero cohesion and tensile strength – these mechanical properties were applied to all block interfaces. Block sub-contacts were increased using *block face triangulate radial-8* (adds a center vertex and mid-edge vertices to the block face). Using the large-strain numerical formulation, the model was executed for 10,000 cycles (i.e., time-steps) while applying a rotational velocity of 0.57 deg/s to the centroid of the tilting table. The rotational velocity was then removed and the model solved in static equilibrium with a convergence ratio criterion of 1e-6. This process was repeated until static equilibrium was no longer achievable under the given criteria. The angle of tilt at this point in the simulation was defined as the critical angle of collapse. It is noted that not all the 3DEC parameters discussed herein are likewise reported by Bui et al. [15] for their OOP tilting models.

5.1. Assemblies under out-of-plane tilting

Three types of URM assemblies with varying length-to-height (L/H) ratios (Table 6) were modelled: (1) interior collapse C-walls, (2) exterior collapse C-walls, and (3) exterior collapse C-walls with partitions and lintel supported openings. The interior collapse C-walls (Fig. 7) exhibited expected collapse mechanism G – defined by D'Ayala and Speranza [33] as the central trapezoidal portion of the planar face (e.g., façade or interior wall, in this case) displacing and rotating outwards. Exterior collapse C-walls (Fig. 8) demonstrated hybrid collapse mechanism A-B2 – primary façade overturning at a horizontal hinge around the mid-height coupled with secondary diagonal cracking and overturning of portions of the orthogonal walls (due to insufficient connections). PyBullet collapse progressions also display marginal inward displacement of the orthogonal walls, possibly induced by the primary collapse of the planar face. While all PyBullet models obtained critical angles inversely proportional to L/H (as expected), they underestimated experimental results (up to −23.9 % and −22.6 % for interior and exterior collapse C-walls, respectively) – likewise with DEM-1, (up to −14.5 % and −17.5 %, respectively). By contrast, DEM-2 generally overestimated experimental critical angles (up to +15.32 %) for interior walls and underestimated (as much as −9.25 %) for exterior walls. Simulations of S22 (C-wall with lintel supported openings and an interior partition) (Fig. 9), reproduced a B2 collapse mechanism – i.e., simultaneous overturning of the façade and portions of the orthogonal walls. This is likely attributed to stress concentrations near the openings and loss of frictional resistance in the lintels, leading to IP shear-compression failure (e.g., diagonal cracking) in the orthogonal walls and loss of connection with the façade. Spurious rotations of blocks near the structure base are also observed, possibly due to numerical stability errors in stacked assemblies as previously discussed. Critical angle was also underpredicted in PyBullet (-31.60 %), a conservative estimate compared to DEM-1 and DEM-2 (-11.9 % and −5.39 % error with experimental values, respectively). Overall OOP collapse mechanisms are faithfully replicated for URM assemblies – quantitative discrepancies and premature collapse are accredited to excessive block slipping as previously examined (i.e., mechanical effect of contact stiffness, coupling of contact normal and friction forces). Computational times for URM assemblies are summarized in Table 7 and further discussed in Section 5.2.

5.2. Buildings under out-of-plane tilting

A two-storey URM building (S42) of dimensions 800 mm × 1040 mm × 1260 mm was also simulated under OOP tilting. Four doors and four windows were present at the ground level and upper level, respectively, and supported with wood lintels. Two flexible slab systems were each represented with a set of 8 wood joists (dimensions 1000 mm × 20 mm × 30 mm) oriented perpendicular to the tilting axis and free to move axially. Two vertical reactions of 16.46 N were applied at both ends of each joist in the form of an equivalent density block (dimensions 180 mm × 20 mm × 30 mm) that was fixed with respect to each joist. All three numerical models underpredicted critical angle of specimen S42 (−31.7 %, −23.0 %, and −16.5 % for PyBullet, DEM-1, and DEM-2, respectively). However, expected collapse mechanism B2 was well replicated (Fig. 10), with the horizontal façade overturning hinge developing between the two storeys (although this contrasts with the asymmetrical rotation of the façade in the experimental test). Diagonal shear cracking through the windows closest to the tilting axis was also prominent across all numerical models and experimental testing. Lack of shear resistance in the lintels (similarly exhibited by specimen S22) and excessive slipping at the bed joints (likewise observed in the IP simulations) likely contribute to conservative prediction of collapse. Although execution times for URM assemblies under OOP tilting (for the specified number of time-steps) did not exceed 20 min (Table 7) (up to 7 times faster than DEM-1), specimen S42 required 5.25 h. However, it's noted that the simulation consisted of significantly more time-steps (order of magnitude 10^6 compared to 10^4 for assemblies) due to additional sub-steps required for numerical stability and over double the amount (1829) of rigid bodies, which is associated with an increase of contact points. As previously discussed, computational time in PyBullet (and DEM) is sensitive to selected inputs (e.g., displacement rate, number of sub-contacts, sub-steps, solver) – further investigation is needed to determine the optimal parameters to balance numerical efficiency and mechanical reproducibility. Likewise, additional modifications to the contact model (e.g., alternative friction models, customizations to the source code) are currently being explored to address quantitative incongruities. Despite these limitations, this complete building simulation establishes PyBullet's capability to simultaneously simulate IP and OOP behaviour beyond near collapse at an individual block level with reasonable execution times compared to established discontinuum methods.

6. Conclusions

This study is the first to quantitatively evaluate physics engines for rigorous URM structural analysis, specifically for quasi-static,

Table 6
Comparison of critical angle $\boldsymbol{\theta_{cr}}$ at OOP collapse obtained from PyBullet, DEM-1, DEM-2 [15], and experimental tests [112] for all specimens.

Specimen		*S1*	*S2*	*S3*	*S5*	*S6*	*S7*	*S8*	*S9*	*S10*	*S22*	*S42*
Aspect ratio, L/H		1.5	1.5	1.5	1.09	1.77	1.088	0.816	0.816	1.633	-	-
Critical angle, θ_{cr} [°]	*PyBullet*	10.85	-	-	15.41	9.20	12.56	16.16	-	9.79	7.62	9.07
	DEM−1	12.37	-	-	16.46	10.27	13.91	16.41	-	10.21	9.81	10.23
	DEM−2	14.68	-	-	18.11	12.30	16.01	18.06	-	12.46	10.54	11.09
	Experimental	14.25	12.73	13.71	19.24	11.75	16.22	19.90	19.39	12.02	11.14	13.28

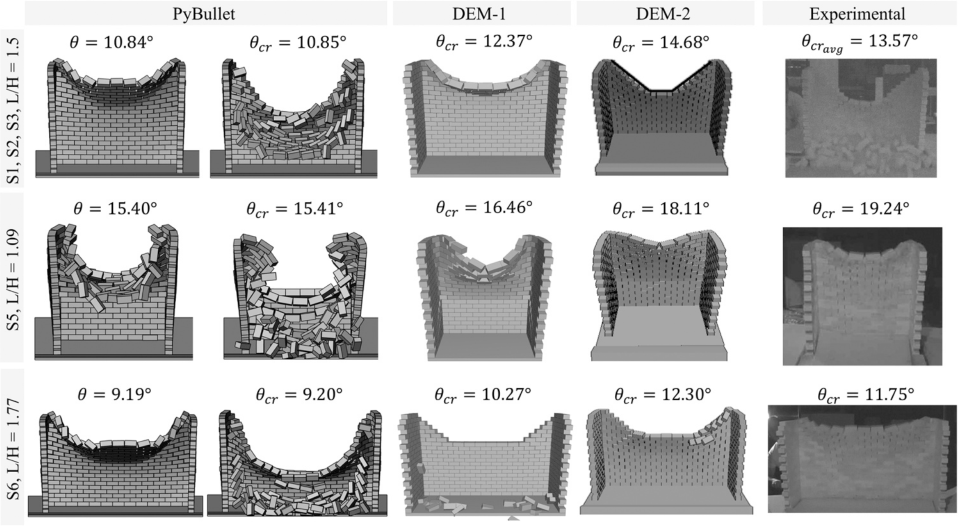


Fig. 7. Observed collapse mechanisms in PyBullet (progressive), DEM-1, DEM-2 [15], and experimental tests [112] for interior collapse C-walls (specimens S1, 2, 3, 5, 6).

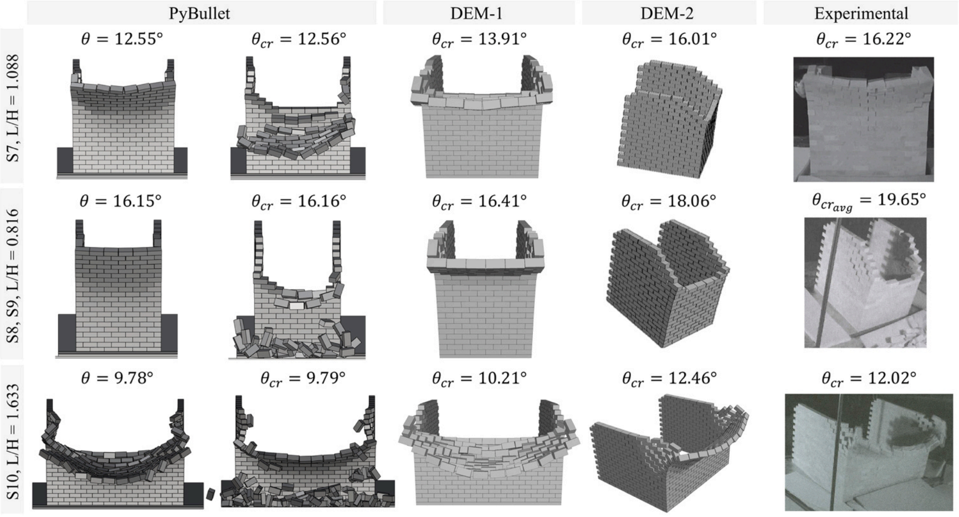


Fig. 8. Observed collapse mechanisms in PyBullet (progressive), DEM-1, DEM-2 [15], and experimental tests [112] for exterior collapse C-walls (specimens S7, 8, 9, 10).

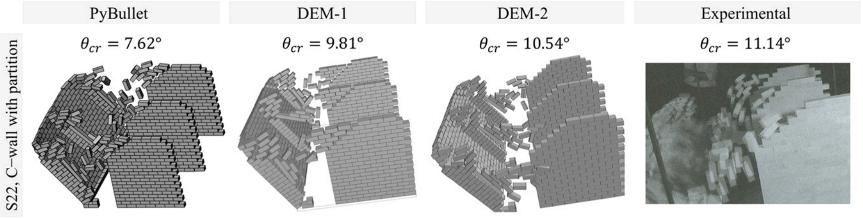


Fig. 9. Observed collapse mechanisms in PyBullet (progressive), DEM-1, DEM-2 [15], and experimental tests [112] for specimen S22 (C-wall with partition).

Table 7
Comparison of computational times for OOP tilting simulations from DEM-1 and PyBullet.

Specimen		[–]	*S1,2,3*	*S5*	*S6*	*S7*	*S8,9*	*S10*	*S22*	*S42*
No. rigid bodies		[–]	390	327	429	453	411	663	849	1829
Execution time	*PyBullet*	[*min*]	6.61	7.11	10.06	15.42	16.10	11.77	19.51	315.47
	DEM−1	[*min*]	31.14	49.44	22.72	27.28	39.38	49.07	39.64	103.77

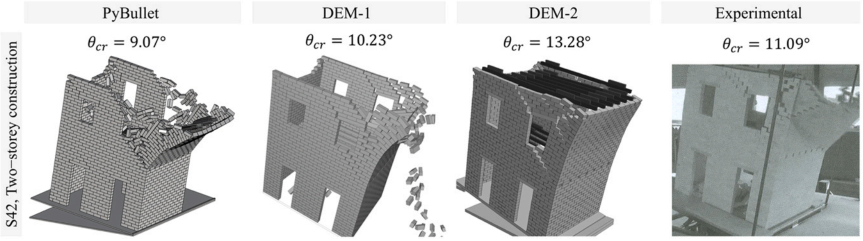


Fig. 10. Observed collapse mechanisms in PyBullet (progressive), DEM-1, DEM-2 [15], and experimental tests [112] for specimen S42 (two-story building).

monotonic in-plane (IP) shear-compression, IP settlements, and out-of-plane (OOP) tilting for meso-scale dry-joint URM assemblies and buildings. This novel adaptation demonstrates the potential of physics engines to be used in engineering contexts that are different from their original intended applications in virtual animations, breaking boundaries. PyBullet and its underlying software Bullet Physics (used in this paper), demonstrate potential to become alternative discontinuum modelling techniques that can address shortcomings of existing approaches (e.g., prohibitive analysis time, accessibility for practitioners and applied researchers). This is substantiated by the following findings from this study:

- Quasi-static response of dry-joint URM walls, loaded monotonically, is captured reliably using PyBullet's built-in contact model, as supported by quantitative discussions presented herein.
- Force capacities for IP shear-compression simulations on walls were in agreement with those obtained from other numerical models (i.e., DEM). Respective stiffnesses were aligned with experimental and numerical results for the AR = 1.0 walls but underestimated for the AR = 1.2 walls. Crack propagation and patterns matched observations from previous numerical and experimental outputs. These results were achieved under ~5 min of simulation time per wall.
- IP settlement simulations exhibited the expected force-displacement response: decreasing support force until development of failure mechanism followed by a period of increasing support force until collapse. PyBullet simulations notably reproduced 3D effects (e.g., OOP rotations) – a limitation of 2D rigid block models (which understandably assume plane stress for an ideally plane problem) – and failure mechanisms (e.g., diagonal stepped cracking patterns and orthogonal wall overturning) observed in experimental tests. Vertical support reactions and ultimate displacements are reasonably matched with those from numerical and experimental results.
- OOP tilting tests for assemblies and buildings simulated expected collapse mechanisms corroborated from DEM and experimental results, albeit, with conservative predictions (as much as −31.70 %) for critical angle at collapse. Computational times ranged from around 6 min to ~20 min for assemblies.

This exploratory study takes the first step to investigate URM structural analysis applications using PyBullet. However, PyBullet's capabilities for URM structural analysis – at the time of publication – are limited due to the default contact model and associated assumptions on block interactions. As such, the authors of this study also acknowledge additional shortcomings of the presented results:

- Complex loading schemes such as dynamic and cyclic testing have yet to be explored using PyBullet. Findings from subsequent studies will provide deeper insights into the capabilities and limitations of applications of physics engines for advanced discontinuum structural analysis (e.g., seismic collapse analysis, full-scale debris assessment).
- Underestimations of coupled contact normal and friction forces can lead to premature block slipping, resulting in conservative predictions for failure mechanisms that are dictated by bed joint shear strength. While parameters (e.g., solver iterations, contact stiffness and damping) were deliberately not calibrated to address this discrepancy, further investigation into their relationship with mechanical and material properties can appropriately inform input selection. Alternative friction models (e.g., box and polyhedral approximations, bristle model) may also improve representation of friction phenomena and associated shear resistance – this is similarly proposed by discontinuum dry-joint studies (as discussed in Section 2).
- The default contact model in PyBullet is only representative of dry-joint block interfaces. As such, modifications to the physics engine source code are required to capture mortar joint behaviours (e.g., cohesion, tensile strength, shear stiffness, dilatancy). Rigid body assumptions also prevent reproducing individual unit failures. While physics engines are capable of modelling soft bodies, this was not explored in the present study. Additional advancements are necessary for PyBullet (and other physics engines) to become reliable structural analysis tools (à la DEM, AEM, FEM, etc.) for advanced URM systems (e.g., reinforced URM, retrofitted systems, mortared joints).
- Simulations of URM assemblies required reasonable execution times; however, building-scale constructions demanded significantly more computations. The selected time-step – while chosen for mechanical authenticity and numerical stability – is also significantly smaller than those used in traditional physics engine applications (and therefore may not be compatible for real-time applications). Additional improvements (e.g., GPU computing resources, alternative solvers and numerical methods) can accelerate computations.

CRediT authorship contribution statement

A. Wang: Formal analysis, Methodology, Investigation, Conceptualization, Validation, Visualization, Data curation, Writing – original draft, Writing review & editing. **B. Pulatsu:** Methodology, Investigation, Conceptualization, Supervision, Writing – review & editing. **S. Andrews:** Methodology, Investigation, Conceptualization, Supervision, Writing – review & editing. **D. Malomo:** Methodology, Investigation, Conceptualization, Funding acquisition, Supervision, Project administration, Writing original draft, Writing – review & editing.

Acknowledgements

The authors acknowledge funding from the Government of Canada's New Frontiers in Research Fund (NFRF) [Ref. No. NFRFE-2022–00312]. The work of the first author was also supported by the Fonds de recherche du Québec Nature et Technologies (FRQNT), through the B2X scholarship program. The authors would like to thank the reviewers for their time reading the manuscript – their comments have significantly contributed to the improvement of the quality of the study. The first author would like to thank Dr. E. Giordano and L. Davis for helping in gathering some of the experimental data for this study, and J. Belanger for her support during the analysis and writing process. The first author is also grateful to M. Wang and S. Wang, for their constant support during the long nights of analysis and post-processing that this work has entailed. This paper is dedicated to the loving memory of S. Wang.

Data availability

Data will be made available on request.

References

[1] Abeling S, Ingham JM. Volume loss fatality model for as-built and retrofitted clay brick unreinforced masonry buildings damaged in the 2010/11 Canterbury earthquakes. Structures 2020;24:940–54. https://doi.org/10.1016/j.istruc.2020.02.014.

[2] Afif M, Said Y, Atri M. Computer vision algorithms acceleration using graphic processors NVIDIA CUDA. Clust Comput 2020;23:3335–47. https://doi.org/10.1007/s10586-020-03090-6.

[3] Andrews S., Erleben K., Ferguson Z. (2022) Contact and friction simulation for computer graphics. In: Proceedings - SIGGRAPH 2022 Courses. Association for Computing Machinery, Inc.

[4] Anitescu M, Potra FA. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. Nonlinear Dyn 1997;14:231–47. https://doi.org/10.1023/A:1008292328909.

[5] Baraff D. (1994) Fast contact force computation for nonpenetrating rigid bodies. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1994. Association for Computing Machinery, New York, NY, USA, pp 23–34.

[6] Baumgarte J. Stabilization of constraints and integrals of motion in dynamical systems. Comput Methods Appl Mech Eng 1972;1:1–16. https://doi.org/10.1016/0045-7825(72)90018-7.

[7] Beatini V, Royer-Carfagni G, Tasora A. A regularized non-smooth contact dynamics approach for architectural masonry structures. Comput Struct 2017;187:88–100. https://doi.org/10.1016/j.compstruc.2017.02.002.

[8] Beatini V, Royer-Carfagni G, Tasora A. Modeling the shear failure of segmental arches. Int J Solids Struct 2019;158:21–39. https://doi.org/10.1016/j.ijsolstr.2018.08.023.

[9] Bello RA, Günaydin M, Altunişik AC. Structural collapse visualization using blender and BCB. In: Mosallam AS, El Bhiri B, Karbhari VM, Saadeh S, editors. Sustainable Civil Infrastructures. Cham: Springer Nature Switzerland; 2023. p. 163–72.

[10] Berton J.A., Chuang K.-L. (2016) Effects of Very High Frame Rate Display in Narrative CGI Animation. In: 2016 20th International Conference Information Visualisation (IV). pp 395–398.

[11] Bianchini N, Mendes N, Calderini C, Lourenço PB. Modelling of the dynamic response of a reduced scale dry joints groin vault. J Build Eng 2023;66:105826. https://doi.org/10.1016/j.jobe.2023.105826.

[12] Birgin EG, Martínez JM, Raydan M. Nonmonotone spectral projected gradient methods on convex sets. SIAM J Optim 2000;10:1196–211. https://doi.org/10.1137/S1052623497330963.

[13] Boni C, Ferretti D, Lenticchia E. Effects of brick pattern on the static behavior of masonry vaults. Int J Arch Herit 2022;16:1199–219. https://doi.org/10.1080/15583058.2021.1874565.

[14] Bouckaert I., Godio M., de Almeida J.P. (2022) A strategy for generating pushover curves of block assemblies including post-peak branch using the discrete element method. In: 3rd EUROPEAN CONFERENCE ON EARTHQUAKE ENGINEERING & SEISMOLOGY.

[15] Bui TT, Limam A, Sarhosis V, Hjiaj M. Discrete element modelling of the in-plane and out-of-plane behaviour of dry-joint masonry wall constructions. Eng Struct 2017;136:277–94. https://doi.org/10.1016/j.engstruct.2017.01.020.

[16] Calvi GM, Moratti M, O'Reilly GJ, et al. Once upon a time in Italy: the tale of the Morandi Bridge. Struct Eng Int 2019;29:198–217. https://doi.org/10.1080/10168664.2018.1558033.

[17] Casapulla C, Portioli F. Experimental and analytical investigation on the frictional contact behavior of 3D masonry block assemblages. Constr Build Mater 2015;78:126–43. https://doi.org/10.1016/j.conbuildmat.2014.12.100.

[18] Castellazzi G, Pantò B, Occhipinti G, et al. A comparative study on a complex URM building: part II—issues on modelling and seismic analysis through continuum and discrete-macroelement models. Bull Earthq Eng 2022;20:2159–85. https://doi.org/10.1007/s10518-021-01147-4.

[19] Cattari S, Camilletti D, D'Altri AM, Lagomarsino S. On the use of continuum Finite Element and Equivalent Frame models for the seismic assessment of masonry walls. J Build Eng 2021;43:102519. https://doi.org/10.1016/j.jobe.2021.102519.

[20] Catto E. (2011) Soft constraints reinventing the spring. In: Game Developer Conference (GDC).

[21] Chen SY, Moon FL, Yi T. A macroelement for the nonlinear analysis of in-plane unreinforced masonry piers. Eng Struct 2008;30:2242–52. https://doi.org/10.1016/j.engstruct.2007.12.001.

[22] Chen X, Wang X, Wang H, et al. Simulating the failure of masonry walls subjected to support settlement with the combined finite-discrete element method. J Build Eng 2021;43:102558. https://doi.org/10.1016/j.jobe.2021.102558.

[23] Chung SJ, Pollard N. Predictable behavior during contact simulation: a comparison of selected physics engines. Comput Animat Virtual Worlds 2016;27:262–70. https://doi.org/10.1002/cav.1712.

[24] Ciocci MP, Sharma S, Lourenço PB. Engineering simulations of a super-complex cultural heritage building: Ica Cathedral in Peru. Meccanica 2018;53:1931–58. https://doi.org/10.1007/s11012-017-0720-3.

[25] Collins J, Chand S, Vanderkop A, Howard D. A review of physics simulators for robotic applications. IEEE Access 2021;9:51416–31. https://doi.org/10.1109/ACCESS.2021.3068769.

[26] Coumans E, Bai Y. PyBullet, a Python module for physics simulation for games. Robot Mach Learn 2023. ⟨http://pybullet.org⟩.

[27] Courant R, Friedrichs K, Lewy H. On the partial difference equations of mathematical physics. IBM J Res Dev 1967;11:215–34. https://doi.org/10.1147/rd.112.0215.

[28] Cundall PA. Formulation of a three-dimensional distinct element model-Part I. A scheme to detect and represent contacts in a system composed of many polyhedral blocks. Int J Rock Mech Min Sci 1988;25:107–16. https://doi.org/10.1016/0148-9062(88)92293-0.

[29] Cundall P.A. (1971) The measurement and analysis of accelerations in rock slopes. PhD Thesis. University of London (Imperial College of Science and Technology).

[30] Cundall PA, Hart RD. Numerical Modeling of Discontinua. Comprehensive Rock Engineering. Elsevier; 1993. p. 231–43. Vol. 2.

[31] Cundall PA, Strack ODL. A discrete numerical model for granular assemblies. Geotechnique 1979;29:47–65. https://doi.org/10.1680/geot.1979.29.1.47.

[32] D'Altri AM, Sarhosis V, Milani G, et al. Modeling strategies for the computational analysis of unreinforced masonry structures: review and classification. Arch Comput Methods Eng 2020;27:1153–85. https://doi.org/10.1007/s11831-019-09351-x.

[33] D'Ayala D, Speranza E. Definition of collapse mechanisms and seismic vulnerability of historic masonry buildings. Earthq Spectra 2003;19:479–509. https://doi.org/10.1193/1.1599896.

[34] Damiani N, DeJong MJ, Albanesi L, et al. Distinct element modeling of the in-plane response of a steel-framed retrofit solution for URM structures. Earthq Eng Struct Dyn 2023;52:3030–52. https://doi.org/10.1002/eqe.3910.

[35] Damiani N, DeJong MJ, Albanesi L, et al. Parametric study on the in-plane performance of a steel frame retrofit solution for URM buildings using DEM. Eng Struct 2024;302:117293. https://doi.org/10.1016/j.engstruct.2023.117293.

[36] Davis L, Cogliano M, Casotto C, et al. Pragmatic seismic collapse meso-scale analysis of old Dutch masonry churches. Earthq Eng Struct Dyn 2024;53:622–45. https://doi.org/10.1002/eqe.4037.

[37] Dimitri R, Tornabene F. A parametric investigation of the seismic capacity for masonry arches and portals of different shapes. Eng Fail Anal 2015;52:1–34. https://doi.org/10.1016/j.engfailanal.2015.02.021.

[38] Domaneschi M, Cimellaro GP, Scutiero G. A simplified method to assess generation of seismic debris for masonry structures. Eng Struct 2019;186:306–20. https://doi.org/10.1016/j.engstruct.2019.01.092.

[39] Dubois F., Jean M. (2003) LMGC90 une plateforme de développement dédiée à la modélisation des problèmes d'interaction. In: Actes du 16eme Congrès français de mécanique. pp 111–118.

[40] Erleben K. Velocity-based shock propagation for multibody dynamics animation. –es ACM Trans Graph 2007;26:12. https://doi.org/10.1145/1243980.1243986.

[41] Erleben K, Macklin M, Andrews S, Kry PG. The Matchstick Model for anisotropic friction cones. Comput Graph Forum 2020;39:450–61. https://doi.org/10.1111/cgf.13885.

[42] Fazioli F., Ficuciello F., Fontanelli G.A., et al (2016) Implementation of a soft-rigid collision detection algorithm in an open-source engine for surgical realistic simulation. In: 2016 IEEE International Conference on Robotics and Biomimetics, ROBIO 2016. pp 2204–2208.

[43] Ferrante A, Loverdos D, Clementi F, et al. Discontinuous approaches for nonlinear dynamic analyses of an ancient masonry tower. Eng Struct 2021;230:111626. https://doi.org/10.1016/j.engstruct.2020.111626.

[44] Ferrante A, Schiavoni M, Bianconi F, et al. Influence of Stereotomy on Discrete Approaches Applied to an Ancient Church in Muccia, Italy. J Eng Mech 2021;147:4021103. https://doi.org/10.1061/(asce)em.1943-7889.0002000.

[45] Fita JL, Besuievsky G, Patow G. Earthquake simulation on ancient masonry buildings. J Comput Cult Herit 2020;13:1–18. https://doi.org/10.1145/3372421.

[46] Foti D, Vacca V, Facchini I. DEM modeling and experimental analysis of the static behavior of a dry-joints masonry cross vaults. Constr Build Mater 2018;170:111–20. https://doi.org/10.1016/j.conbuildmat.2018.02.202.

[47] Furukawa A, Spence R, Ohta Y, So E. Analytical study on vulnerability functions for casualty estimation in the collapse of adobe buildings induced by earthquake. Bull Earthq Eng 2010;8:451–79. https://doi.org/10.1007/s10518-009-9156-z.

[48] Gagliardo R, Godio M, Portioli FPA, Landolfo R. Seismic analysis of failure mechanisms in adjacent interacting stone masonry buildings via rigid block modeling. Bull Earthq Eng 2023;22:6195–224. https://doi.org/10.1007/s10518-023-01659-1.

[49] Gagliardo R, Portioli FPA, Cascini L, et al. A rigid block model with no-tension elastic contacts for displacement-based assessment of historic masonry structures subjected to settlements. Eng Struct 2021;229:111609. https://doi.org/10.1016/j.engstruct.2020.111609.

[50] Galvez F, Dizhur D, Ingham JM. Correction to: adjacent interacting masonry structures: shake table test blind prediction discrete element method simulation (Bulletin of Earthquake Engineering, (2023), 10.1007/s10518-023-01640-y). Bull Earthq Eng 2023:1–27. https://doi.org/10.1007/s10518-023-01655-5.

[51] Ghiassi B, Vermelfoort AT, Lourenço PB. Masonry mechanical properties. In: Ghiassi B, Milani GBT-NM of M and HS, editors. Woodhead Publishing Series in Civil and Structural Engineering. Woodhead Publishing; 2019. p. 239–61.

[52] Glondu L.L., Schvartzman S.C., Marchal M., et al (2012) Efficient Collision Detection for Brittle Fracture Efficient Collision Detection for Brittle Fracture 1-IRISA/INRIA Rennes 2-ENS Cachan, Antennes de Bretagne 3-INSA Rennes 4-

URJC Madrid. In: ACM SIGGRAPH/Eurographics Symposium on Computer Animation. pp 1–10.

[53] Goldstein H, Poole C, Safko J. Classical Mechanics. Pearson Education; 2011.

[54] Gonen S, Pulatsu B, Erdogmus E, et al. Effects of spatial variability and correlation in stochastic discontinuum analysis of unreinforced masonry walls. Constr Build Mater 2022;337:127511. https://doi.org/10.1016/j.conbuildmat.2022.127511.

[55] Govender N, Wilke DN, Kok S. Collision detection of convex polyhedra on the NVIDIA GPU architecture for the discrete element method. Appl Math Comput 2015;267:810–29. https://doi.org/10.1016/j.amc.2014.10.013.

[56] Grant DN, Dennis J, Sturt R, et al. Explicit modelling of collapse for Dutch unreinforced masonry building typology fragility functions. Bull Earthq Eng 2021;19:6497–519. https://doi.org/10.1007/s10518-020-00923-y.

[57] Graziotti F., Rossi A., Mandirola M., et al (2016) Experimental characterization of calcium-silicate brick masonry for seismic assessment. In: Brick and block masonry: trends, innovations and challenges—proceedings of the 16th international brick and block masonry conference, IBMAC. pp 1619–1628.

[58] Grunwald C, Khalil AA, Schaufelberger B, et al. Reliability of collapse simulation – comparing finite and applied element method at different levels. Eng Struct 2018;176:265–78. https://doi.org/10.1016/j.engstruct.2018.08.068.

[59] Haessig DA, Friedland B. On the modeling and simulation of friction. J Dyn Syst Meas Control Trans ASME 1991;113:354–62. https://doi.org/10.1115/1.2896418.

[60] Hamano T., Onosato M., Tanaka F. (2016) Performance comparison of physics engines to accelerate house-collapsing simulations. In: SSRR 2016 - International Symposium on Safety, Security and Rescue Robotics. pp 358–363.

[61] Hart R, Cundall PA, Lemos J. Formulation of a three-dimensional distinct element model-Part II. Mechanical calculations for motion and interaction of a system composed of many polyhedral blocks. Int J Rock Mech Min Sci 1988;25:117–25. https://doi.org/10.1016/0148-9062(88)92294-2.

[62] He H, Zheng J, Chen Y, Ning Y. Physics engine based simulation of shear behavior of granular soils using hard and soft contact models. J Comput Sci 2021;56: 101504. https://doi.org/10.1016/j.jocs.2021.101504.

[63] He H, Zheng J, Schaefer VR. Simulating shearing behavior of realistic granular soils using physics engine. Granul Matter 2021;23:56. https://doi.org/10.1007/s10035-021-01122-5.

[64] He H, Zheng J, Schaefer VR, et al. Simulation of realistic granular soils in triaxial test using physics engine. Comput Part Mech 2024;11:529–44. https://doi.org/10.1007/s40571-023-00637-3.

[65] Heyman J. The stone skeleton. Int J Solids Struct 1966;2:249–79. https://doi.org/10.1016/0020-7683(66)90018-7.

[66] Housner GW. The behavior of inverted pendulum structures during earthquakes. Bull Seismol Soc Am 1963;53:403–17. https://doi.org/10.1785/bssa0530020403.

[67] Imagire T, Johan H, Nishita T. A fast method for simulating destruction and the generated dust and debris. Vis Comput 2009;25:719–27. https://doi.org/10.1007/s00371-009-0319-3.

[68] Itasca Consulting Group Inc (2022) 3DEC — Three-Dimensional Distinct Element Code, Ver. 7.0.

[69] Itasca Consulting Group Inc (2014) UDEC 6.0 User's Guide.

[70] Jean M. The non-smooth contact dynamics method. Comput Methods Appl Mech Eng 1999;177:235–57. https://doi.org/10.1016/S0045-7825(98)00383-1.

[71] Klaus G, Glette K, Høvin M. Evolving locomotion for a 12-DOF quadruped robot in simulated environments. BioSystems 2013;112:102–6. https://doi.org/10.1016/j.biosystems.2013.03.008.

[72] Lacoursiere C. A Regularized Time Stepper for Multibody Systems. Department of Computing Science, Umeå University; 2006.

[73] Lagomarsino S, Cattari S. PERPETUATE guidelines for seismic performance-based assessment of cultural heritage masonry structures. Bull Earthq Eng 2015;13: 13–47. https://doi.org/10.1007/s10518-014-9674-1.

[74] Lancioni G, Lenci S, Piattoni Q, Quagliarini E. Dynamics and failure mechanisms of ancient masonry churches subjected to seismic actions by using the NSCD method: the case of the medieval church of S. Maria in Portuno. Eng Struct 2013; 56:1527–46. https://doi.org/10.1016/j.engstruct.2013.07.027.

[75] Lemos JV. Discrete element modeling of masonry structures. Int J Arch Herit 2007;1:190–213. https://doi.org/10.1080/15583050601176868.

[76] Li C, Liang W, Quigley C, et al. Earthquake safety training through virtual drills. IEEE Trans Vis Comput Graph 2017;23:1275–84. https://doi.org/10.1109/TVCG.2017.2656958.

[77] Li R, Yu J, Zhou X, Zhao M. Analysis of progressive collapse process of RC structures based on hybrid framework of FEM-physics engine. Eng Fail Anal 2023; 147:107138. https://doi.org/10.1016/j.engfailanal.2023.107138.

[78] Liang J, Fillmore S, Ma O. An extended bristle friction force model with experimental validation. Mech Mach Theory 2012;56:123–37. https://doi.org/10.1016/j.mechmachtheory.2012.06.002.

[79] Liu J, El-Assaly M, Garcia Mendez W, et al. A low-cost timber cladding system for the sustainable retrofit of masonry buildings: mechanical characterization under diagonal compression. Eng Struct 2025;322:119099. https://doi.org/10.1016/j.engstruct.2024.119099.

[80] Lloyd J.E. (2005) Fast implementation of Lemke's algorithm for rigid body contact simulation. In: Proceedings - IEEE International Conference on Robotics and Automation. pp 4538–4543.

[81] Lourenço PB, Oliveira DV, Roca P, Orduña A. Dry joint stone masonry walls subjected to in-plane combined loading. J Struct Eng 2005;131:1665–73. https://doi.org/10.1061/(asce)0733-9445(2005)131:11(1665).

[82] Lourenço PB, Ramos LF. Characterization of cyclic behavior of dry Masonry joints. J Struct Eng 2004;130:779–86. https://doi.org/10.1061/(ASCE)0733-9445(2004)130:5(779).

[83] Lourenço PB, Silva LC. Computational applications in masonry structures: from the meso-scale to the super-large/super-complex. Int J Multiscale Comput Eng 2020;18:1–30. https://doi.org/10.1615/IntJMultCompEng.2020030889.

[84] Lovreglio R, Gonzalez V, Feng Z, et al. Prototyping virtual reality serious games for building earthquake preparedness: the Auckland City Hospital case study. Adv Eng Inform 2018;38:670–82. https://doi.org/10.1016/j.aei.2018.08.018.

[85] Lu X, Guan H, Sun H, et al. A preliminary analysis and discussion of the condominium building collapse in surfside, Florida, US, June 24, 2021. Front Struct Civ Eng 2021;15:1097–110. https://doi.org/10.1007/s11709-021-0766-0.

[86] Ma Q.T., Parshottam S., Montalla M. (2018) Modelling rocking behaviour using physics engine simulation. In: 11th National Conference on Earthquake Engineering 2018, NCEE 2018: Integrating Science, Engineering, and Policy. Earthquake Engineering Research Institute, pp 1212–1222.

[87] Maciel A, Halic T, Lu Z, et al. Using the PhysX engine for physics-based virtual surgery with force feedback. Int J Med Robot Comput Assist Surg 2009;5:341–53. https://doi.org/10.1002/rcs.266.

[88] Macklin M., Storey K., Lu M., et al (2019) Small steps in physics simulation. In: Proceedings - SCA 2019: ACM SIGGRAPH / Eurographics Symposium on Computer Animation. Association for Computing Machinery, Inc.

[89] Malomo D, DeJong MJ. A macro-distinct element model (M-DEM) for simulating the in-plane cyclic behavior of URM structures. Eng Struct 2021;227:111428. https://doi.org/10.1016/j.engstruct.2020.111428.

[90] Malomo D, Mehrotra A, DeJong MJ. Distinct element modeling of the dynamic response of a rocking podium tested on a shake table. Earthq Eng Struct Dyn 2021;50:1469–75. https://doi.org/10.1002/eqe.3404.

[91] Malomo D, Pinho R, Penna A. Simulating the shake table response of unreinforced masonry cavity wall structures tested to collapse or near-collapse conditions. Earthq Spectra 2020;36:554–78. https://doi.org/10.1177/8755293019891715.

[92] Malomo D, Pinho R, Penna A. Using the applied element method for modelling calcium silicate brick masonry subjected to in-plane cyclic loading. Earthq Eng Struct Dyn 2018;47:1610–30. https://doi.org/10.1002/eqe.3032.

[93] Malomo D, Pulatsu B. Discontinuum models for the structural and seismic assessment of unreinforced masonry structures: a critical appraisal. Structures 2024;62. https://doi.org/10.1016/j.istruc.2024.106108.

[94] Mazhar H, Heyn T, Pazouki A, et al. CHRONO: a parallel multi-physics library for rigid-body, flexible-body, and fluid dynamics. Mech Sci 2013;4:49–64. https://doi.org/10.5194/ms-4-49-2013.

[95] Mehrotra A, DeJong MJ. A methodology to account for interface flexibility and crushing effects in multi-block masonry collapse mechanisms. Meccanica 2020; 55:1237–61. https://doi.org/10.1007/s11012-020-01161-x.

[96] Milaszewicz JP. Improving Jacobi and Gauss-Seidel Iterations. Linear Algebra Appl 1987;93:161–70. https://doi.org/10.1016/S0024-3795(87)90321-1.

[97] Minga E, Macorini L, Izzuddin BA, Caliò I. 3D macroelement approach for nonlinear FE analysis of URM components subjected to in-plane and out-of-plane cyclic loading. Eng Struct 2020;220:110951. https://doi.org/10.1016/j.engstruct.2020.110951.

[98] Misseri G, DeJong MJ, Rovero L. Experimental and numerical investigation of the collapse of pointed masonry arches under quasi-static horizontal loading. Eng Struct 2018;173:180–90. https://doi.org/10.1016/j.engstruct.2018.06.009.

[99] Müller M, Chentanez N, Kim TY. Real time dynamic fracture with volumetric approximate convex decompositions. ACM Transactions on Graphics. New York, NY, USA: Association for Computing Machinery; 2013.

[100] Nie Q., Zhao Y., Xu L., Li B. (2020) A Survey of Continuous Collision Detection. In: Proceedings - 2020 2nd International Conference on Information Technology and Computer Application, ITCA 2020. IEEE, pp 252–257.

[101] Oktiovan YP, Messali F, Pulatsu B, et al. A contact-based constitutive model for the numerical analysis of masonry structures using the distinct element method. Comput Struct 2024;303:107499. https://doi.org/10.1016/j.compstruc.2024.107499.

[102] Oliveira D.V. (2003) Experimental and numerical analysis of blocky masonry. PhD Thesis. Universidade do Minho.

[103] Penna A, Senaldi IE, Galasco A, Magenes G. Numerical simulation of shaking table tests on full-scale stone masonry buildings. Int J Arch Herit 2016;10:146–63. https://doi.org/10.1080/15583058.2015.1113338.

[104] Portioli F, Casapulla C, Cascini L. An efficient solution procedure for crushing failure in 3D limit analysis of masonry block structures with non-associative frictional joints. Int J Solids Struct 2015;69–70:252–66. https://doi.org/10.1016/j.ijsolstr.2015.05.025.

[105] Portioli F, Cascini L. Large displacement analysis of dry-jointed masonry structures subjected to settlements using rigid block modelling. Eng Struct 2017; 148:485–96. https://doi.org/10.1016/j.engstruct.2017.06.073.

[106] Pulatsu B., Erdogmus E., Bretas E.M., Lourenço P.B. (2019a) In-plane static response of dry-joint masonry arch-pier structures. In: AEI 2019: Integrated Building Solutions - The National Agenda - Proceedings of the Architectural Engineering National Conference 2019. pp 240–248.

[107] Pulatsu B, Erdogmus E, Lourenço PB, Quey R. Simulation of uniaxial tensile behavior of quasi-brittle materials using softening contact models in DEM. Int J Fract 2019;217:105–25. https://doi.org/10.1007/s10704-019-00373-x.

[108] Pulatsu B, Gencer F, Erdogmus E. Study of the effect of construction techniques on the seismic capacity of ancient dry-joint masonry towers through DEM. Eur J Environ Civ Eng 2022;26:3913–30. https://doi.org/10.1080/19648189.2020.1824823.

[109] Pulatsu B, Gonen S, Erdogmus E, et al. In-plane structural performance of dry-joint stone masonry Walls: a spatial and non-spatial stochastic discontinuum analysis. Eng Struct 2021;242:112620. https://doi.org/10.1016/j.engstruct.2021.112620.

[110] Pulatsu B, Gonen S, Lourenço PB, et al. Computational investigations on the combined shear–torsion–bending behavior of dry-joint masonry using DEM. Comput Part Mech 2023;10:249–60. https://doi.org/10.1007/s40571-022-00493-7.

[111] Raka E, Spacone E, Sepe V, Camata G. Advanced frame element for seismic analysis of masonry structures: model formulation and validation. Earthq Eng Struct Dyn 2015;44:2489–506. https://doi.org/10.1002/eqe.2594.

[112] Restrepo Vélez LF, Magenes G, Griffith MC. Dry stone masonry walls in bending-Part I: Static tests. Int J Arch Herit 2014;8:1–28. https://doi.org/10.1080/15583058.2012.663059.

[113] Salvalaggio M, Bernardo V, Lourenço PB. Exploring seismic fragility and strengthening of masonry built heritage in Lisbon (Portugal) via the Applied Element Method. Eng Struct 2024;320:118890. https://doi.org/10.1016/j.engstruct.2024.118890.

[114] Sangirardi M, Liberatore D, Addessi D. Equivalent frame modelling of masonry walls based on plasticity and damage. Int J Arch Herit 2019;13:1098–109. https://doi.org/10.1080/15583058.2019.1645240.

[115] Santarelli S, Bernardini G, Quagliarini E. Earthquake building debris estimation in historic city centres: from real world data to experimental-based criteria. Int J Disaster Risk Reduct 2018;31:281–91. https://doi.org/10.1016/j.ijdrr.2018.05.017.

[116] Scattarreggia N, Salomone R, Moratti M, et al. Collapse analysis of the multi-span reinforced concrete arch bridge of Caprigliola, Italy. Eng Struct 2022;251:113375. https://doi.org/10.1016/j.engstruct.2021.113375.

[117] Schiavoni M, Giordano E, Roscini F, Clementi F. Numerical modeling of a majestic masonry structure: A comparison of advanced techniques. Eng Fail Anal 2023;149:107293. https://doi.org/10.1016/j.engfailanal.2023.107293.

[118] Schiavoni M, Roscini F, Clementi F. Comparative analysis between continuous and discontinuous methods for the assessment of a cultural heritage structure. Meccanica 2024. https://doi.org/10.1007/s11012-024-01885-0.

[119] Sellán S, Luong J, Mattos Da Silva L, et al. Breaking good: fracture modes for realtime destruction. ACM Trans Graph 2023;42. https://doi.org/10.1145/3549540.

[120] Sharma S, Marasca A, Ponte M, Bento R. Modelling the in-plane cyclic behaviour of typical Portuguese rubble stone masonry using the applied element method. Structure 2022:1224–42.

[121] Shen Y., Sun X. (2011) Research and improvement of collision detection based on oriented bounding box in physics engine. In: 2011 IEEE 3rd International Conference on Communication Software and Networks, ICCSN 2011. pp 73–76.

[122] Sheng Chen G, Liu X. Friction. In: Sheng Chen G, Liu X, editors. Friction Dynamics Principles and Applications. Woodhead Publishing; 2016. p. 91–159.

[123] Silcowitz M, Niebe S, Erleben K. Interactive rigid body dynamics using a projected gauss-seidel subspace minimization method. In: Richard P, Braz J, editors. Communications in Computer and Information Science. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. p. 218. vol. 229.

[124] Silva LC, Lourenço PB, Milani G. Derivation of the out-of-plane behaviour of masonry through homogenization strategies: micro-scale level. Comput Struct 2018;209:30–43. https://doi.org/10.1016/j.compstruc.2018.08.013.

[125] Smoljanović H, Nikolić Ž, Živaljić N. A combined finite-discrete numerical model for analysis of masonry structures. Eng Fract Mech 2015;136:1–14. https://doi.org/10.1016/j.engfracmech.2015.02.006.

[126] So E, Spence R. Estimating shaking-induced casualties and building damage for global earthquake events: a proposed modelling approach. Bull Earthq Eng 2013;11:347–63. https://doi.org/10.1007/s10518-012-9373-8.

[127] Stewart DE, Trinkle JC. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. Int J Numer Methods Eng 1996;39:2673–91. https://doi.org/10.1002/(SICI)1097-0207(19960815)39:15<2673::AID-NME972>3.0.CO;2-I.

[128] Tagel-Din H. (1998) A new efficient method for nonlinear, large deformation and collapse analysis of structures. Ph. D. thesis, Civ. Eng. Dept., Univ. Tokyo.

[129] Thomas R, Zhang W. Real-time fracturing in video games. Multimed Tools Appl 2023;82:4709–34. https://doi.org/10.1007/s11042-022-13049-x.

[130] Tomaževič M. Dynamic modelling of masonry buildings: storey mechanism model as a simple alternative. Earthq Eng Struct Dyn 1987;15:731–49. https://doi.org/10.1002/eqe.4290150606.

[131] Valentino J, Gilbert M, Gueguin M, Smith CC. Limit analysis of masonry walls using discontinuity layout optimization and homogenization. Int J Numer Methods Eng 2023;124:358–81. https://doi.org/10.1002/nme.7124.

[132] Vasconcelos G. (2005) Experimental investigations on the mechanics of stone masonry: Characterization of granites and behavior of ancient masonry shear walls. PhD Thesis. Universidade do Minho.

[133] Vasconcelos G, Lourenço PB. In-plane experimental behavior of stone Masonry walls under cyclic loading. J Struct Eng 2009;135:1269–77. https://doi.org/10.1061/(asce)st.1943-541x.0000053.

[134] Vlachakis G, Colombo C, Giouvanidis AI, et al. Experimental characterisation of dry-joint masonry structures: interface stiffness and interface damping. Constr Build Mater 2023;392:130880. https://doi.org/10.1016/j.conbuildmat.2023.130880.

[135] Whiting E, Ochsendorf J, Durand F. Procedural modeling of structurally-sound masonry buildings. ACM Trans Graph 2009;28:1–9. https://doi.org/10.1145/1618452.1618458.

[136] Wilson R, Szabó S, Funari MF, et al. A comparative computational investigation on the in-plane behavior and capacity of dry-joint URM walls. Int J Arch Herit 2024;18:849–70. https://doi.org/10.1080/15583058.2023.2209776.

[137] Zhang Z, Davis L, Malomo D. Distinct element macro-crack networks for expedited discontinuum seismic analysis of large-scale URM structures. J Build Eng 2024;97:110962. https://doi.org/10.1016/j.jobe.2024.110962.