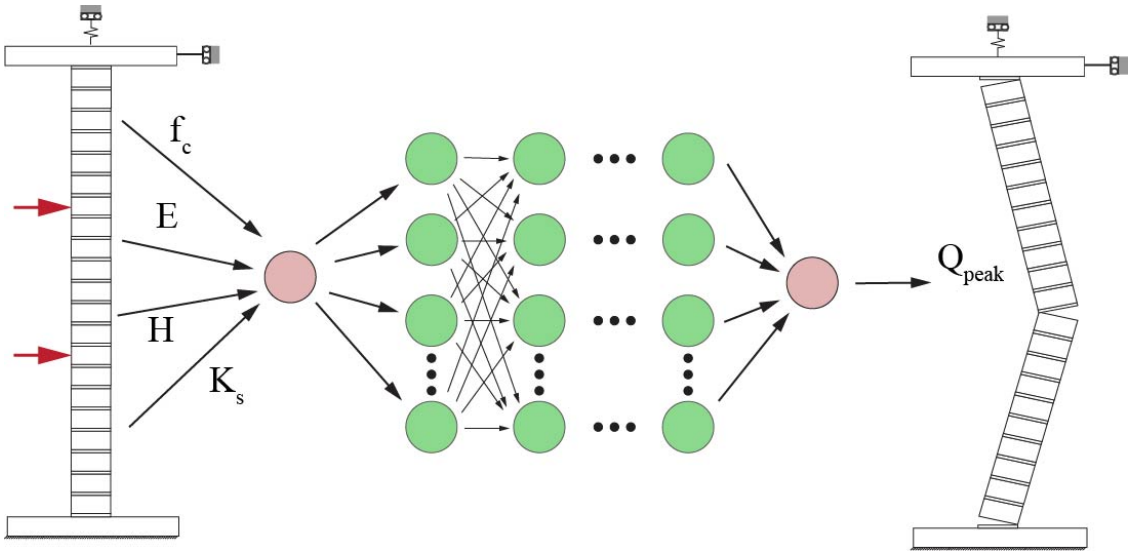


Data-Driven Prediction of Out-of-Plane Force Capacity in Unreinforced Masonry Walls Under Blast Loading

A Modeling Approach Driven by Machine Learning

Master's thesis in Structural Engineering

Jesper Danielsson
Samuel Mårtensson

DEPARTMENT OF STRUCTURAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Data-Driven Prediction of Out-of-Plane Force Capacity in Unreinforced Masonry Walls under Quasi-Static Loading

A Modeling Approach Driven by Machine Learning

JESPER DANIELSSON, SAMUEL MÅRTENSSON

CHALMERS

Department of Architecture and Civil Engineering
Division Structural Engineering
Name of research group (if applicable)
Chalmers University of Technology
Gothenburg, Sweden 2026

Data-Driven Prediction of Out-of-Plane Force Capacity in Unreinforced Masonry Walls under Quasi-Static Loading
A Modeling Approach driven by Machine Learning
JESPER DANIELSSON, SAMUEL MÅRTENSSON

Supervisor: Michele Godio, RISE, August Claesson, Tyréns
Examiner: Joosef Leppänen, Head of the Division Structural Engineering

Master's Thesis 2026
Department of Architecture and Civil Engineering
Division Structural Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Schematic representation of the machine learning framework, where material and geometrical parameters are used as inputs to predict the force capacity and failure response of the masonry wall..

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Data-Driven Prediction of Out-of-Plane Force Capacity in Unreinforced Masonry Walls under Quasi-Static Loading
A Modelling Approach Driven by Machine Learning
JESPER DANIELSSON, SAMUEL MÅRTENSSON
Division of Structural Engineering
Chalmers University of Technology

ABSTRACT

Unreinforced masonry walls compromises a large part of the heritage buildings and older civilian buildings in city centers around the world. The current analytical evaluation methods for the out-of-Plane force capacity remain unreliable, which complicates assessments of buildings in regards to updated strength requirements. Masonry walls consist of alternating layers of brick and mortar which makes the discretization into a numerical model more complicated. The interfaces and contacts between materials increase rapidly with added layers resulting in very computationally heavy models. There are therefore a need for a simplified approach to approximate the force capacity, which may be achieved through the application of machine learning.

Numerical modeling within this thesis utilizes a 2D micro-model, where both brick and mortar are included to capture crushing, in any of the materials. The explicit dynamic solver in Abaqus is used to design a quasi-static four point bending test. Geometric and material properties, along with global force and stiffness parameters, are identified and parameterized to generate a comprehensive set of loading cases. A dataset containing the parameters for each loading case of the unreinforced masonry wall is created for machine learning purposes.

In this study, several machine learning models are developed with the common objective of predicting the peak force capacity of unreinforced masonry walls. The developed models differ in their prediction targets, including direct prediction of the peak force capacity, prediction of the force–displacement response up until the peak, and the discovery of a new analytical expression for the peak force capacity.

The report is written in English.

Keywords: unreinforced masonry, out-of-plane, force capacity, numerical models, micro-model, machine learning.

Acknowledgements

The authors would like to express their sincere gratitude to all individuals and organizations who contributed to the completion of this thesis.

First and foremost, we would like to extend our sincere appreciation to our supervisors, Michele Godio at RISE Research Institutes of Sweden and August Claesson at Tyréns, for their continuous support, guidance, and valuable expertise throughout this work. Their knowledge and feedback have been invaluable to the progress and quality of this thesis.

We would also like to thank Viktor Peterson at the Swedish Defence Research Agency (FOI) and Mathias Flansbjer at RISE, whose discussions and input have been valuable and greatly appreciated.

Furthermore, we would like to thank our examiner, Joosef Leppänen, for valuable guidance and constructive feedback during the course of the thesis.

This thesis was conducted within the framework of the larger RISE project umbrella Blast Capacity of Masonry Walls, which falls within the research area CFORT D1 – Explosion and Weaponry Effects. The authors gratefully acknowledge the support provided through this initiative, which made this work possible.

Additional funding and support were provided through the project ThickAsBrick | Data-driven modelling of the out-of-plane strength of blast-loaded masonry walls for assessment and fortification, funded by the Fortification Foundation.

Finally, we would like to thank everyone who, directly or indirectly, contributed to this work and supported us throughout the thesis process.

Jesper Danielsson, Samuel Mårtensson, Gothenburg, June 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

FE	Finite Element
IVP	Intial Value Problem
GP	Genetic Programming
ML	Machine Learning
MLP	Multi Layered Perceptron
NODE	Neural Ordinary Differential Equation
ODE	Ordinary Differential Equation
OOP	Out-Of-Plane
URM	UnReinforced Masonry
RK4	Fourth-order Runge Kutta
RISE	Reaserch Institute Of Sweden
ROM	Reduced Order Modeling
SR	Symbolic Regression

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Parameters

t	Thickness
h_m	Height mortar
H	Height
λ	Slenderness
E	Elastic modulus
f_c	Compressive strength
f_t	Tensile strength
K_s	Spring stiffness
F_O	Pre-compressive force

SI units

MPa	Pressure
kN	Force
mm	Geometry
kN/mm	Stiffness

Contents

Nomenclature **xi**

List of Figures **xvii**

List of Tables **xxi**

1 Introduction **1**
1.1 Background . . . 1
1.2 Aim . . . 2
1.3 Objectives . . . 2
1.4 Limitations . . . 3
1.5 Societal, ethical and ecological aspects . . . 3
1.6 Methods . . . 3
1.6.1 Literature Review . . . 4
1.6.2 Existing Analytical Models on the peak force capacity . . . 4
1.6.3 Finite Element Analysis . . . 5
1.6.4 Machine Learning Models for Predicting OOP Force Capacity 5
1.7 Research Novelty . . . 6
1.8 Project pipeline . . . 7

2 Masonry Walls **9**
2.1 Failure mechanisms in masonry walls . . . 9
2.2 Analytical models on the out-of-plane strength . . . 11
2.2.1 Eurocode 6 – Lateral loading . . . 11
2.2.2 Martens and Vermeltfoort lateral loading model . . . 14
2.2.3 Edri and Yankelevsky reduced order model on the out-of-plane dynamic response . . . 18

3 Numerical model **21**
3.1 Modelling of Quasi-Static Loading Conditions . . . 21
3.2 Development and Setup of the Finite Element Framework . . . 22
3.2.1 Modeling of the Single-Wythe Wall Specimen . . . 23
3.2.2 Concrete Damaged Plasticity Model for Masonry: Formulation and Implementation . . . 25
3.3 Validation of the ABAQUS Model Against Experimental Data . . . 28
3.4 Parametrization of the Material Units . . . 32
3.4.1 Background to Brick as a Building Material . . . 32
3.4.2 Parameterization of the Brick Units . . . 33
3.4.3 Background to Mortar as a Building Material . . . 35
3.4.4 Parametrization of the Mortar Units . . . 36

3.4.5 Global Parametrization of the Numerical Model 37
3.4.6 Execution and Results From the Parametric Study 38

4 Machine Learning 41
4.1 Introduction to Machine Learning 41
4.2 The Learning Paradigm 42
4.3 Artificial Neural Networks (ANN's) 44
4.4 Network Architecture, Scaling and Normalization 46
4.5 The Loss Function 49
4.6 Backpropagation 50
4.7 Neural Ordinary Differential Equations 51

5 Machine Learning Models 55
5.1 Design of Multi-layered Perceptron Predicting the Out-of-Plane Peak Force Capacity 55
5.1.1 Results of the MLP 58
5.1.2 Discussion of the MLP prediction results 61
5.2 Neural ODE model predicting the out-of-plane force capacity 62
5.2.1 Design and Architecture of the Neural ODE Model 62
5.2.2 Results of the Neural ODE on the Out-of-Plane Force Capacity 64
5.2.3 Discussion of the NODE prediction Results 68
5.3 Symbolic Regression 69
5.3.1 Design and utilization of the PySR model 69
5.3.2 Results of the Symbolic Regression 72
5.3.3 Discussion of the PySR Prediction Results 75

6 Conclusion & Future Work 77

Bibliography 81

A Appendix 1 - Mass scaling I
A.1 Mass scaling I
A.2 Out of plane step mass scaling III

B Appendix 2 - Time step V
B.1 Pre - compressive time step VI
B.2 Out of plane time step VIII

C Appendix 3 - Mesh size XI

D Appendix 4 - Pre compressive force XV

E Appendix 5 - Double wythe numerical model XVII
E.1 Double wythe wall XVII
E.2 Validation of double-wythe wall XVIII

F Appendix 6 — Distribution of Parametric Values XXI

G Appendix 7 — Direct SR Prediction of the Force capacity XXV

H Appendix 8 — Distributed load compared with four point loading in ABAQUS XXVII

List of Figures

1.1 Overview of the theoretical, numerical, and machine-learning workflow. 7

2.1 Different failure modes for a URM wall, as prescribed by experimental tests at RISE (Godio et al., 2023a). 10
2.2 Structural model for the derivation of the force capacity for out-of-plane loading in EC6. 11
2.3 Parity plot between the experientially obtained peak force and the peak force calculated in accordance with EC6. 14
2.4 Two different thrust line geometries showing a triangular (left) and parabolic (right) shape depending on the loading condition. 15
2.5 Definition of the parameters ψ and d_G, of the stress–strain area and $\delta_G = d_G/\varepsilon_u$. 15
2.6 Reduced-order interpretation of arching action. The thrust-line assumptions describe the arching mechanism governing the wall response, while the dynamic reduced-order model represents this behavior through the generalized mid-height displacement $U(t)$ and the nonlinear spring $K_e(U)$. 18
2.7 Reduced-order interpretation of arching action. The thrust-line assumptions describe the arching mechanism governing the wall response, while the quasi-static reduced-order model represents this behavior through the generalized mid-height displacement U and the nonlinear resistance $K_e(U)$. 19

3.1 Single wythe wall as modeled in ABAQUS, where reference points (RP) are indicated with yellow crosses and the force and velocity is indicated by red arrows. 24
3.2 Stress–strain relationships in compression and tension for the adopted material model, with markers indicating the characteristic values used in the formulation. 28
3.3 Results from the numerical model (blue) compared to the experimental results (orange) for the W1 wall. 30
3.4 Results from the numerical model (blue) compared to the experimental results (orange) for the W2 wall. 30
3.5 Results from the numerical model (blue) compared to the experimental results (orange) for the W3 wall. 31
3.6 Results from the numerical model (blue) with a very high stiffness compared to a different numerical model with rigid top boundary (orange). 32
3.7 Distributions of the selected parametric values. 39
3.8 Spearman correlation matrix. 40

4.1 Conceptual image of the processing of a perceptron unit with several inputs and one output (Nielsen, 2015). 42
4.2 The ReLU function for the input $z \in$ [-6, 6] (Nielsen, 2015). 45
4.3 The Tanh function for the input $z \in$ [-6, 6] (Nielsen, 2015). 46
4.4 Schematic of a ANN architecture (Nielsen, 2015). 47
4.5 Schematic of a ANN architecture with transformation layers (Nielsen, 2015). 48
4.6 The optimization trajectory in the weight space. 51

5.1 The Spearman correlation matrix for the subset of parameters chosen from the parameter space. 56
5.2 SHAP analysis for parameter evaluation toward target value. 58
5.3 Evolution of training loss for the MLP model. 59
5.4 Parity plot of validation and training for the MLP model. 59
5.5 Evaluation of the MLP against the experimental tests. 60
5.6 Reduced order modeling (ROM) of force-displacement system at the mid-height of the wall strip. 62
5.7 The Loss vs. Epochs during the training of the NODE. 64
5.8 Predicted trajectory of the force-displacement trajectories in the validation subset for different values of the boundary spring K_s. 65
5.9 Parity plot of the peak force capacity Q_{peak} between Abaqus data and the predicted value for all training and validation cases. 66
5.10 Force-Displacemend field prediction of the physical RISE experiment data for Specimens W1, W2 and W3. 66
5.11 Force-Displacemend field prediction of the physical RISE experiment data for Specimens W4, W5 and W8. Double-wythe bending failure. . 67
5.12 Force-Displacemend field prediction of the physical RISE experiment data for Specimens W13 and W14. 67
5.13 Force-Displacemend field prediction of the physical RISE experiment data for Specimens W6, W7 and W12. 68
5.14 Q_{peak} parity plot analytically searched with the resulting *Thrust*. . . . 71
5.15 Thrust equation found using PySR compared with ABAQUS results. 72
5.16 Force capacity with equation of newly developed thrust force expression. 73
5.17 Comparison of the expression for force capacity in EC6 (blue) and PySR (red). 74
5.18 Comparison of EC6 (+) and PySR (dots) expressions against the experimental data. 74

A.1 Mass scaling increase effect in the pre compressive step with the validation of wall W3. II
A.2 Mass scaling increase effect on the out of plane step with the validation of wall W3. III

B.1 Different step times effect on the reaction forces in the pre compressive step. VI
B.2 Different step times effect on the reaction forces in the out-of-plane step. VIII

C.1 Different size mesh of the numerical model and the responses of the model. XII

D.1 Simple calculation for the decision of the maximum value of the parametric range of the pre-compressive force XV

E.1 Double wythe wall as modeled in ABAQUS, with reference points highlighted by yellow crosses and applied forces shown as red arrows. XVIII
E.2 Results from the numerical model (blue) compared with the experimental data of walls W4 and W5. XIX
E.3 Results from the numerical model (blue) compared with the experimental tests of wall W7. XIX
E.4 Final picture of W7 from ABAQUS with shear failure close to the supports. XX

F.1 Distribution of mechanical parameters affecting brick properties . . . XXI
F.2 Distribution of mechanical parameters affecting mortar properties . . XXII
F.3 Distribution of global parameters effecting the walls XXIII

G.1 Parity plot between the ABAQUS results and the developed equation of PySR. XXV
G.2 Comparison of the experimental data and the new expression XXVI

H.1 Distributed load over the single wythe wall XXVII
H.2 Comparison between four point bending (orange) with distributed load (blue) and the experimental data (green) for wall W1. XXVIII
H.3 Comparison between four point bending (orange) with distributed load (blue) and the experimental data (green) for wall W2. XXVIII
H.4 Comparison between four point bending (orange) with distributed load (blue) and the experimental data (green) for wall W3. XXIX

List of Tables

2.1 Overview of the walls presented in the experimental tests (Godio et al., 2023a). 10

3.1 Parameters introduced into the CDP model to govern the modified Drucker–Prager plasticity yield criterion 25
3.2 Material parameters used for validation, with assumed values indicated in italics. 29
3.3 Applied loads and spring stiffness used in the validation study, and the calibrated values used in ABAQUS. 29
3.4 Ratio between Young's modulus and compressive strength of bricks. . 34
3.5 Parametric range of brick. 35
3.6 Mechanical properties of different kinds of mortar. 36
3.7 Parametric range of mortar. 37

5.1 Summary of the MLP architecture used for prediction of Q_{peak}. . . . 57
5.2 Summary of the MLP training setup. 57
5.3 Summary of prediction performance for the MLP model. 60
5.4 Summary of prediction performance for the MLP model on RISE data. 61
5.5 Summary of the NODE architecture and framework design. 64
5.6 Summary of user defined parameters in PySR. 70

A.1 Computational time of different mass scaling in the pre compressive step II
A.2 Computational time of different mass scaling in the out of plane step IV

B.1 Computational time of different mass scaling in the pre compressive step VII
B.2 Computational time of different step times of the out-of-plane step, with the max ratio of the kinetic and internal energy. IX

C.1 Computational time of different mesh sizes in the numerical model for wall W3 XII

1 Introduction

This chapter outlines the background of the study, providing the context for the research. It then states the aim and objectives that guide the investigation, followed by a discussion of the study's limitations, which define the scope and constraints of the research. Further, the methodologies used to reach the end result are presented. Lastly, the project pipeline gives an overview of the project structure.

1.1 Background

Unreinforced brick masonry (URM) buildings constitute a large part of Sweden's architectural heritage and civilian building stock, both as exterior facades and as massive load-bearing walls (Gustafsson, 2023). Many of these buildings were constructed in the early twentieth century and play a significant role in architectural history and cultural heritage. As a result, they are often protected and intended to be preserved. In addition to heritage buildings, important infrastructure, social infrastructure, and defense related facilities such as railway stations, schools, hospitals, and certain military buildings, are also constructed using brick masonry.

Societal changes have led to an increase in threats from blasts, most notably in the form of terrorism and crime in civilian locations, but also from the potential threat of military attack (Johansson & Laine, 2012). A blast generates extraordinary out-of-plane (OOP) loads on structures. Older buildings were never designed to withstand such extreme loading conditions. Although masonry is a heavy and bulky material, it is weak in tension, particularly at the interfaces between brick and mortar (Gustafsson, 2023). To safeguard both occupants and structures, assessments must be carried out to determine the damage a blast could inflict on a building. Depending on the size and location of the blast, consequences can range from small cracks and exterior damage requiring minor to none repairs to total structural failure rendering the use of the building completely (Kallas & Napolitano, 2024).

Calculations for assessing blast-loaded masonry rely on force-capacity expressions based on a three-hinge bending mechanism (Abrams et al., 1996; Anderson, 1984; European Committee for Standardization, 2006; Martens et al., 2017), where the governing parameters are the compressive strength of the masonry and the depth of the plastic hinge. However, there is no consensus regarding the values of these parameters, and each method yields varying results. Recent four-point bending tests have also shown that the failure mechanisms of URM walls differ beyond bending, with shear or shear–bending failures as a possibility depending on the slenderness of the wall (Godio et al., 2023a), indicating that the current expressions based solely on bending are insufficient. Further, it is unclear how to accurately scale the properties of mortar and bricks into the representative overall properties. Together, these

uncertainties restrict the applicability and reliability of current analytical analysis, and ultimately hinder effective assessment and design of blast-resistant heritage masonry structures. While numerical models are widely used to analyses and assess the out-of-plane strength of URM walls, they require extensive expertise, careful modeling, and significant computational resources (D'Altri et al., 2020). Therefore, numerical models that accurately capture the OOP capacity of URM walls can, with the aid of machine learning, bridge the limitations of existing analytical expressions, thereby significantly reducing the time and effort required for simplified URM wall evaluations.

Current knowledge of the blast response of URM walls remains scattered (Godio & Flansbjer, 2025a). To consolidate and further develop this knowledge, Centrum för Fortifikatorisk Kompetens (CFORT) has established the research area "CFORT-D1 - Explosion and weaponry effects." Within this area, the RISE project umbrella "Blast capacity of masonry walls" conducts research aimed at improving the understanding of URM wall behaviour under blast loading. A further initiative, funded by the Fortification Foundation and titled "ThickAsBrick | Data-driven modeling of the out-of-plane strength of blast-loaded masonry walls for assessment and fortification," supports the development of modern data-driven models for assessing URM response, with particular emphasis on the application of machine learning.

1.2 Aim

The aim of this thesis is to develop improved methods for assessing the OOP strength of masonry walls. This includes creating a new force capacity model that overcomes limitations in current code-based approaches through the use of ML. The intended outcome is to enable more accurate assessments of masonry structures, thereby supporting safer and more reliable design practices.

1.3 Objectives

To comprise the objectives some specific research questions have been developed.

- Can the OOP peak force capacity be approximated from the static wall parameters using machine learning techniques?
- Can the force-displacement evolution for the OOP in URM walls be approximated using machine learning models?
- Can machine learning facilitate discovery of an improved analytical expression of the OOP peak force capacity in URM walls?

To address these questions, different force-capacity models need to be reviewed and understood to identify their limitations and to evaluate how they compare with numerical models and experimental data. Numerical models that accurately represent

the behavior of masonry walls are created and analyzed for different geometries, boundary conditions and material parameters. Experimental data exists and to some extent used to verify the numerical models. ML models are developed based on these numerical results capable of predicting the OOP peak force capacity of different URM walls. Then further developing one or several expressions to describe force capacity.

1.4 Limitations

The analyzes are based on available literature, simulations, and test data. The focus are on the quasi-static evaluation of brick masonry walls as a first step in the development of a model to evaluate impulse loads, although the impulse loads themselves where not applied. The numerical models are created for evaluation as brick masonry walls. Other types of masonry walls where not evaluated even if there are a similar failure process. The ML model developed is scalable and potentially applicable to other masonry typologies but not confirmed.

1.5 Societal, ethical and ecological aspects

Understanding how blasts affect brick masonry walls is key to the construction and retrofitting of existing buildings in order to protect occupants, thereby increasing societal resilience. From an ethical and ecological standpoint, the findings may contribute to more material-efficient designs. By improving the understanding and prediction of the OOP capacity of brick masonry walls, unnecessary conservatism in design may be reduced, leading to optimized use of construction materials without compromising safety and with a positive environmental impact. The findings may also indicate that current design codes are insufficient and call for an increase in material usage, particularly in blast-resistant design applications, which could have a negative environmental impact.

1.6 Methods

This chapter presents the workflow used to develop an assessment model for the out-of-plane capacity of URM walls. The approach consists of a scoping literature review, nonlinear finite element modelling in Abaqus to generate a parametric database, and machine learning trained on the simulation results with theory-guided regularization for improved generalization.

1.6.1 Literature Review

The aim of the thesis is to develop a model for assessing the OOP load capacity of URM walls by combining theory and numerical modeling with ML. To capture the fundamental information required, the research topic spans multiple disciplines and evidence types. These include analytical theory, experimental investigations, design standards, numerical modeling, and machine learning approaches. A scoping review methodology is adopted, as presented by Arksey and O'Malley (Arksey & O'Malley, 2005). The approach is selected for its suitability to capture key concepts across a broad research field and to identify gaps for further development.

The methodological framework of the scoping review approach includes five main stages to ensure a robust structure for producing results (Arksey & O'Malley, 2005). The first stage is to identify the research question and determine which parts of the formulation are most important in order to guide the search strategy. Examples of key terms include "out-of-plane load" and "unreinforced masonry walls". Secondly, with guidance from the first stage, relevant studies are identified through database searches in EBSCO, ScienceDirect, IEEE, and Google Scholar. In combination with database searching, forward and backward "snowballing" is applied to establish a hybrid search strategy (Wohlin et al., 2022). Technical reports and standards are also included to identify influential publications and follow-up work through citation tracking.

The sources are then condensed into a final selection by narrowing down the research question and filtering the material to the most relevant studies (Arksey & O'Malley, 2005). In the fourth stage, the selected sources are "charted" to capture the key takeaways from each source and group the findings thematically to support specific parts of the project. Extracted information may include wall geometry, boundary conditions, reported force capacity, observed failure modes, and numerical or simulation methodologies. The final stage is to summarize and report the results for each category in a way that can be used for further investigation in the project.

1.6.2 Existing Analytical Models on the peak force capacity

Analytical models for the out-of-plane response of URM walls are evaluated to provide mechanical baseline predictions for the ML models. The models studied in the theory chapter describe the response through arching action, thrust force development and three-hinge bending mechanisms (Martens et al., 2017). In this thesis, they are used both to compare peak-force predictions and to be used as reference for the discovered equations of the the symbolic regression (SR) results.

Further, a reduced-order formulation based on Edri and Yankelevsky is presented (Edri & Yankelevsky, 2018). Although originally developed for dynamic response, the model is relevant for quasi-static loading because it reduces the wall response to a dominant deformation coordinate. By neglecting inertia, the response can be interpreted as a nonlinear resistance governed by the mid-height displacement. This provides a mechanical motivation for the NODE model, where the OOP force capacity response is learned as a function of U_{mid} and the static wall parameters, similar to the formulation in this model.

1.6.3 Finite Element Analysis

The finite element (FE) analysis serves two main purposes that are vital to the development of the assessment model: (i) to reproduce the governing mechanical response of URM walls subjected to out-of-plane loading, and (ii) to generate a parametric database of results for subsequent use in the ML models. To conduct the numerical analysis, the finite element method (FEM) is employed. FEM provides an approximate solution to the governing equilibrium problem by discretizing the physical domain into a finite set of subdomains, where the response is represented within an approximated function space (Ciarlet & Lunéville, 2023). The resulting finite-dimensional problem is then solved using numerical schemes which, under certain conditions, converge towards the solution of the original continuous problem, providing results that closely represent the physical behavior of the wall.

The FE–analysis in this thesis was conducted using the software Abaqus. Abaqus is a general-purpose FE program capable of solving nonlinear structural problems, including both geometric and material nonlinearities (Dassault Systèmes SIMULIA, 2024). The software was applied to simulate the OOP response of URM walls, following the guidance provided in the Abaqus documentation for nonlinear analysis. As part of this approach, aspects such as time increment size control, element type selection, and convergence studies are considered in order to obtain reliable numerical results.

To enable efficient modeling in Abaqus, the workflow was automated through Python scripting in Abaqus/CAE. The scripting interface in Abaqus provides direct access to all necessary modules: model creation, meshing, job submission, and post-processing functionality, through Python commands (Dassault Systèmes SIMULIA, 2017). A parametric script was employed to systematically vary the wall parameters (e.g. geometry, boundary conditions and material properties), run the analysis and extract relevant data such as the peak force capacity. In addition, the `abqpy` Python package was used to support the scripting workflow by access to tools for Abaqus/Python development, improving robustness and efficiency when generating and executing simulation scripts.

1.6.4 Machine Learning Models for Predicting OOP Force Capacity

The purpose of the machine learning (ML) models was to provide generalized predictions of the OOP force capacity of URM walls. The ML models where trained on a parametric dataset generated from verified FE simulations. This simulation-driven modeling approach follows a workflow commonly used in engineering applications, where FE analyses are performed to generate structured data. The ML models use this data and enables fast predictions without costly repetitions of numerical simulations.

Three different ML-based modeling strategies are used in this work. The first model is a Multi-Layer Perceptron (MLP), which was trained to predict the scalar peak force capacity Q_{peak} (Goodfellow et al., 2016). The second model is a Neural Ordi-

nary Differential Equation (NODE), which was used to predict the nonlinear force-displacement response of the wall until the peak force is reach. The third model was symbolic regression, which was applied to discover explicit analytical expressions from the simulation-based data. Together, these models provide different methods of predictive capability and results.

The MLP was used as a direct regression model for predicting the peak OOP force capacity (Nielsen, 2015). The input to the model is a vector of static wall and boundary parameters and the output where the OOP peak force capacity. The performance of the MLP is evaluated using parity plots and performance measures. Since the MLP only predicts the peak value, it provides a computationally efficient model for estimating the maximum OOP capacity, but it does not describe the full deformation response.

The second model is a Neural Ordinary Differential Equation (NODE), which was used to model the force-displacement response rather than only the peak value (Worsham & Kalita, 2025). In this approach, the lateral force was treated as a state variable evolving with respect to the mid-height displacement U. The NODE learns the derivative of the response from the FE-generated curves and reconstructs the full force-displacement curve through numerical integration. This allows the model to capture not only the peak force, but also the stiffness development up until the peak force capacity.

The third approach was symbolic regression, applied to obtain explicit mathematical expressions from the FE simulation dataset (Kronberger et al., 2025). Symbolic regression is used to discover equations that relate the input parameters to either the peak force capacity. The purpose of this step is to increase the ability to interpret the model prediction by translating the data-driven response into closed form expressions that can be compared with analytical arching theory.

Together, the three models provide complementary descriptions of the OOP response. The MLP gives a fast prediction of the peak capacity, the NODE describes the full nonlinear response curve, and symbolic regression provides interpretable expressions that can be used to analyze the governing relationships in the dataset.

1.7 Research Novelty

The novelty of this thesis lies in the application of machine learning to predict the OOP response of URM walls. Previous studies within the project umbrella "Blast capacity of masonry walls" have mainly focused on experimental investigations of URM walls under quasi-static, impact, and blast-related loading conditions (Godio & Flansbjer, 2025a; Godio & Flansbjer, 2025b; Godio et al., 2021, 2023a, 2023b). These studies form the experimental foundation for understanding the physical behavior, failure mechanisms, and influence of boundary conditions.

While machine learning has previously been applied to masonry structures for damage assessment (Rezaie et al., 2022) and prediction of in-plane strength (Na-

jafgholipour & Zohrei, 2026), its application to OOP URM wall response remains limited. This thesis is therefore positioned between the existing experimental research on OOP masonry response and the emerging use of machine learning for structural assessment. The contribution is the combination of parametric FE modeling with ML-based prediction of OOP capacity and response, extending the available research toward a data-driven modeling framework for URM walls.

1.8 Project pipeline

The overall project can be viewed in sections that flow through a pipeline to produce several results. The overall pipeline considering all sections is visualized in an interactive flow chart in Figure 1.1.

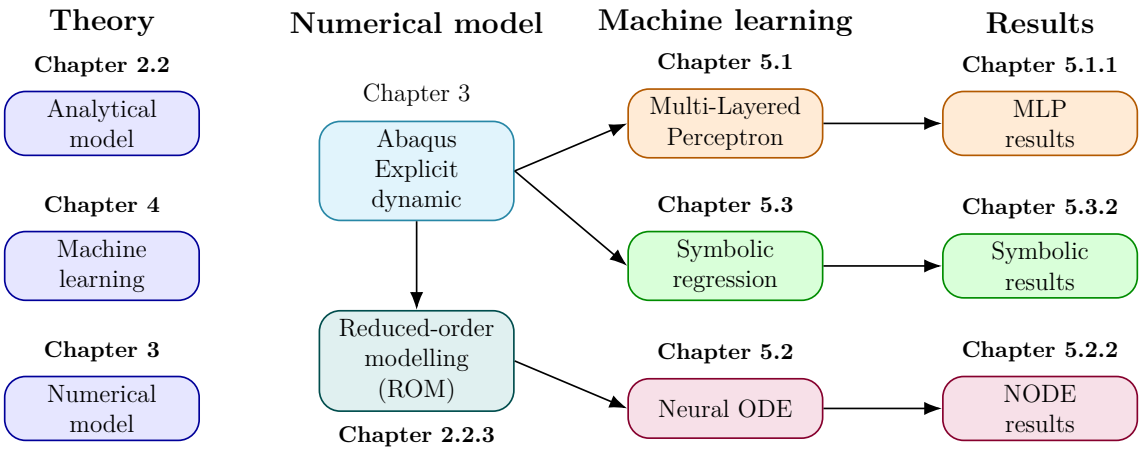


Figure 1.1: Overview of the theoretical, numerical, and machine-learning workflow.

2 Masonry Walls

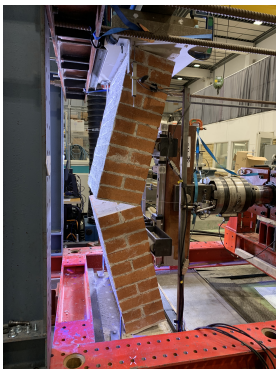
Within this chapter, the failure mechanisms and analytical expressions currently used to approximate the OOP peak force capacity of URM walls are examined. Furthermore, the limitations of these analytical expressions are highlighted in reference with experimental test data.

2.1 Failure mechanisms in masonry walls

Most of the current literature and design codes treat OOP failure of URM walls with the failure mode of bending (European Committee for Standardization, 2006; Martens et al., 2017). A recent test study investigated how URM walls behaved in out of plane loading (Godio et al., 2023a). Focusing on the arching effect in determining the wall strength and failure mechanism. The study conducted tests on eleven different walls consisting of both single wythe (115 mm) and double wythe (245 mm) specimens under a quasi-static four point bending. The masonry was built between two concrete supports and subjected to three different support conditions, simply supported, rigid and non rigid to simulate a wide range of building conditions. With different vertical compressions applied to the walls before the OOP load.

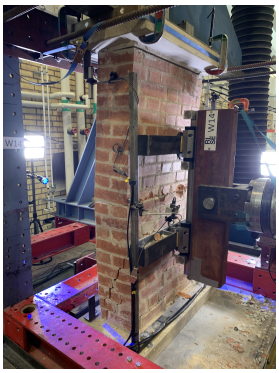
The experiments concluded that the support conditions of the walls dictated in large part the failure mechanism and the structural performance. Where simply supported walls developed no arching effect and failed through bending characterized by a formation of three hinges. These walls exhibited a lower OOP capacity while reaching the maximum OOP load with small deformations and large displacements until complete failure. Rigid supports however experienced a large arching caused by the vertical restraint, and shear failures was observed in the double wythe walls, However an increase was observed in the OOP strength regardless of failure mode. The two walls with a non-rigid support condition showed a intermediate failure mode, both diagonal shear cracks close to the supports and a characteristic cracks along the middle of the wall typical for bending failures. Where findings concluded that the support conditions directly influences both the failure mode and the OOP capacity of the walls.

Wall thickness also played an important role to the OOP strength of the walls. Single-wythe walls was dominated by bending failure regardless of its support condition or its vertical compression. Double-wythe walls had a more varied response and with rigid support conditions generated high arching effect resulting in brittle shear failures, with a small displacement until total collapse. Figure 2.1 below depicts three failure modes of the double-wythe walls from the experiment.



(a) Wall W5 with a bending failure.

(b) Wall W12 with a shear failure.



(c) Wall W14 with a bending/shear failure.

Figure 2.1: Different failure modes for a URM wall, as prescribed by experimental tests at RISE (Godio et al., 2023a).

An overview of the wall failure modes, boundary conditions, and OOP peak force capacities from the experimental campaign is presented in Table 2.1. Of the tested specimens, only three consisted of single-wythe walls, while eight were double-wythe walls. For the double-wythe walls, pure bending failure was observed exclusively in simply supported specimens, whereas walls exhibiting shear or combined shear–bending failure were associated with partially or fully rigid boundary conditions.

Table 2.1: Overview of the walls presented in the experimental tests (Godio et al., 2023a).

	Wall	Failure	Boundary condition	Force Capacity [kN]
Single wythe	W1	Bending	Rigid	26
	W2	Bending	Simply Supported	8
	W3	Bending	Rigid	36
Double wythe	W4	Bending	Simply Supported	45
	W5	Bending	Simply Supported	43
	W6	Shear	Rigid	148
	W7	Shear	Rigid	168
	W8	Bending	Simply Supported	74
	W12	Shear	Rigid	152
	W13	Bending/Shear	Non-Rigid	133
	W14	Bending/Shear	Non-Rigid	118

The experimental campaign shows that many analytical expressions and design code assumptions are not sufficient to describe the OOP behavior of URM walls which will be highlited in later sections. Most models assume a bending failure mechanism, but the tests show that the failure mode depends strongly on the support conditions. The OOP strength of the wall also changes depending on the level of confinement and vertical compression applied, regardless of the failure mode.

2.2 Analytical models on the out-of-plane strength

An URM wall exposed to a distributed load from the OOP have several failure mechanisms depending on its geometrical and material properties (European Committee for Standardization, 2006). The most common one is a bending failure which can be modeled as a three point plastic hinge model. Research on these types of failures have been done for many years. As the structural response of the wall is governed by many variables of non-linear nature, the force capacity is difficult to model from static analysis. Despite the challenge, there are studies made to result in analytical design models for masonry structures to be used in design. Since then there have been efforts to build upon this research and advance the expression further in order to capture the mechanics of the three point plastic hinge bending mechanism.

2.2.1 Eurocode 6 – Lateral loading

The model which are most commonly used today is the design equation for lateral loading in Eurocode 6 (EC6). This model was designed from the research in (European Committee for Standardization, 2006) which adopts the theory of a plastic hinge between two rigid blocks resulting in a bending failure. The loading conditions and mechanism are shown in Figure 2.2. The shape of the plastic hinge mechanism at the ultimate limit state is assumed and a rotational equilibrium is taken per rigid body around the center point C.

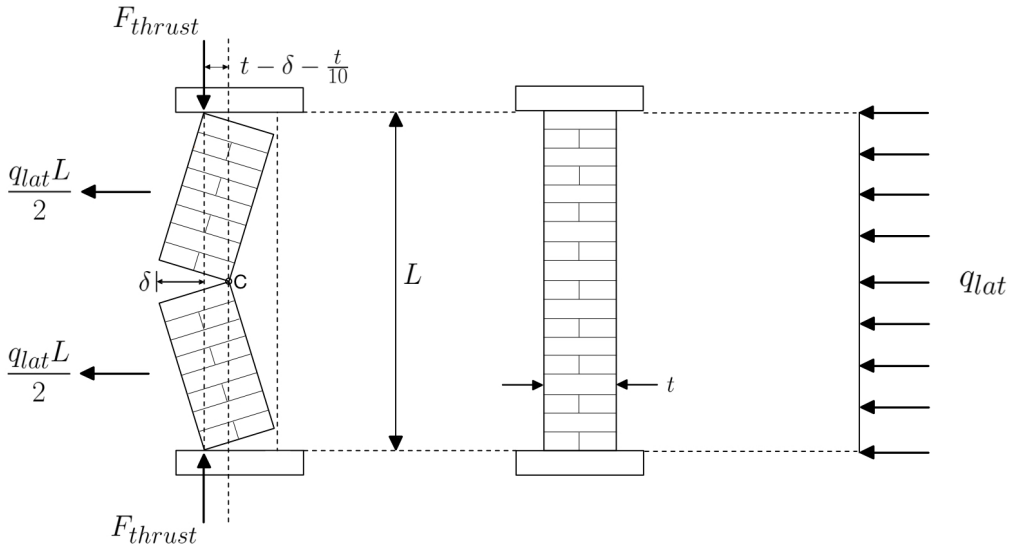


Figure 2.2: Structural model for the derivation of the force capacity for out-of-plane loading in EC6.

The conditions to be met for the plastic hinge rigid body mechanism are explicitly listed in EC6 (European Committee for Standardization, 2006) which have to be

identified to the specific loading case before hand. As the equilibrium is maintained by the arch thrust, generated as an axial thrust force F_{thrust}, the upper and lower boundary supports have to be rigid enough to support this force. The full derivation of the equilibrium equation to the final expression is presented in Derivation 2.2.1.

Derivation 2.2.1: Rotational equilibrium around point C from Figure 2.2

The rotational Equilibrium taken from the loading case in Figure 2.2 is considered (European Committee for Standardization, 2006). The distributed load q_{lat} is divided into reaction forces at hid height of each rigid block. The generated thrust force F_{thrust} is acting on C though an eccentricity. The full equilibrium expression becomes

$$\frac{q_{lat}L}{2}\left(\frac{L}{2}\right) - 2F_{thrust}\left(t - \frac{t}{10} - \delta\right) = 0.$$

Then by isolating the lateral distributed load q_{lat} in the left hand side. The expression for the total OOP force capacity is taken as

$$q_{lat} = \frac{8F_{thrust}\left(t - \frac{t}{10} - \delta\right)}{L^2}$$

To further evolve the expression, the assumptions on the generated thrust force F_{thrust} gives the expression in Equation 2.2 which is inserted into the equilibrium equation

$$q_{lat} = \frac{8\left(1.5\, f_k\, \frac{t}{10}\right)\left(t - \frac{t}{10} - \delta\right)}{L^2}$$

The assumption is done that for slenderness λ) values < 25 the deflection δ can be neglected and therefore set as zero. The rest of the formula is then multiplied together to form the expression

$$q_{lat} = 1.08\, f_k\, \left(\frac{t}{L}\right)^2.$$

For simplicity purposes the EC6 derivation also round the value of 1.08 down to 1. Resulting in the simple expression of

$$q_{lat} = f_k\, \left(\frac{t}{L}\right)^2.$$

The calculation of the thrust force F_{thrust} is done on the assumption that the maximum thrust force, that is able to be generated in the URM wall, has developed when the OOP force force capacity reaches its peak (Martens et al., 2017). From this assumption, the thrust force can be taken as the amount of force, over the plastic hinge depth, which corresponds to the stress magnitude of the compressive strength

of masonry. The plastic hinge depth is assumed to be $\frac{1}{10}$ of the total thickness of the URM wall. The compressive strength of masonry is obtained through Equation 2.1

$$f_k = K\,(f_{k,brick})^{\alpha} \cdot (f_{k,mortar})^{\beta} \tag{2.1}$$

Variables

- K - is the constant taken from table 3.3 in (European Committee for Standardization, 2006).
- $f_{k,brick}$ - is the compressive strength of brick.
- $f_{k,mortar}$ - is the compressive strength of mortar.
- α - is given as 0.7
- β - is given as 0.3

After the compressive strength of masonry has been obtained it can be used to approximate the maximum thrust force F_{thrust}. It is taken as the force required to crush the masonry unit at the depth of the plastic hinge. The full expression is shown in Equation 2.2.

$$F_{thrust} = 1.5\, f_k\, \frac{t}{10} \tag{2.2}$$

Variables

- t - is the total thickness of the masonry wall.
- f_k - is the characteristic compressive strength of the masonry.

Using the assumptions made in EC6 and following the steps in Derivation 2.2.1, the final expression of the OOP peak force capacity is obtained as Equation 2.3. The expression can also be accompanied by a safety factor γ, set after assessment done by an engineer.

$$q_{lat} = f_k \left(\frac{t}{L}\right)^2 \tag{2.3}$$

The adopted design method from EC6 has many limitations in its derivations. Firstly, it is based on tests in lateral loading done in 1976 (Martens et al., 2017). Secondly, the derivation is done upon many assumptions which are are uncertain from case to case. The width of the plastic hinge is standardized to $\frac{1}{10}$ of the total wall thickness, but could vary form wall to wall. It is also assumed that the maximum thrust force have been developed at the time of the peak lateral load, which also can vary from case to case. Further limitation include boundary condition dependencies (Swedish Institute for Standards (SIS), 2012). The peak force capacity can only be derived from a fixed support scenario as the design model requires an arch thrust, so for simply supported cases the model cannot be used. Lastly, the final expression is produced by neglecting the effect of the deflection for certain wall types.

To test the accuracy of the EC6 Equation 2.3, it has been used to approximate the peak force capacity of the specimens presented in Section 2.1. The results are presented in Figure 2.3.

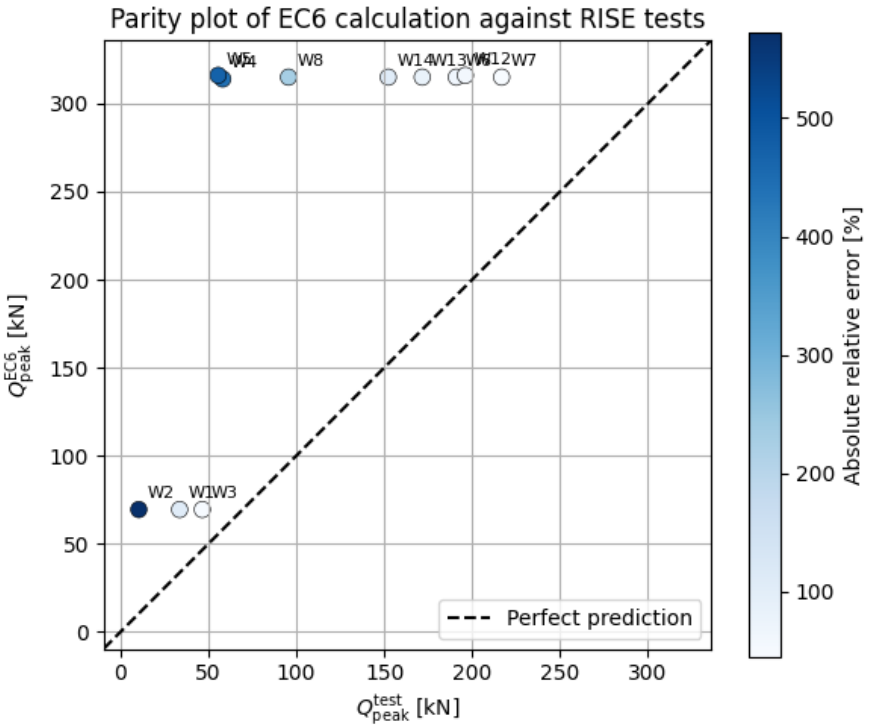


Figure 2.3: Parity plot between the experientially obtained peak force and the peak force calculated in accordance with EC6.

It should be noted that some of the measured points correspond to loading cases where the boundary condition is simply supported. Here, no thrust force can be developed and the EC6 formula should not be used. In other cases the specimens exhibit failures other than bending, which makes it difficult for the formula to capture the true peak force capacity.

2.2.2 Martens and Vermeltfoort lateral loading model

The lateral peak force capacity model derived by Martens and Vermeltfoort is a continuation of the EC6 model in Equation 2.3 (Martens et al., 2017). One consideration is the effect of the deformation which is derived using a detailed approximation of the arch rise. This is performed using the evolution of depth in the compression zone, linked to the shape of the thrust line in the URM wall. The shape depends on the loading condition, where a horizontal point load results in a triangular shape, and a lateral distributed load gives the thrust line a parabolic shape. An illustration of the two different thrust line geometries is shown in Figure 2.4.

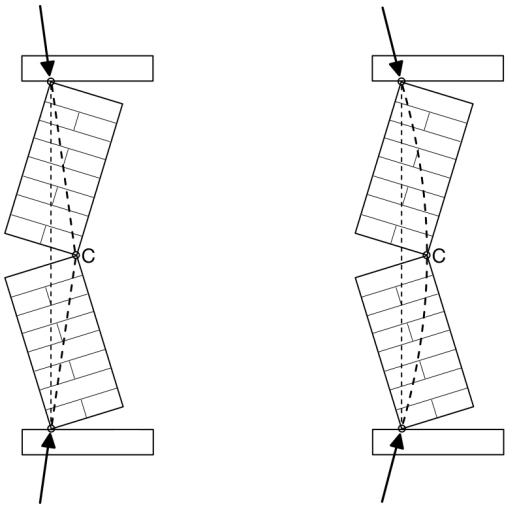
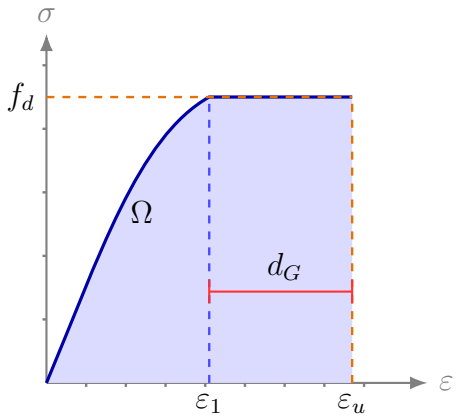


Figure 2.4: Two different thrust line geometries showing a triangular (left) and parabolic (right) shape depending on the loading condition.

The rise of the arch is defined as the distance from the vertical loading lined to point C in Figure 2.4 (Martens et al., 2017). The load bearing capacity is reached when the thrust force F_{thrust} reaches the maximum load baring capacity of the cross-section at mid-height or supports. The thrust force is taken from equilibrium and the maximum load bearing capacity is defined in Equation 2.4.

$$N_{Rd} = \psi \, x \, f_k \tag{2.4}$$

- x is the depth of the compression zone.
- ψ is the area ratio of Ω and the orange rectangle in Figure 2.5.
- f_k is the compressive strength of masonry.



σ–ε-diagram	ψ	δ_G
Linear	0.500	0.333
Parabolic-rectangular	0.810	0.416
Rectangular stress block	1.00	0.500

Figure 2.5: Definition of the parameters ψ and d_G, of the stress–strain area and $\delta_G = d_G/\varepsilon_u$.

Based on the definitions in Figure 2.5 a linear elastic definition of the peak force capacity can be derived (Martens et al., 2017). The derivation, similar to EC6, neglects the deflection of the URM wall and only considers the rise of the arch. The full derivation is presented in Derivation 2.2.2.

Derivation 2.2.2: Martens linear elastic arching resistance

By neglecting the deflection of the wall, the lever arm from the thrust line to the center becomes $t - 2\delta_G x$. Inserting this into the equilibrium equation and isolating the thrust force F_{thrust} the expression becomes.

$$F_{thrust} = \frac{qL^2}{8(t - 2\delta_G x)}$$

The peak force will correspond to the lateral load that is resisted by the maximum load bearing capacity of the cross-section N_{Rd} from Equation 2.4. Inserted into the expression and isolating the lever arm it becomes.

$$\frac{qL^2}{8xf_k} = x(t - 2\delta_G x)$$

To find the peak force the derivative of the right hand side with respect to x is taken and put equal to zero, the resulting right hand side becomes.

$$t - 4\delta_G x = 0 \quad \rightarrow \quad x = \frac{t}{4\delta_G}$$

By inserting this expression into the equilibrium equation the full expression is derived to.

$$q = \frac{8\psi f_d}{L^2}\frac{t}{4\delta_G}\left(t - 2\delta_G \frac{t}{4\delta_G}\right)$$

By simplifying the equilibrium equation the resulting expression becomes.

$$q = \frac{\psi f_k}{\delta_G}\left(\frac{t}{L}\right)^2$$

Approximating the solution by using the shape of the linear elastic stress-strain geometries, corresponding to linear, parabolic-rectangular and rectangular, the solutions are given as.

$$q = 1.5 f_d \left(\frac{t}{L}\right)^2$$

$$q = 1.95 f_d \left(\frac{t}{L}\right)^2$$

$$q = 2.0 f_d \left(\frac{t}{L}\right)^2$$

The linear elastic approximations of the peak force is given from the geometrical simplifications on the area under the stress-strain curves (Martens et al., 2017). Depending on the shape being linear, parabolic-rectangular or rectangular there are three outcomes to the final expression. These are listed in Equation 2.5.

$$q = 1.5 f_d \left(\frac{t}{L}\right)^2 \quad q = 1.95 f_d \left(\frac{t}{L}\right)^2 \quad q = 2.0 f_d \left(\frac{t}{L}\right)^2 \tag{2.5}$$

Masonry is not infinitely stiff, so a fully linear elastic formulation would not capture the real behavior of the URM wall (Martens et al., 2017). To capture the nonlinear mechanism of the three-point hinge, the model interprets this as contraption of the thrust line. The increased axial stress is the result of the strain caused by this contraption. When the stress reaches the maximum load bearing capacity, the peak

force is reached. The derivation of peak force for a non linear mechanism is shown in Derivation 2.2.3.

Derivation 2.2.3: Martens nonlinear arching resistance

The contraption of the thrust line evolves from an initial length s_0, which corresponds to the initial arch rise f_0 of the wall. The compression of the wall is related to the increase in arch rise $\Delta f = f_0 - f_1$ which defines the steepness of the parabolic shape through the coefficient a. The formula for the length of the initial thrust line and its evolution are given as.

$$s_0 = \frac{1}{4a_0}\left[a_0 L\sqrt{a_0^2 L^2 + 1} + \operatorname{arcsinh}(a_0 L)\right] \qquad a_0 = \frac{4f_0}{L^2}$$

$$s_1 = \frac{1}{4a_1}\left[a_1 L\sqrt{a_1^2 L^2 + 1} + \operatorname{arcsinh}(a_1 L)\right] \qquad a_1 = \frac{4f_1}{L^2}$$

The evolution of the contraption is related to the strain by taking the relative shortening of the thrust line with respect to its initial length. The stress is given through the constitutive relationship using Young's modulus. The axial stress is then obtained combining the stress with area ratio of the stress-strain curve ψ and the depth of the compression zone x.

$$\varepsilon_{av} = \frac{s_0 - s_1}{s_0} \qquad \sigma_{av} = E\varepsilon_{av} \qquad N_d = \psi x \sigma_{av}$$

Because of the deformation of the wall the wall thickness of the plastic hinge at mid-height is reduced. Because if the reduction the thickness the compression zone changes with the contraption of the wall in accordance with $x = \frac{t-\Delta f}{4\delta_G}$. Introducing this to Equation 2.4 the axial compression and the load bearing capacity becomes.

$$N_d = \psi\frac{t - \Delta f}{4\delta_G}\sigma_{av} \qquad N_{Rd} = \psi\frac{t - \Delta f}{4\delta_G}f_k$$

Due to the nonlinearity of the formula the change in arch rise Δf corresponding to $N_d = N_{Rd}$ have to be found numerically or graphically. When the correct rise of the arch have been found the load bearing capacity is inserted to the equilibrium equation, giving the expression for the peak force capacity as.

$$q = \frac{8N_{Rd}(t - 2\delta_G x)}{L^2}$$

The peak force capacity is the amount of force required to make the resistance of the wall reach its load bearing capacity (Martens et al., 2017). For the nonlinear three point hinge mechanism this can be calculated using Equation 2.6.

$$q_{lat} = \frac{8N_{Rd}\,(t - 2\delta_G x)}{L^2} \tag{2.6}$$

The final model expression corresponding to both the linear elastic and the nonlinear equations can be used similar to the EC6 formula. When tested on the experimental data from RISE it showed similar deviations to the EC6 formula (Godio et al., 2023a). This gives further motivation to the fact that a new model for assessing the OOP peak force capacity have to be produced.

2.2.3 Edri and Yankelevsky reduced order model on the out-of-plane dynamic response

The model proposed by Edri and Yankelevsky reduces the response of an arching URM wall to an equivalent single-degree-of-freedom representation (Edri & Yankelevsky, 2018). Although their formulation was developed for dynamic loading, the central reduced-order idea is also relevant for quasi-static loading. The full wall deformation is not described by all nodal degrees of freedom, but is instead represented by a dominant deformation mode and a generalized displacement, chosen as the mid-height out-of-plane displacement U.

The response of the wall is approximated by a three-hinge-arch bending mechanism (Edri & Yankelevsky, 2018). To capture this behavior the mechanism is represented by an equivalent nonlinear mass-spring system. The nonlinear resistance, associated with the arching action and generated thrust force, is modeled as a nonlinear spring K_e together with the applied med height displacement U. The reduced model also includes a mass term to capture the full dynamic behavior by modeling the inertial effects. A model schematic of the three-hinge bending mass-spring system is shown in Figure 2.6.

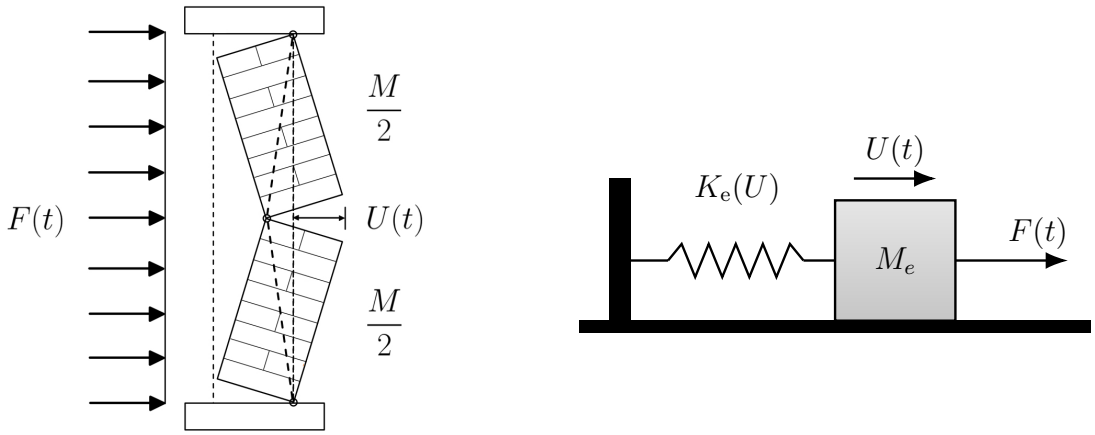


Figure 2.6: Reduced-order interpretation of arching action. The thrust-line assumptions describe the arching mechanism governing the wall response, while the dynamic reduced-order model represents this behavior through the generalized mid-height displacement $U(t)$ and the nonlinear spring $K_e(U)$.

For a quasi-static loading case the inertial effects, governed by the mass and accel-

eration, are neglected (CAE Assistant, 2025). As the loading rate is governed by a constant velocity, the displacement field is linear over the entire loading time. Because of this, the driving trajectory of the system becomes the displacement instead of time. The ROM of a quasi-static loading case is presented in Figure 2.7.

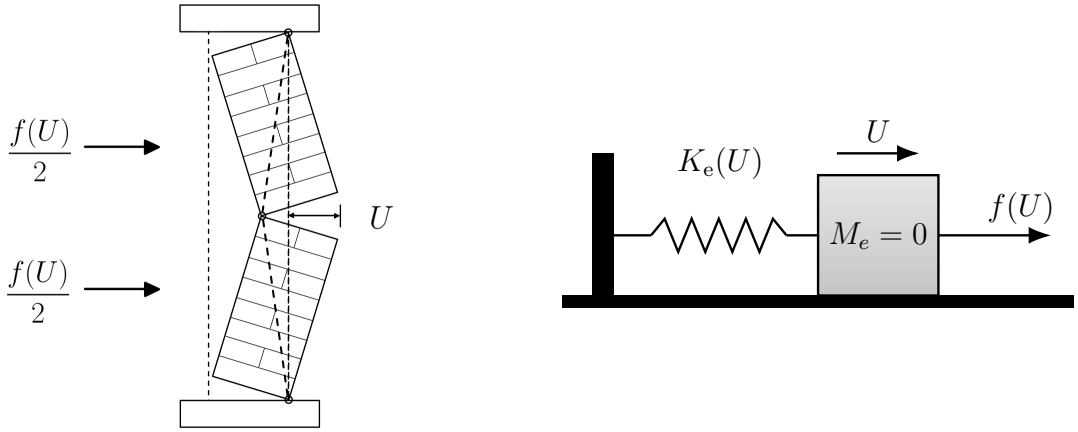


Figure 2.7: Reduced-order interpretation of arching action. The thrust-line assumptions describe the arching mechanism governing the wall response, while the quasi-static reduced-order model represents this behavior through the generalized mid-height displacement U and the nonlinear resistance $K_e(U)$.

As the studied models are evaluated in a quasi-static four point bending loading scenario, the ROM model in figure 2.6 is used during this thesis (Godio et al., 2023a). The main dataset is produced from high fidelity simulations using ABAQUS 2024 (Dassault Systèmes Simulia Corp., 2024). For computational efficiency, the adopted machine learning models presented in later chapters will adopt similarities to the quasi-static ROM model.

3 Numerical model

This chapter presents the numerical model developed in ABAQUS, including the model setup and the parameters varied to evaluate their influence on the lateral capacity of URM walls. The model is based on and validated against the experimental campaign conducted at RISE (Godio et al., 2023a), with numerical predictions compared to the experimental results to assess model accuracy.

It should be noted that the analytical expressions presented in Section 2.2 are derived for uniformly distributed loading over the wall specimen. Consequently, an investigation into the differences between distributed loading and four-point bending, including their influence on the failure response, is presented in Appendix H. Owing to the numerical challenges associated with developing a stable model under distributed loading, the four-point bending configuration was adopted for the present study. The results obtained from the numerical model are subsequently used to establish the framework and define the input–output relationships for the machine learning (ML) models.

To accurately represent the experimental walls described in (Godio et al., 2023a), a detailed micro-model of both the mortar and bricks is developed to capture material damage and its effect on the out-of-plane (OOP) capacity. The explicit solver in ABAQUS is used for this purpose, as it has proven more effective than the implicit solver in handling contact interactions, damage, and failure mechanisms (Dassault Systèmes Simulia Corp., 2024), while also requiring fewer computational resources. However, the explicit solver is primarily developed for modeling dynamic procedures involving high-impact events. Therefore, when modeling a quasi-static four-point bending test, specific strategies are required to ensure accurate results when using an explicit approach. These strategies are presented and discussed in the following sections.

3.1 Modelling of Quasi-Static Loading Conditions

The explicit dynamic solver in Abaqus is used to evaluate models subjected to dynamic effects (Dassault Systèmes Simulia Corp., 2024). The solver approximates the solution to a full dynamic problem by evaluating each part of the equation of motion (EOM). These parts include how the stiffness varies with the internal displacements as $\mathbf{Ku}$. Further, it accounts for the applied damping of the system, if there is any present, and how it evolves with the internal velocities in line with $\mathbf{V\dot{u}}$. Lastly, the dynamics are also effected by the inertial forces which correlates to how the mass is effected by the internal acceleration according to $\mathbf{M\ddot{u}}$. The sum of all these effects gives the external excitation force f. The full form of the EOM is presented in Equation 3.1.

$$\mathbf{K}(t)\mathbf{u} + \mathbf{V}(t)\dot{\mathbf{u}} + \mathbf{M}(t)\ddot{\mathbf{u}} = \mathbf{f}(t) \tag{3.1}$$

As the current studied model is a four-point bending test, the structural response of the specimen is better captured in a quasi-static environment (Dassault Systèmes Simulia Corp., 2024). In Abaqus, this is achieved by reducing the injected inertial forces in the dynamic approach until that part of the equation can be neglected. This is done by reducing the internal accelerations to zero, giving the resulting mass effect as $\mathbf{M}\ddot{\mathbf{u}} = 0$. To further simplify the model and reduce computational time, the damping of the system have also been overlooked, by not including any damping in the global response of the system, giving a resulting damping part of $\mathbf{V}\dot{\mathbf{u}} = 0$. Per this definition, the equation to solve for each quasi-static state is given in Equation 3.2 defined as the resistance captured by the total reaction force.

Another key aspect of quasi-static loading is that the trajectory path variable changes (Dassault Systèmes Simulia Corp., 2024). In a dynamic system, the accelerations are used to obtain the velocities by integration. The velocities are in turn integrated themselves to reach the displacement field of the system. In the quasi-static model, the applied velocity is prescribed with a constant speed which removes the effect of accelerations. As a result of the prescribed velocity, the displacement field becomes linear. This gives the possibility to model the system with a linear displacement trajectory over the testing period from u_0 to u_f.

$$\mathbf{K}(u)\mathbf{u} = \mathbf{f}(u) \tag{3.2}$$

To achieve a model with quasi-static response there are a range of conditions that have to be met (CAE Assistant, 2025). The techniques used to meet these conditioned is presented in the next section with a more detailed outline of the approach presented in Appendix A and B.

3.2 Development and Setup of the Finite Element Framework

The main purpose of the numerical model is to create a large set of results for the OOP capacity for URM walls with different parametric changes to the model. Both globally depending on boundary conditions, vertical compression and geometric shape, as well as local parameters such as part dimensions, Young's modulus, and compressive and tensile strength. To allow for this large set of parameters while limiting computational time, the model was developed as a wall strip in two dimensions, enabling a large number of results to be obtained efficiently.

The model is designed in two steps, the pre-compressive step where the vertical force is applied and the OOP step where the lateral load is introduced. The vertical load is applied at the top support, with addition of the self weight. The self-weight of the model is calculated by the volumetric parameters of the units (support,brick

and mortar) and there densities which are added to the vertical load. The load is applied in a smooth way over the entire step time to reduce the injected inertial forces which suppress kinetic energy of the system (Natário et al., 2014). A deeper investigation in selection of the step time for the pre-compressive step can be seen in Appendix B.1.

For the OOP step a velocity is added instead of a force. Applying a force in the OOP step makes it hard to control the displacement because of the unstable loading rate. Velocity with a constant speed removes this problem, as long as its applied slow enough (Dassault Systèmes Simulia Corp., 2024), a further look into the selection of the velocity can be seen in Appendix B.2. As for the pre-compressive step the first 10 seconds of the addition of the velocity is done in a smooth manner, ramping it up until it reaches the selected velocity and keeping it constant.

The support conditions (simply supported, non-rigid, and rigid) are represented by a wire connected between the top of the wall and a fixed point above it (Dassault Systèmes Simulia Corp., 2024). A stiffness value is assigned to the wire, where a low stiffness represents the simply supported condition and a high stiffness represents the rigid condition. Intermediate stiffness values correspond to the non-rigid support condition. The fixation point is activated only during the OOP analysis step to ensure that no additional stiffness is introduced to the wall during the pre-compression step.

A mesh sensitivity study has also been made to ensure that the model is stable and produces accurate results, a deeper investigation has been made on the single-wyhte wall and it can be seen in Appendix C. A small mesh increases the computational time of the models, keeping it as coarse as possible is off essence to reduce the computational time.

3.2.1 Modeling of the Single-Wythe Wall Specimen

The single-wythe wall was constructed in the experimental campaign (Godio et al., 2023a) using a running bond between successive layers. However, since the numerical model is two-dimensional, the running bond is not represented. Observations from the experiments showed that failure of the single-wythe walls was governed by bending, characterized by hinge-like behavior. Cracks developed in the upper, middle, and lower regions of the wall, primarily along the mortar–brick interfaces. This indicates that the running bond had a limited influence on the overall OOP capacity of the wall.

The single-wythe wall model in Abaqus is shown in Figure 3.1. Yellow crosses indicate the locations of the reference points, where history outputs, boundary conditions, and applied forces are defined. Reference points are connected to the model using kinematic coupling to the closest surface (Dassault Systèmes Simulia Corp., 2024).

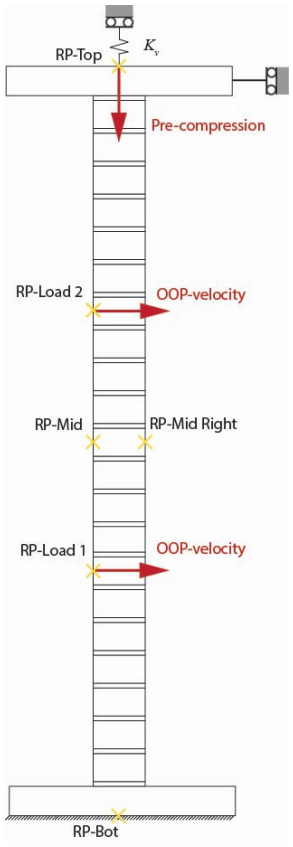


Figure 3.1: Single wythe wall as modeled in ABAQUS, where reference points (RP) are indicated with yellow crosses and the force and velocity is indicated by red arrows.

The bottom reference point (RP-bot) is fully constrained, with boundary conditions applied in both the x-direction (horizontal) and y-direction (vertical), as well as in rotation. In contrast, the top reference point (RP-top) is constrained only in the x-direction and in rotation. The pre-compressive load is applied at RP-top during the pre-compression step.

RP - Load 1 and 2 is where the OOP-velocity is added and as previously discussed this is added as slow as possible to keep the model as close to a quasi-static problem as possible. RP-Mid and RP-Mid right is used to retrieve the displacement of the wall to match where the experimental campaign retrieved its values.

Between each layer, a contact interaction is defined. The normal behavior is modeled using hard contact, while the tangential behavior is defined using a penalty formulation with a friction coefficient of one (Dassault Systèmes Simulia Corp., 2024), effectively preventing slip between the interfaces. A damping coefficient of 1 is introduced to reduce oscillations at the interfaces. These interface properties represent a contact with no tensile strength in the normal direction, allowing the layers to separate easily, while providing high shear resistance in the horizontal-direction, thereby preventing relative sliding. The selected contact parameters have been calibrated and validated, and will be further discussed in the following section.

A model for a double-wythe wall was also developed. However, an insufficient number of reproducible results were obtained for the model to be considered sufficiently reliable for inclusion in the main discussion. As such, it is not presented here but is instead provided in Appendix E.

3.2.2 Concrete Damaged Plasticity Model for Masonry: Formulation and Implementation

The material model employed to simulate failure due to crushing and cracking is the Concrete Damaged Plasticity (CDP) model. This model is widely used in the literature for the analysis of masonry and concrete structures (Avasthi et al., 2026; Yucel, 2025). The CDP model combines isotropic damaged elasticity with pressure-dependent plasticity to represent the nonlinear behavior of quasi-brittle materials. It is capable of capturing the distinct failure mechanisms associated with tensile cracking and compressive crushing, as well as stiffness degradation during the damage process.

The formulation is based on a modified Drucker-Prager-type yield criterion (Dassault Systèmes SIMULIA, 2024), which allows the model to account for the influence of confinement on material strength. This is particularly relevant for masonry structures, where multiaxial stress states and interaction between compressive and tensile failure modes play a significant role. In addition, the incorporation of damage variables enables the progressive degradation of stiffness, providing a realistic representation of crack initiation and propagation. Parameters controlling the Drucker-Prager condition are presented in Table 3.1 and are standard values in modeling of masonry walls (Lourenço et al., 1995).

Table 3.1: Parameters introduced into the CDP model to govern the modified Drucker–Prager plasticity yield criterion

Parameter	**Value**
Dilation angle ψ	31
Eccentricity ϵ	0.01
Ratio of biaxial to uniaxial compressive strength f_{b0}/f_{c0}	1.16
Shape factor K_c	2/3
Viscosity parameter μ	1×10^{-5}

In this study, the CDP model is adopted to describe the nonlinear response of the masonry constituents with a Hill-type criterion, as presented in Lourenço et al. (1995). Implementation is done with a number of simplifications to ease the material model into the parametrization of the model for the construction of the stress-strain relationships. The Hill-type criterion is simplified by defining characteristic stress levels, namely the initial strength (σ_i), mean strength (σ_m), and residual strength (σ_r), as well as a simplified relation for the corresponding inelastic strain levels in

compression. An overview of these assumptions is given in Equations 3.3. Here, the peak stress (σ_p) represents the maximum compressive strength of the material.

$$\sigma_i = \frac{\sigma_p}{3} \tag{3.3a}$$

$$\sigma_m = \frac{\sigma_p}{2} \tag{3.3b}$$

$$\sigma_r = \frac{\sigma_p}{10} \tag{3.3c}$$

$$\kappa_m = 3\kappa_p \tag{3.3d}$$

To ensure numerical stability and realistic crack propagation, the post-peak softening branch is defined as a gradual decay rather than an abrupt drop to zero stress. Consequently, the softening branch diminishes into a small residual stress (σ_r) in the softening regime, which is consistent with common implementations of the CDP model (Dassault Systèmes SIMULIA, 2024).
Beyond the residual stress level, a cutoff strain must be defined to establish a limit of the softening curve. This is introduced through a tolerance parameter.

$$\varepsilon = \mathrm{tol}\,(\sigma_m - \sigma_r) \tag{3.4a}$$

$$m = \frac{2(\sigma_m - \sigma_p)}{\kappa_m - \kappa_p} \tag{3.4b}$$

$$\kappa_r = \kappa_m + \frac{\sigma_m - \sigma_r}{m} \ln\left(\frac{\varepsilon}{\sigma_m - \sigma_r}\right) \tag{3.4c}$$

An assumption for the peak strain κ_p is required, which is obtained from literature and discussed further in Section 3.4. Based on the above simplifications, the compressive inelastic stress-strain behavior of the material can be described by the piecewise function given in Equations 3.5.

$$\sigma(\kappa) = \sigma_i + (\sigma_p - \sigma_i)\sqrt{2\frac{\kappa}{\kappa_p} - \left(\frac{\kappa}{\kappa_p}\right)^2} \qquad \kappa < \kappa_p \tag{3.5a}$$

$$\sigma(\kappa) = \sigma_p + (\sigma_m - \sigma_p)\left(\frac{\kappa - \kappa_p}{\kappa_m - \kappa_p}\right)^2 \qquad \kappa_p \leq \kappa \leq \kappa_m \tag{3.5b}$$

$$\sigma(\kappa) = \sigma_r + (\sigma_m - \sigma_r)\exp\left(\frac{\kappa - \kappa_m}{\sigma_m - \sigma_r}\right) \qquad \kappa_m < \kappa \leq \kappa_r \tag{3.5c}$$

The fracture energy for the compression G_c is defined as the area under the inelastic stress-strain curve until the residual stress level, A simplified expression is defined.

$$G_c = \frac{\pi}{4}\sigma_p\kappa_p + (\kappa_m - \kappa_p)\left(\sigma_p + \frac{1}{3}(\sigma_m - \sigma_p)\right) \tag{3.5d}$$

Continuum models such as the CDP model used in this study suffer from mesh dependency in failure analysis (de Borst, 1991), particularly when strain-softening and crack development are involved. To overcome this issue, the fracture energy (crack-band) method proposed by Hillerborg et al. (1976) is adopted. This approach introduces an internal finite element length, ensuring that the energy dissipated during softening remains independent of the mesh size.
The previously introduced critical inelastic strain κ_p is normalized with respect to the finite element characteristic length h, which gives the expression for fracture energy condition as

$$G_c = h \int_0^{\kappa_p} \sigma(\kappa)\, d\kappa \tag{3.6}$$

where G_c is the compressive fracture energy and $\sigma(\kappa)$ is the inelastic stress-strain relation.
To ensure mesh objectivity, a scaling factor is introduced such that the numerical energy dissipation matches the material fracture energy:

$$\alpha = \frac{h \int_0^{\kappa_p} \sigma(\kappa)\, d\kappa}{G_c} \tag{3.7}$$

The internal variable κ is then rescaled according to

$$\kappa_{\text{reg}} = \frac{\kappa}{\alpha} \tag{3.8}$$

For the tensile behavior, the fracture energy is assumed to be $\frac{1}{10}$ of the regularized compressive fracture energy. This assumption forms the basis for the definition of the inelastic tensile stress–strain relationship. Consequently, the residual strength is taken as $\frac{1}{10}$ of the peak tensile strength.
The tensile response of brick and mortar is characterized by brittle behavior which leads to the material exhibiting a rapid loss of load-carrying capacity immediately after reaching the peak stress. The softening behavior is described using an exponential decay law, ensuring a realistic representation of post-peak degradation and consistency with the prescribed fracture energy (Lourenço et al., 1995). Equations and assumptions for formulating the tensile stress-strain relationship can be observed in Equations 3.9 below.

$$\sigma_r = \sigma_p \frac{1}{10} \tag{3.9a}$$

$$G_t = \frac{1}{10} G_c \tag{3.9b}$$

$$\kappa_r = \frac{G_t}{h(\sigma_p - \sigma_r)} \tag{3.9c}$$

$$\sigma_t(i) = \sigma_r + (\sigma_p - \sigma_r) \cdot e^{(-(\frac{\kappa(i)}{\kappa_r}))} \tag{3.9d}$$

The general shape of the curves of the material model can be seen in Figure 3.2. Where the assumptions of the initial stress, peak stress, mean stress and residual

stress are highlighted with colored dots in the compressive graph. For the tensile graph the peak stress and residual stress is instead highlighted.

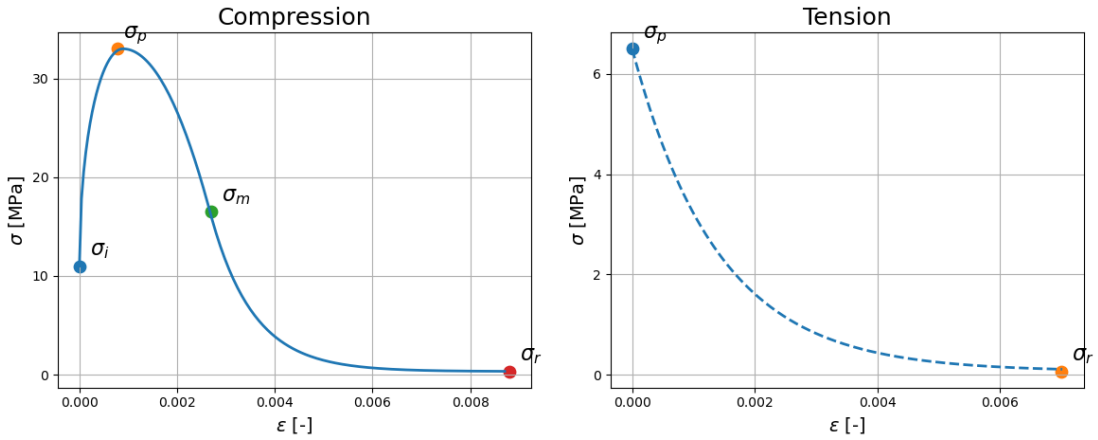


Figure 3.2: Stress–strain relationships in compression and tension for the adopted material model, with markers indicating the characteristic values used in the formulation.

This formulation provides a computationally efficient and numerically stable representation of the nonlinear behavior of masonry materials, while retaining the essential characteristics governing cracking, crushing, stiffness degradation, and energy dissipation (Lourenço et al., 1995).

3.3 Validation of the ABAQUS Model Against Experimental Data

An effort is made to validate the numerical model results against experimental data in order to increase confidence in both the simulation results and the developed ML models. This validation process further supports the potential use of the ML models future evaluations and predictive assessments against physical data.

For the validation of the single-wythe numerical model, three walls (W1, W2 and W3) from the experimental tests, seen in Section 2.1, are used to validate the single-wythe specimens (Godio et al., 2023a). The walls have dimensions of 1535 mm in height, 775 mm in width, and 115 mm in thickness. The bricks measure 250 × 115 × 62 mm, and the mortar joints have a thickness of 10.5 mm. The material parameters for the slab, brick, and mortar presented in the experimental report are listed in Table 3.2 and are assumed to be the same for all walls. In the absence of complete experimental data, parameters were adopted from values reported in the literature and further calibrated through iterative fitting of the obtained results.

Table 3.2: Material parameters used for validation, with assumed values indicated in italics.

Material	f_c [MPa]	f_t [MPa]	E [GPa]	ρ [kg/m^3]	k [-]	ν [-]
Brick	33	6.5	*7.5*	1715	*0.0068*	*0.1*
Mortar	0.85	0.38	0.238	1570	*0.01*	*0.15*
Slab	-	-	30	2500	-	*0.2*

The spring stiffness and pre-stress of the walls are presented in Table 3.3. The experimental values were scaled for use in the ABAQUS model, as the 2D model has a thickness of 1 mm, whereas the experimental walls have a width of 775 mm. Therefore, the experimental values were divided by the wall width.

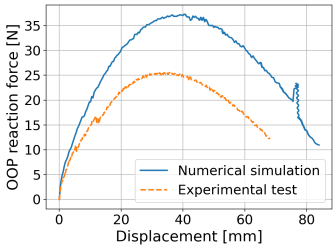
Table 3.3: Applied loads and spring stiffness used in the validation study, and the calibrated values used in ABAQUS.

	Experimental		*Scaled*	
Wall	Pre-stess [kN]	K$_s$[kN/mm]	Pre-stress [N]	K$_s$[N/mm]
W1	17.4	126.7	22.5	163.5
W2	17.4	0	22.5	0
W3	35.7	87.3	46.1	112.6

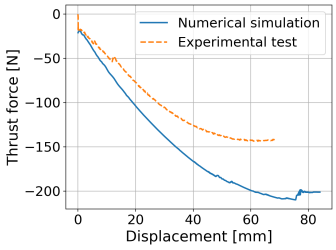
Validation for the first wall (W1) is illustrated in Figure 3.3 where the numerical model is shown in blue and the experimental reference data as a dotted orange line in all three figures. Figure 3.3a depicts the force capacity in the OOP direction versus the displacement at the middle of the wall. The numerical model shows a similar trend to the experimental results, and the displacement at the maximum peak force matches well. However, the maximum value is significantly higher in the model than in the experiment.

Figure 3.3b shows the thrust force versus the displacement at the middle of the wall. A similar trend is observed, but with the same overestimation of the maximum value. Additionally, the thrust force only begins to decline. This may indicate that the wall response in the model has not yet reached the corresponding stage, as the force capacity is still partly maintained in Figure 3.3a.

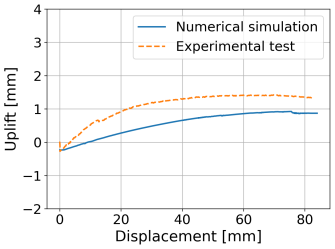
The last plot, Figure 3.3c, shows the uplift at the top of the slab. The progression of the uplift is slower in the numerical model than in the experimental results, although the final value matches well. However, a less satisfactory result is obtained in terms of the maximum force capacity, as the numerical model overestimates the peak value by approximately 44 %.



(a) OOP reaction force vs horizontal displacement.



(b) Thrust force vs horizontal displacement.



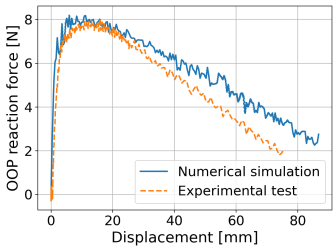
(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure 3.3: Results from the numerical model (blue) compared to the experimental results (orange) for the W1 wall.

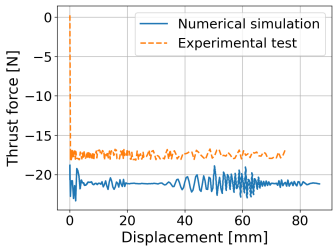
The validation for wall W2 is presented in Figures 3.4. In Figure 3.4a), the maximum OOP force capacity, which is the primary quantity of interest in this study, is well captured by the numerical model. The overall shape of the curve also shows good agreement with only minor deviations.

The vertical reaction force, shown in Figure 3.4b), remains essentially constant due to the simply supported boundary conditions, indicating that no significant arching action develops. This is consistent with the experimental observations. The observed difference is a result of the fact that the gravitational force is added together with the pre-compressive force in the first step, therefore increasing the initial force.

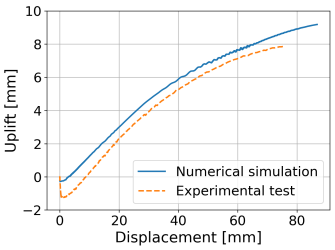
Figure 3.4c) illustrates the uplift at the top slab. Here, a good agreement between the numerical and experimental results is also observed.



(a) OOP reaction force vs horizontal displacement.



(b) Thrust force vs horizontal displacement.



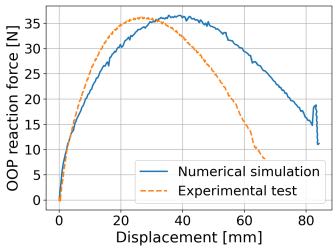
(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure 3.4: Results from the numerical model (blue) compared to the experimental results (orange) for the W2 wall.

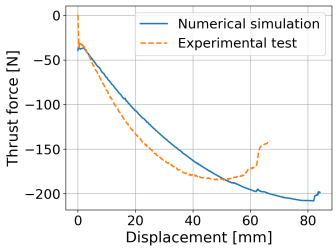
The OOP force response in Figure 3.5a shows a similar maximum value and general shape, although the experimental curve develops faster and reaches its peak earlier. The exact cause is difficult to determine, but one possible explanation is a different elastic modulus in the experimental setup, resulting in a stiffer wall. Since the same material properties were assumed for all walls, no trial-and-error calibration was

performed, as the validation for wall W2 showed good agreement and the adopted value is well supported by the literature seen in Section 3.4. The thrust force in Figure 3.5b also differs in both shape and peak value, which may be related to the offset in displacement. The uplift development in Figure 3.5c is once again slower than that of the experimental test, but the difference between the final uplift is minimal, with account of the numerical data starting at -0.4 mm.

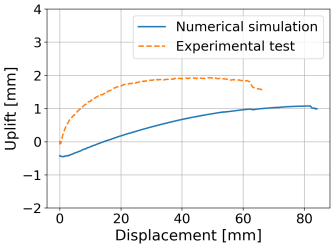
Overall, the validation is considered satisfactory, as the primary objective, capturing the maximum OOP force is achieved.



(a) OOP reaction force vs horizontal displacement.



(b) Thrust force vs horizontal displacement

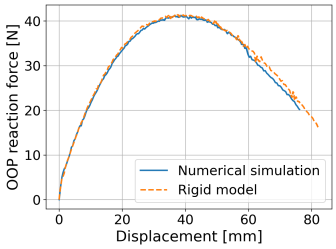


(c) Uplift in the top slab vs displacement in the horizontal direction.

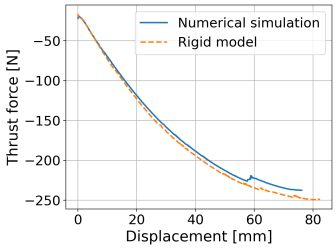
Figure 3.5: Results from the numerical model (blue) compared to the experimental results (orange) for the W3 wall.

The final validation of the numerical model concerns the rigid support condition. In this case, no experimental data is available, and the comparison is instead made between two numerical models. The reference model, shown in Figures 3.5, is defined with a boundary condition fixed in the vertical direction. The alternative model is defined similarly, but with a very large spring stiffness (>500) to approximate rigid support conditions.

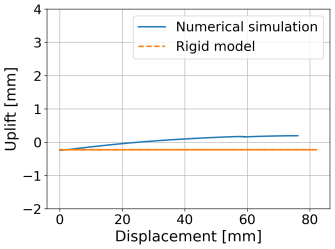
As seen in the graphs, this approach successfully reproduces the behavior of the fixed boundary condition. A small difference can be observed in Figure 3.5c), where a minor uplift remains. However, since the uplift is approximately 0.25 mm, it is small enough to be considered negligible.



(a) OOP reaction force vs horizontal displacement.



(b) Thrust force vs horizontal displacement.



(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure 3.6: Results from the numerical model (blue) with a very high stiffness compared to a different numerical model with rigid top boundary (orange).

For the validation conducted to assess the wall behavior and its force capacity in the OOP direction, two of the three experimental cases show satisfactory results. The first wall (W1), where the numerical model predicts a higher capacity than the measured values, introduces some uncertainty. Internal documents at RISE (Godio & Flansbjer, 2025a), where a simplified 3D model of the wall was developed, also showed an overestimation of the capacity by 28%, compared to 44% in the present micro-model. In terms of absolute values, the measured capacity was 26 kN, while the RISE model predicted 32 kN and the present model 36 kN. This raises some uncertainty as to whether the reported experimental values may be inaccurate or misaligned. Despite this, the micro-model is considered sufficiently accurate in capturing the wall's capacity in the OOP direction.

3.4 Parametrization of the Material Units

Masonry exhibits a wide range of mechanical properties that are strongly influenced by the composition of the brick and mortar. To ensure a reliable numerical model and meaningful results, the parameters must be selected within reasonable ranges based on values reported in the literature, there by accurately representing masonry behavior.

3.4.1 Background to Brick as a Building Material

Brick making has existed for millennia and has continuously evolved over time. The earliest bricks were mud bricks, which were hand-molded and sun-dried. However, these materials were less durable, as weathering processes reduced their longevity, resulting in less permanent structures (Friesem et al., 2011). During the Mesopotamian period, fired bricks were developed, offering improved durability. Their higher production cost meant they were used less frequently and were primarily reserved for important buildings such as religious sites and palaces (Fiala et al., 2019).

In the early modern period (1500–1800), standardization of brick geometry emerged,

while firing techniques remained largely similar to those used in Roman times. It was not until the later modern period that kiln technologies became more standardized, enabling more consistent quality and material properties. Today, bricks are manufactured in a wide variety of shapes, compositions, and material types, including lightweight concrete, lightweight aggregate concrete, and calcium-silicate bricks. Their properties can be tailored, for example by increasing porosity to improve thermal performance or by designing interlocking geometries for construction efficiency. For the purposes of this thesis, the conventional solid fired clay brick is adopted as the reference material for parameterization.

The material properties of solid fired clay bricks are a key factor in the overall strength of masonry structures (Gustafsson, 2023) and must therefore be carefully evaluated to accurately determine the load-bearing capacity of a wall. Common clay bricks are primarily manufactured from fine-grained soils derived from the weathering and erosion of bedrock. Consequently, the mineral composition of the raw material varies depending on its geological origin and geographic location, which in turn significantly influences the mechanical properties and strength of the bricks (Moumni et al., 2024). In addition, manufacturing parameters such as firing temperature and duration in the kiln play a crucial role in determining the final material properties (Karaman et al., 2006), As previously discussed, this variability introduces uncertainty when assessing bricks in heritage structures, where material properties can differ significantly.

3.4.2 Parameterization of the Brick Units

Modern bricks typically exhibit compressive strengths in the range of 5–70 MPa, where the upper limit corresponds to specially engineered units with high cement content (Cabané et al., 2022). In contrast, heritage and ancient bricks found in historical structures may have significantly lower strengths, with reported values as low as 1 MPa (Binici et al., 2020). This highlights the considerable variability in compressive strength across different types of brick. For the purpose of parametrization, a wide range of values is therefore adopted. Additionally, due to the similarity in the ratio of mechanical properties between solid red clay bricks and bricks with high cement content, this range is considered acceptable.

In practice, the tensile strength of bricks is commonly related to their compressive strength. Experimental studies, modeling approaches, and design codes suggest that the tensile strength typically ranges between 10 % and 30 % of the compressive strength (European Committee for Standardization, 2006; Lourenço, 1996). To ensure realistic material parametrization and preserve the inherently weak behavior of bricks in tension, the tensile strength is therefore defined within this proportional range.

The elastic modulus is another key parameter with a wide range of reported values and a significant influence on the OOP displacement behavior of masonry walls.

Similar to tensile strength, the elastic modulus can be correlated with compressive strength. According to European Committee for Standardization, 2006, this relationship for masonry is expressed as $E/f_c = 1000$. In contrast, American guidelines suggest a lower ratio of approximately 550 (Federal Emergency Management Agency, 1997). However, these code-based relationships do not distinguish between the contributions of bricks and mortar independently and are intended for full masonry walls rather than individual units.

In this study, the fraction between E/f_c as found in EC6, is adopted as a basis for establishing a reasonable relationship between compressive strength and elastic modulus. The range of ratios investigated is summarized in Table 3.4.

Table 3.4: Ratio between Young's modulus and compressive strength of bricks.

Brick type	$f_{c,b}$ [MPa]	E_b [MPa]	$E_b/f_{c,b}[-]$	Reference
Clay	34.5	12940	375	Kim and Lim, 2025
Clay	36.3	12700	350	
Clay	32.8	12326	376	
Clay	18.3	4416	241	Shi et al., 2021
Clay	16.9	3850	228	
Clay	19.8	8697	439	Li et al., 2025
Shale	25.3	9575	378	
Coal Gangue	13.8	6635	481	
Clay	10.1	2001	198	Singh and Munjal, 2017
Clay	8.2	1116	136	
Concrete	16.7	5954	357	
Clay	17.7	5,300	299	Kaushik et al., 2007

Based on the data presented in Table 3.4, the ratio between elastic modulus and compressive strength is found to vary approximately between 100 and 500. Although values outside this range are reported in the literature, these are typically specially engineered bricks for low elastic value, hand molded or outliers compared to other reported values (Nazimi et al., 2024). Such cases are excluded in this study to maintain a representative and practically relevant parameter range.

For the calibration of the CDP model, the peak inelastic strain is used to define the damage evolution curve as described in Subsection 3.2.2. Based on the literature review presented in Nazimi et al., 2024, an empirical relationship for the peak strain as a function of the stiffness-to-strength ratio has been proposed and is given in Equation 3.10:

$$\varepsilon_p = 0.9 \left(\frac{E_b}{f_{c,b}} \right)^{-0.9} \tag{3.10}$$

To account for the inherent variability and uncertainty associated with this empirical

expression, and as also indicated in Nazimi et al., 2024, the coefficient (0.9) is varied within a range of 0.7 to 1.1. This allows the model to capture a broader spectrum of possible material responses. A overview of the parametric range of the mechanical properties of brick can be seen in Table 3.5.

Table 3.5: Parametric range of brick.

Parameter	Range	Unit
$f_{c,b}$	$5 - 70$	MPa
$f_{t,b}$	$(0.1 - 0.3) \times f_{c,b}$	MPa
E_b	$(100 - 500) \times f_{c,b}$	MPa
ε_p	$(0.7 - 1.1)(E_b/f_{c,b})^{-0.9}$	–

As the compressive strength varies, the tensile strength, elastic modulus, and peak strain are defined as functions of this value through the ratios given in the table. This ensures a consistent relationship between the mechanical properties across the considered parameter space.

3.4.3 Background to Mortar as a Building Material

Mortar is composed of aggregates, a binder, water, and, when necessary, additives. These components form a workable paste that hardens over time and bonds masonry units together (Gustafsson, 2023). The type of mortar used is therefore of great importance for the overall strength of masonry walls. Historically, different types of binders have been used in mortar, with the most common being:

- Clay
- Gypsum
- Lime
- Cement
- Lime–cement

Clay mortar is the oldest type and was primarily used alongside the development of sun-dried bricks. Although it is uncommon today, it is still used in some parts of the world. Lime mortar has been one of the most widely used binders. It is produced by limestone calcination to form quicklime, which is then hydrated and subsequently hardens through a carbonation process by reacting with carbon dioxide (Manoharan & Umarani, 2022). Historically, two main types of lime mortar have been used. Air lime mortars harden through exposure to carbon dioxide, while hydraulic lime mortars contain pozzolanic additives that enable hardening through chemical reactions with water, in addition to carbonation (Malathy et al., 2023). Cement-based and lime–cement mortars were introduced in the mid-19th century with the development of Portland cement. Lime–cement mortars became commonly used in structural elements of buildings, such as pillars.

The wide variety of mortars used in masonry buildings necessitates a clear differentiation of their mechanical properties. To address this, a comprehensive literature review of the mechanical properties of various mortar types has been conducted.

It is noteworthy that lime mortars, which are commonly found in heritage structures, generally exhibit lower compressive strength than cement-based mortars. Accordingly, the parametrization focuses on lime-based mortars and lime–cement mortars with low cement content in order to better represent the mortars typically found in heritage buildings.

3.4.4 Parametrization of the Mortar Units

The ratio for the tensile strength and elastic modulus are determined using the same procedure as for the bricks, where tensile strength replaces the elastic modulus. The values reported by Soleymani et al., 2022 are identified as outliers, particularly with respect to the measured elastic modulus ratio, and are therefore excluded from the determination of the parameter boundaries. In addition, to maintain a reasonably confined parametric range for the elastic modulus ratio, the values reported by Garijo et al., 2018 are also excluded. Based on these considerations, the selected ratio range for the elastic modulus is defined between 50 and 1000. The collected data are summarized in Table 3.6.

Table 3.6: Mechanical properties of different kinds of mortar.

Mortar type	$f_{c,m}$ [MPa]	$f_{t,m}$	$f_{t,m}/f_{c,m}$ [-]	E_m [MPa]	$E_m/f_{c,m}$ [-]	ε_p [-]	Reference
Lime:Sand	13.3	2.9	0.22	683	51	-	Costigan et al., 2015
Lime:Sand	10	1.3	0.13	401	40	-	
Lime:Sand	2.7	0.6	0.22	197	73	-	
Cement:Lime:Sand	22.2	4.8	0.22	1587	71	-	
Lime:Sand	1.1	-	-	4652	4191		Soleymani et al., 2022
Cement:Lime:Sand	3.0	-	-	6465	2140	-	
Gypsum	5.2	-	-	4953	953	-	
Lime:Sand	3.1	0.41	0.13	-	-	-	Lee et al., 2024
Lime:Sand	2.3	0.27	0.12	-	-	-	
Lime:Sand	3.4	0.43	0.13	-	-	-	
Lime:Sand	3.2	0.39	0.12	5000	1562	-	Garijo et al., 2018
Lime:Sand	1.7	0.24	0.14	2800	1647	-	
Lime:Sand	1.7	0.24	0.14	2800	1647	-	
Cement:Sand	42.1	-	-	17522	416.2	0.0042	Kim and Lim, 2025
Cement:Lime:Sand	31.2	-	-	19832	636	0.0023	
Cement:Sand	12.7	-	-	8989	708	0.0036	
Cement:Sand	3.1	-	-	545	176	0.0087	Kaushik et al., 2007
Cement:Sand	20.6	-	-	3750	182	0.0185	
Cement:Lime:Sand	15.2	-	-	3300	217	0.0270	
Lime:Sand	1.55	0.27	0.17	1463	943		Bokan Bosiljkov et al., 2023
Lime:Sand	1.91	0.22	0.12	1713	896	-	
Lime:Sand	1.24	0.32	0.26	1097	885	-	
Lime:Sand	1.83	0.91	0.48	1841	1006	-	
Lime:Sand	0.65	0.32	0.49	673	1068	-	

The full parametric range selected based on the literature review is presented in Table 3.7. Where the compressive strength of the material is dictating the tensile

strength and elastic modulus based on a ratio that has been previously described. It should be noted that, in addition to the mortar properties, the joint thickness is varied from 5 mm to 20 mm in order to examine how the strength depends on the mortar-to-brick ratio (Yadav & Pal, 2023). Mortar joint thickness can vary significantly between different structures and even in walls within the same structure.

Table 3.7: Parametric range of mortar.

Parameter	Range	Unit
f_c	$0.50 - 10$	MPa
f_t	$(0.1 - 0.5) \times f_c$	MPa
E	$(50 - 1000) \times f_c$	MPa
ε_p	$0.002 - 0.03$	–
h_m	$5 - 25$	mm

3.4.5 Global Parametrization of the Numerical Model

The global parameters of the parametrization of the model include the pre-compressive force, stiffness of the spring and the height of the wall. Each been decided by different measures. Starting with the pre-compressive force range, the maximum value has been concluded by calculations according to "Eurocode 1: Actions on Structures" (2002) and assumed values for mass and geometry for the walls and floors, calculations can be seen in Appendix D.

Stiffness of the spring is decided iteratively based on a comparison between a fully rigid boundary condition at the top and when the stiffness reaches the same result. This comparison has been made in full in Section 3.3 with the conclusion that the model will be fully fixed with a spring stiffness of 500 N/mm.

The maximum wall height considered in the parametric study was set to 4 m (Gustafsson, 2023), while the minimum height was set to approximately 1.3 m, 3 bricks lower than the experimental specimen. The wall height was increased in increments of three bricks to ensure that the OOP loading remained applied at the center of a brick unit and that the loading configuration remained consistent throughout the wall height, with loading points located at one-third and two-thirds of the wall height. Larger walls contain a greater number of bricks and mortar joints, resulting in an increased number of finite elements and consequently longer computational times for each model. For walls with approximately twice the height of the experimental wall, the computational time was roughly doubled.

3.4.6 Execution and Results From the Parametric Study

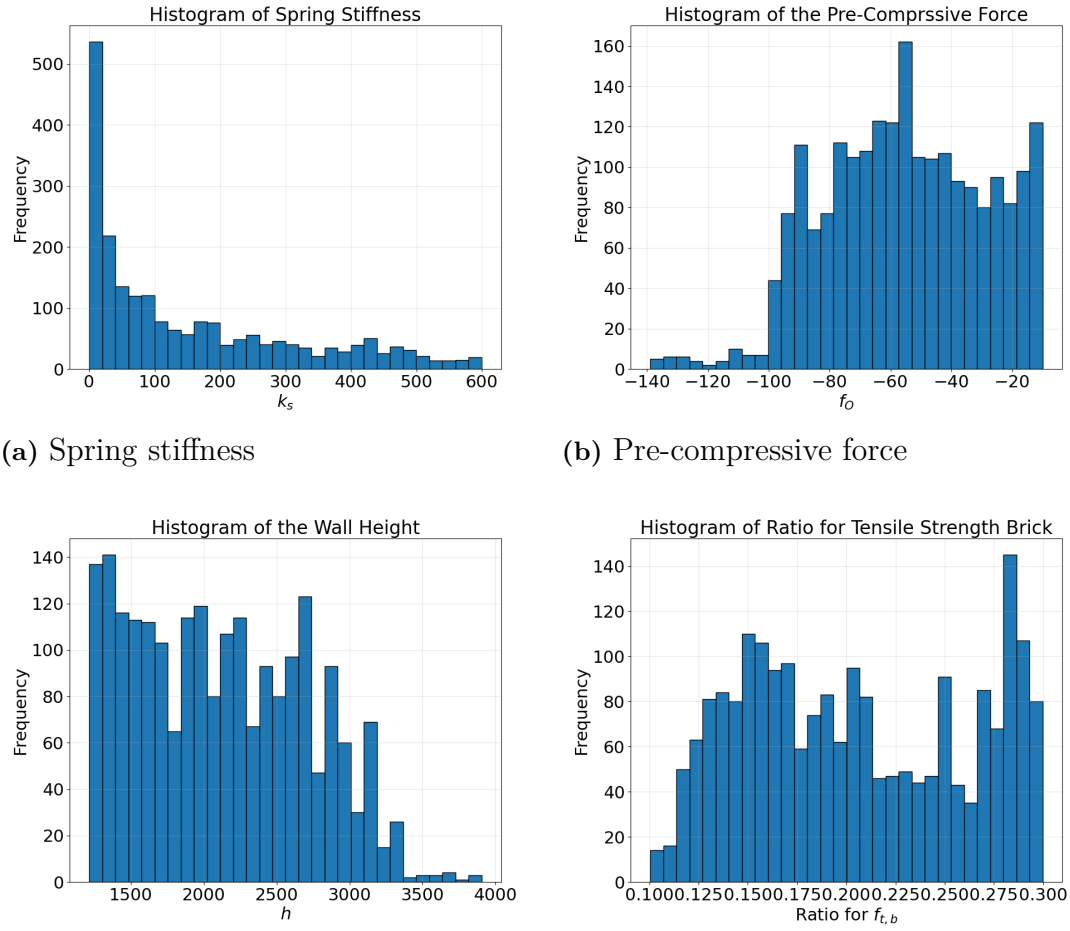
The numerical models were generated and executed automatically through scripting in python, after which the results and the input parameters were stored in a dataset. The normal run time of each single-wythe model was around 12 minutes and creation of the entire dataset of 2000 different models resulted in a total run time of approximately of 24000 minutes or 16.7 days. After running the models, all of the results with a ratio between the kinetic and internal energy exceeding 5 % where removed as they where deemed as to unstable and a typical cutoff for quasi-static solutions in ABAQUS (Dassault Systèmes Simulia Corp., 2024). The result file comprises of all parameters varied and constant. The results retrieved from the file are shown in the list below.

- Q_{peak} - Maximum force capacity
- F_{thrust} - Thrust force at maximum lateral strength
- $F_{thrust,peak}$ - Maximal Thrust force
- $F_{thrust,peak,ad}$ - Thrust force with removed vertical force
- u_{max} - Maximum displacement horizontal in the middle of the wall
- u_Q - Displacement horizontally at maximum lateral force

Beyond this, the OOP force–displacement, thrust–displacement, and uplift–displacement responses are stored for each simulated iteration, together with a screenshot of the final deformed state. The screenshot provides a means of identifying the wall failure mechanism, however as it is captured at the end of the simulation, it may not necessarily correspond to the failure state associated with peak force capacity.

The parametric range was established using a Monte Carlo approach with uniformly distributed sampling (Bonate, 2001). Certain adjustments were required to accommodate the dataset. The histogram in Figure F.3a, showing the frequency of samples on the vertical axis and the spring stiffness on the horizontal axis, illustrates an increased occurrence of lower stiffness values. This was introduced to better represent walls with simply supported boundary conditions and low rigidity.

The distributions of the pre-compressive force, wall height, and mortar compressive-to-tensile strength ratio are also presented in the histograms shown in Figure 3.7. These parameters exhibit distributions that are generally uniform; however, they are not perfectly uniform because the parametric ranges were expanded during the generation of the dataset, resulting in some values occurring less frequently than others.



(a) Spring stiffness

(b) Pre-compressive force

(c) Wall height

(d) Brick strength ratio

Figure 3.7: Distributions of the selected parametric values.

For taller walls, fewer values are included in the dataset, an reason might be the kinetic/internal energy threshold is reached more frequently. Consequently, these cases were removed from the dataset. Histograms of all parametrized values are provided in Appendix F. Correlations between the different parameters within the selected parametric range were evaluated using a Spearman correlation matrix (Spearman, 1904),It shows whether increasing one variable generally causes another variable to increase, decrease, or remain unrelated. The coefficient ranges from $-1 \leq \rho \leq 1$, where a value of -1 indicates a strong negative correlation, meaning that an increase in one variable generally results in a decrease in the other variable. Conversely, a value of 1 represents a strong positive correlation, while a value close to 0 indicates little to no monotonic relationship between the variables. The resulting correlation matrix for the selected parameters can be seen in Figure 3.8.

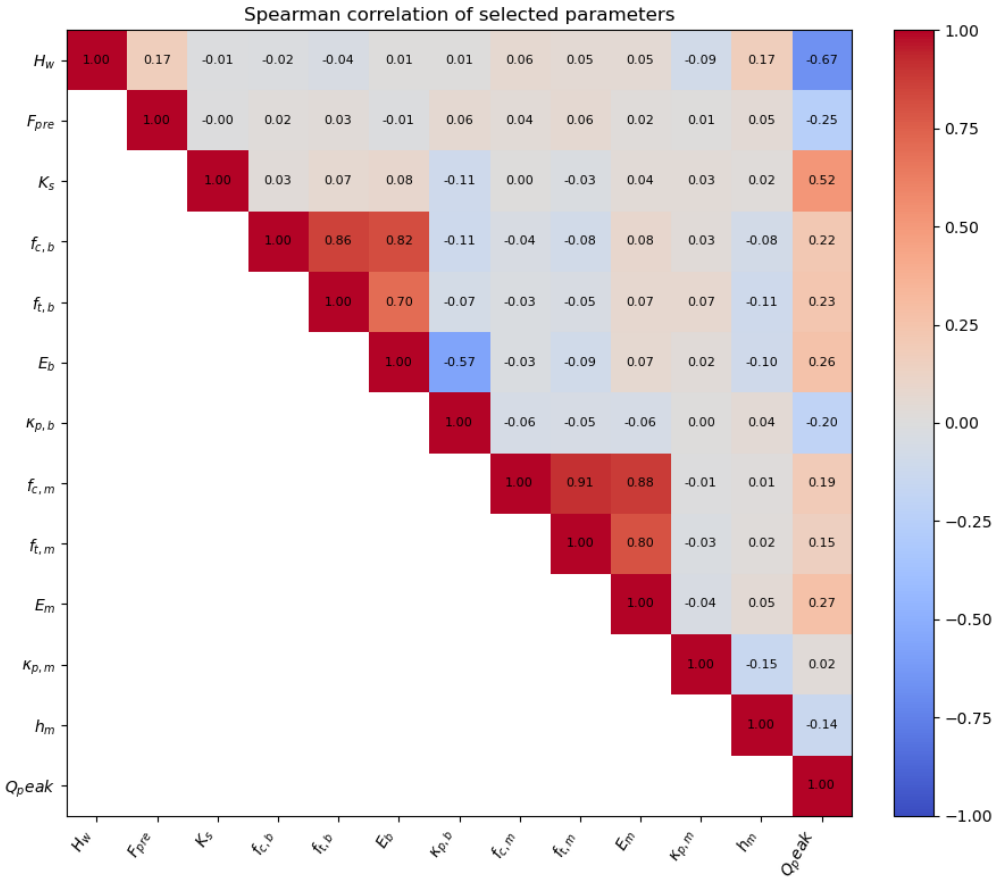


Figure 3.8: Spearman correlation matrix.

Based on the correlation matrix, the strongest correlations are observed between the compressive strength, tensile strength, elastic modulus, and peak strain parameters for both brick and mortar. This is expected, as the tensile strength, elastic modulus, and peak strain are derived from the compressive strength, as described in Subsections 3.4.2 and 3.4.4.

Beyond these relationships, the correlations between the peak force capacity (Q_{peak}) and the input parameters are of greater interest. The strongest correlations with Q_{peak} are observed for the wall height, spring stiffness, pre-compression, and the elastic modulus of both the mortar and brick materials. Additional correlations are present, however, these are the most significant and relevant within the investigated parametric range.

4 Machine Learning

This section outlines the foundational theory and core concepts of machine learning relevant to this thesis. It traces the progression from early definitions of the perceptron to the architecture of modern Artificial Neural Networks, and explores the training paradigms used to fit these models to intended datasets. Focus is maintained specifically on the theoretical components directly implemented in the proposed model.

4.1 Introduction to Machine Learning

A machine learning model uses mathematical methods and algorithms to automatically learn patterns from a training dataset rather than relying on explicitly programmed rules (Murphy, 2012). During this training process, the model analyzes examples containing input variables, known as features, and their corresponding outcomes, known as targets, to detect structural relationships between them. Once these underlying connections are learned, the model can make decisions and predictions on new data where the output is unknown. Ultimately, the practical value of a model depends entirely on its ability to generalize, meaning it successfully applies the patterns learned from the initial data to make reliable and accurate predictions for new unseen observations, as a model that only succeeds on its training data is not useful in practice (Murphy, 2012).

Ever since the earliest computers were invented, scientists have explored how these machines could retain information and apply it to solve new problems (Mitchell, 1997). A major source of inspiration for this goal came directly from biology, the idea of mimicking the human brain's activity using computational algorithms. This concept was famously pioneered by Warren McCulloch and Walter Pitts in 1943 (McCulloch & Pitts, 1943). In their foundational research, they sought to explain how the nervous system works by modeling the behavior of biological neurons using mathematical logic. Their work demonstrated that networks of artificial neurons could perform logical calculations, successfully reducing the gap between biological brain function and computational algorithms.

Building directly upon this foundational idea of modeling biological brain cells with mathematical logic, the perceptron was developed as the very first type of artificial neuron (Nielsen, 2015). Conceptually, it acts as a simplified, computational version of a real neuron. A perceptron works by taking in multiple distinct inputs, evaluating them, and producing a single output response. To determine this output, the perceptron relies on a built-in threshold rule. It essentially weighs the incoming signals; if the combined strength of these inputs surpasses a predefined threshold, the artificial neuron activates and changes its output. If the threshold is not met, the output remains inactive. The basic structure and decision-making process of

this early artificial neuron can be seen in Figure 4.1.

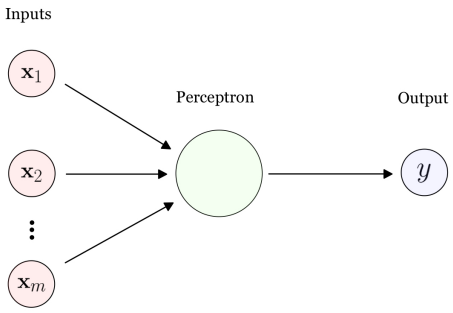


Figure 4.1: Conceptual image of the processing of a perceptron unit with several inputs and one output (Nielsen, 2015).

While the perceptron represented a major breakthrough in modeling neural networks, its rigid, all-or-nothing threshold approach limited its long-term potential (Nielsen, 2015). To overcome this, modern architectures incorporate more sophisticated artificial neurons that utilize smooth, non-linear so called "activation functions". However, defining the architecture of a network only establishes its capacity. For a network to perform specific predictive tasks, the internal parameters must be optimized which are defined by the model's learning paradigm.

4.2 The Learning Paradigm

To enable computers to learn from data has been a topic of interest since the early development of computer science (Mitchell, 1997). The concept of artificial learning is partly inspired by how humans learn, where experience is used to improve performance over time. In machine learning, this idea is translated into mathematical models and learning algorithms. A commonly used definition of machine learning is given by Mitchell (1997), who states that:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

In machine learning, the task T defines the specific objective the model is designed to accomplish (Goodfellow et al., 2016). It is important to distinguish that the task is not the process of learning itself, but rather the concrete problem the model is applied to solve. While machine learning encompasses a wide variety of task categories, this thesis focuses exclusively on the prediction of continuous numerical

values. Consequently, the primary task addressed in this work is *regression*.

In machine learning, regression tasks are specifically designed to predict continuous numerical values based on a given set of inputs (Goodfellow et al., 2016). Mathematically, the objective is to approximate a mapping function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$. This function takes an input vector $\mathbf{x}$ which contains n different variables or features and maps it to a single, continuous target number. To illustrate this within a context, consider the task of determining a beam's design force capacity. In this scenario $\mathbf{x}$ could be material parameters and the output would be the design force of the beam.

The Performance Measure P evaluates how effectively a machine learning model accomplishes its designated task (Goodfellow et al., 2016). To monitor the model's learning progress and determine its accuracy, its performance must be mathematically quantified. The specific metric chosen depends entirely on the nature of the problem being solved.

For regression tasks, where the predicted output is a continuous numerical value in $\mathbb{R}$, performance is evaluated by measuring the distance (or error) between the model's predictions and the actual true values (Murphy, 2012). The most common performance metrics used to quantify this error include: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Coefficient of Determination (R^2), these are listed in Equations 4.1 - 4.4.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.1}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{4.2}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4.3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{4.4}$$

Variables

- N: is the total number of samples evaluated.
- y_i: is the true (observed) value for the i-th sample.
- $\hat{y}_i$: is the predicted value generated by the model for the i-th sample.
- $\bar{y}$: is the mean (average) of all true values in the dataset.

The Experience E characterizes the specific environment in which a machine learning algorithm interacts with and processes information (Goodfellow et al., 2016). This experience is fundamentally shaped by the dataset, which acts as a structured dictionary of individual "examples" that reflect observed behaviors or sequences. In the broader landscape of machine learning, these experiences are typically categorized into two primary categories: *supervised* and *unsupervised* learning.

Supervised learning is what is defined as labeled data which means that it maps input data to specific outputs using predefined input-output pairs (Mitchell, 1997). Through training the model learns the underlying patterns in the the features that correlate to known target values. *Unsupervised* training on the other hand, extracts hidden patterns from raw unlabeled data (Mitchell, 1997). Without target values the model is trained to discover inherent correlations through statistical similarities.

To accurately evaluate a model's ability to generalize, the available dataset cannot be used entirely for learning. Instead, it is partitioned into three independent subsets: a training set, a validation set, and a testing set (Goodfellow et al., 2016). This separation is critical to prevent *overfitting* which occurs when a model memorizes the exact data points of the training set rather than learning the underlying patterns. Consequently, the model will perform exceptionally well on training data but will fail to generalize.

In this thesis, the model operates strictly within a supervised regression paradigm. The dataset is organized into a design matrix $\mathbf{x} \in \mathbb{R}^{m \times n}$ containing m distinct examples, each with n features alongside a corresponding target vector $\mathbf{y}$ of continuous values (Goodfellow et al., 2016). Errors are evaluated across all the previously defined performance metrics in equations 4.1 - 4.4. To effectively capture the highly complex, non-linear relationships necessary to accurately map $\mathbf{x}$ to $\mathbf{y}$, the chosen architecture for this objective is an Artificial Neural Network (ANN).

4.3 Artificial Neural Networks (ANN's)

The initial building blocks of a neural network are *perceptrons* previously mentioned in section 4.1. As also mentioned in the same section they have the drawback of being to simple too solve the non-linear mappings of today's applicable machine learning problems (Nielsen, 2015). This is because they are equipped with a simple step function given in Equation 4.5

$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases} \tag{4.5}$$

Variables

- x: is the input tensor (features).
- w: is the weight.
- b: is an added bias.

To enable a neural network to produce sensitive, gradual changes in its output in response to small variations in input features, the artificial neurons must incorporate continuity (Nielsen, 2015). Rather than being restricted to binary outputs like perceptrons, modern artificial neurons compute continuous values across a specified

range. Mathematically, this process begins by calculating the linear combination, z. For a given input example, z is calculated as the weighted sum of all input features plus a bias term: $z = \sum_{j=1}^{n} w_j x_j + b$, with the summation occurring over the n feature dimensions established at the end of Section 4.2.

Building upon the linear calculation of z, a neural network must introduce non-linearity to successfully map complex data (Goodfellow et al., 2016). Without it, the network would degrade into a simple linear regression model. To overcome this structural limitation, the intermediate value z is passed through an activation function. This critical step provides the non-linear transformation necessary for the artificial neuron to learn and represent complex relationships within the features. This thesis utilizes two primary activation functions: the Rectified Linear Unit ($ReLU$) and the Hyperbolic Tangent ($Tanh$). The graphical representation of the functions is presented in Figure 4.2 and 4.3.

The Rectified Linear Unit $ReLU$ is the standard activation function for the hidden layers of most modern deep learning architectures (Nielsen, 2015). It operates using a straightforward piecewise linear rule defined as $f(z) = \max(0, z)$. This means that it outputs the input directly if positive, else the output is zero. The simplicity makes the function very computational efficient while still contributing to the model learning complex patterns.

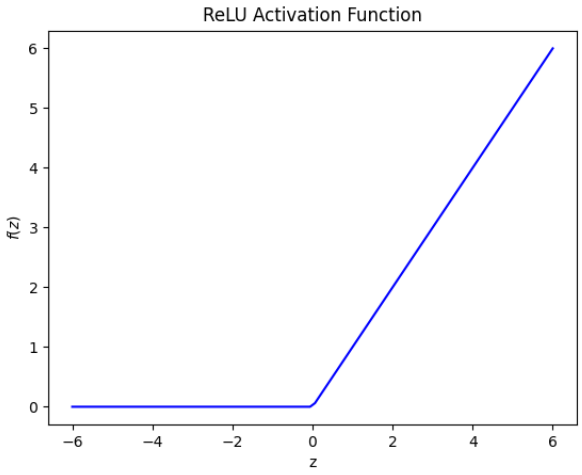


Figure 4.2: The ReLU function for the input $z \in$ [-6, 6] (Nielsen, 2015).

The Hyperbolic Tangent $Tanh$ function introduces non-linearity by compressing real-valued inputs into a continuous range between -1 and 1 (Goodfellow et al., 2016). Because the outputs mean naturally falls around zero, the network avoids systematically pushing weight updates in a single, biased direction during backpropagation. This inherent symmetry prevents gradients from continuously increasing, ensuring balanced weight adjustments that lead to a significantly faster and more stable training convergence. The mathematical expression for the function is presented in Equation 4.6.

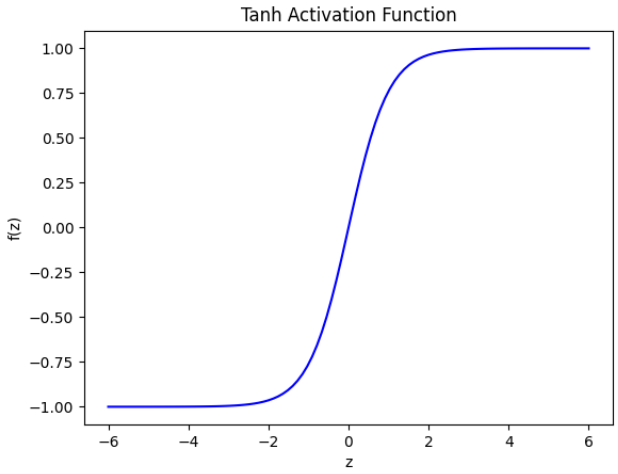


Figure 4.3: The Tanh function for the input $z \in$ [-6, 6] (Nielsen, 2015).

$$\tanh(z) = \frac{\sinh(z)}{\cosh(z)} = \frac{e^z - e^{-z}}{e^z + e^{-z}} = \frac{e^{2z} - 1}{e^{2z} + 1} \tag{4.6}$$

To solve advanced regression tasks, these individual building blocks must be systematically interconnected (Nielsen, 2015). By organizing multiple artificial neurons into structured, sequential layers they collectively form an Artificial Neural Network (ANN). The specific arrangement, depth, and connectivity of these aggregated layers define the network's architecture, the principles of which are detailed in the following section.

4.4 Network Architecture, Scaling and Normalization

Grouped and interconnected artificial neurons make up what's called sequential layers as stated previously. The layers can be categorized into three main groups of layers to represent a feed forward neural network (Nielsen, 2015). The initial part of the network is called *input layer* which are defined by the input parameters, the depth of this layer depends on the amount of features used. The main body of the architecture are the *hidden layers*, these layers are equipped with the selected activation function. These layers can have any arbitrary depth n and amount m selected. The last section is the *output layer* which is defined by the weighted output from the last hidden layer. A schematic of the general architecture of a feed forward neural network is shown in Figure 4.4.

Within this structured architecture, data flows in a single direction from the input layer, through the hidden layers, to the output layer in a process known as $forward$ *propagation* (Nielsen, 2015). Rather than calculating the output of each artificial neuron individually as $z = \sum w_j x_j + b$, the computations for an entire layer are

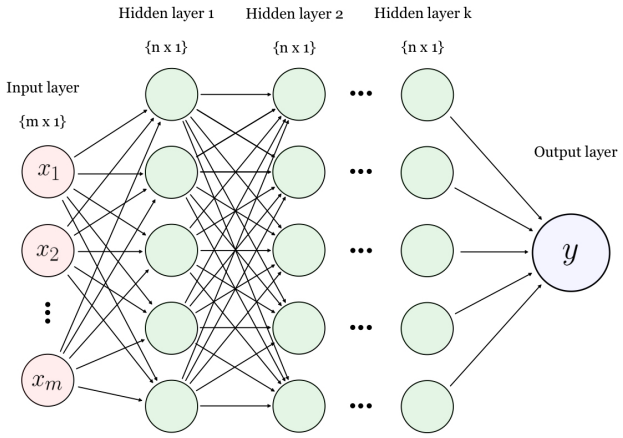


Figure 4.4: Schematic of a ANN architecture (Nielsen, 2015).

performed simultaneously using linear algebra. For a full layer, this feed-forward operation is expressed as a matrix multiplication.

$$\mathbf{Z} = \mathbf{X}\mathbf{W} + \mathbf{b} \tag{4.7}$$

In Equation 4.7, $\mathbf{X}$ represents the input matrix (either the raw features from the input layer or the activated outputs from the previous hidden layer), $\mathbf{W}$ is the weight matrix containing the parameters for all neurons in the current layer, and $\mathbf{b}$ is the corresponding bias vector (Nielsen, 2015). Passing the resulting intermediate matrix $\mathbf{Z}$ through the layer's activation function yields the final output that is fed forward into the following layer.

For the linear combination z to be predicted properly it has to be scaled to fit the boundaries of the activation functions shown in Figures 4.2 and 4.3. The scaling process is one of the most important parts of the network design as it have to be done right for the forward propagation to output properly scaled values (Nielsen, 2015). The transformation can be done in several ways but the most common scaling for regression problems are $Min-Max\ scaling$ or $Standardized\ Scaling$ schemes. The selected scheme depends on the features inherent properties, e.g. are all values could be inherently positive, and can be different schemes for an input–output pair. The two most reoccurring schemes mentioned are mathematically listed in Equations 4.8 and 4.9.

$$\overline{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.8}$$

Variables:

- $\overline{x}$ is the normalized, scaled value.
- x is the original value.

- $\min(x)$ is the minimum value in the feature column.
- $\max(x)$ is the maximum value in the feature column.

$$\overline{x} = \frac{x - \mu}{\sigma} \tag{4.9}$$

Variables:

- $\overline{x}$ is the standardized value.
- x is the original value.
- μ is the mean of the feature column.
- σ is the standard deviation of the feature column.

The added scaling function for the input and output data can be seen as additional layers to the network architecture (Nielsen, 2015). The scaling of the input layer is a transformation or normalization layer for the input features. At the output layer end, an additional inverse transformation layer is added to re-scale the output back to physical size. This can be seen in the evolved architecture schematic of the Artificial Neural Network in figure 4.5.

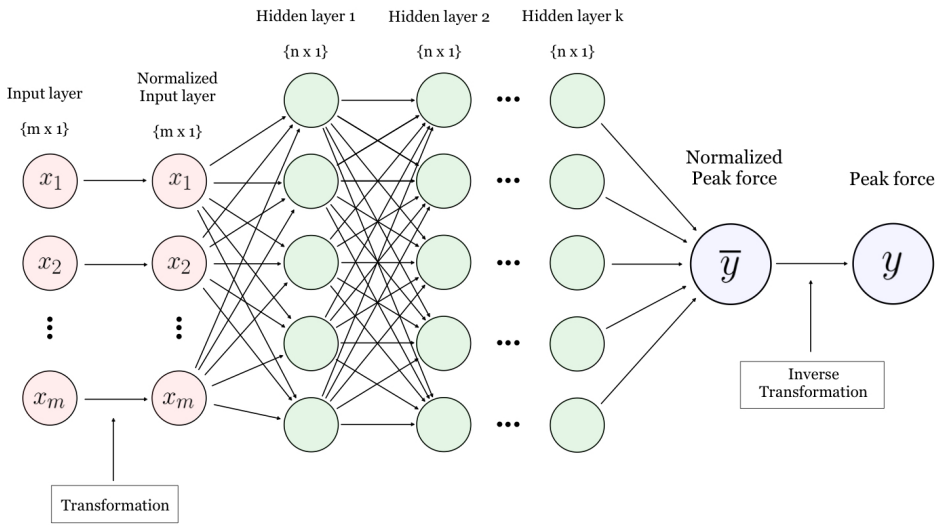


Figure 4.5: Schematic of a ANN architecture with transformation layers (Nielsen, 2015).

Once the design of the artificial neural network is complete and the input data has been properly scaled, the model is ready to be introduced to the training process (Nielsen, 2015).

4.5 The Loss Function

The training process involves defining what is generally called a *loss function*, which measures the model error for each prediction (Goodfellow et al., 2016). Similarly to the performance metrics introduced in Section 4.2, which measures performance externally, the model requires an internal differentiable objective function to close the distance to the *target value* at each training run, also called *epochs*.

For regression the standard choice for loss function is the mean squared error (MSE) introduced in Section 4.2 in Equation 4.1. Because the error is squared the function will give large penalties if the prediction includes large deviations from the target value, this forces the model to minimize significant outliers in the prediction (Goodfellow et al., 2016).

The training process should also work to mitigate overfitting to the training data as introduced in Section 4.2. The risk of overfitting can be reduced by introducing additional terms to the existing loss function (Goodfellow et al., 2016). These terms, so called *regularization* terms, are used to add additional penalties depending on the adjusted weights. The most common are L_1 and L_2 regularization.

L_1 *Regularization* (*Lasso*) adds a penalty term equivalent to the absolute value of the magnitude of the network's weights (Goodfellow et al., 2016). A unique property of L_1 regularization is that it can reduce the weights of less important features to exactly zero, effectively performing automatic feature selection and creating a sparse model. The penalized loss function is defined as:

$$Loss = MSE + \lambda \sum_{j=1}^{n} |w_j| \tag{4.10}$$

L_2 *Regularization* (*Ridge*) adds a penalty term equivalent to the square of the magnitude of the weights (Goodfellow et al., 2016). L_2 regularization heavily penalizes unusually large weights. This forces the network to distribute its weight values more evenly across all inputs, resulting in a smoother, more robust model that is less sensitive to variations in the training data. The penalized loss function is defined as:

$$Loss = MSE + \lambda \sum_{j=1}^{n} w_j^2 \tag{4.11}$$

In Equations 4.10 and 4.11, MSE represents the original un-regularized objective function (such as the Mean Squared Error), w_j represents the individual weights, and λ is a user-defined hyperparameter that dictates the severity of the applied penalty.

4.6 Backpropagation

Backpropagation is the fundamental learning algorithm used to train artificial neural networks (Nielsen, 2015). The purpose of the algorithm is to determine how the weights and biases in the network should be adjusted in order to reduce the prediction error of the model. The input is passed forward through the network by forward propagation introduced in Equation 4.7 and the activation function. The prediction is then evaluated using the lost function measuring the error. The ability of neural networks to learn through backpropagation is because the activation function are smooth and differentiable like the $ReLU$ and $Tanh$ functions as the example in Equation 4.12.

$$a^{(k)} = f_{ReLU}\left(W^{(k)}a^{(k-1)} + b^{(k)}\right) \tag{4.12}$$

The foundation of the backpropagation is the chain rule from multi-variable calculus (Nielsen, 2015). It provides a method to obtain the derivative of a function which is made up of several nested functions. In a neural network, each layer depends on the output of the previous layer. This implies that the final prediction is a composition of many functions. Applying the chain rule by taking the derivative of the loss function with respect to the layer weights, the expression would end up as in Equation 4.13.

$$\frac{\partial L}{\partial w^{(1)}} = \frac{\partial L}{\partial a^{(k)}}\frac{\partial a^{(k)}}{\partial a^{(k-1)}}\cdots\frac{\partial a^{(2)}}{\partial a^{(1)}}\frac{\partial a^{(1)}}{\partial w^{(1)}} \tag{4.13}$$

The result of the chain rule are the gradients which describes the sensitivity of the loss function with respect to the weights and biases of the neural network (Nielsen, 2015). The chain rule w.r.t all the weights in the network becomes the gradient vector denoted as $\nabla L(w_k)$. Together with the learning rate which is a hyperparameter denoted as η, that decides the step length of the gradient update, the gradient computes the change of the weights. The gradient represents the direction and steepness in the gradient space with respect to the absolute minimum error. The expression of the weight update also known as *gradient decent* is shown in Equation 4.14.

$$w_{k+1} = w_k - \eta\nabla L(w_k) \tag{4.14}$$

The gradients point in the direction of the steepest increase in the loss landscape (Goodfellow et al., 2016). This implies that the negative gradient points toward decreasing loss values and is therefore used to update the weights in the local minimum direction. As optimization progresses, the magnitude of the gradients typically decreases, resulting in progressively smaller weight updates. This is illustrated in Figure 4.6, showing the optimization trajectory in the weight space.

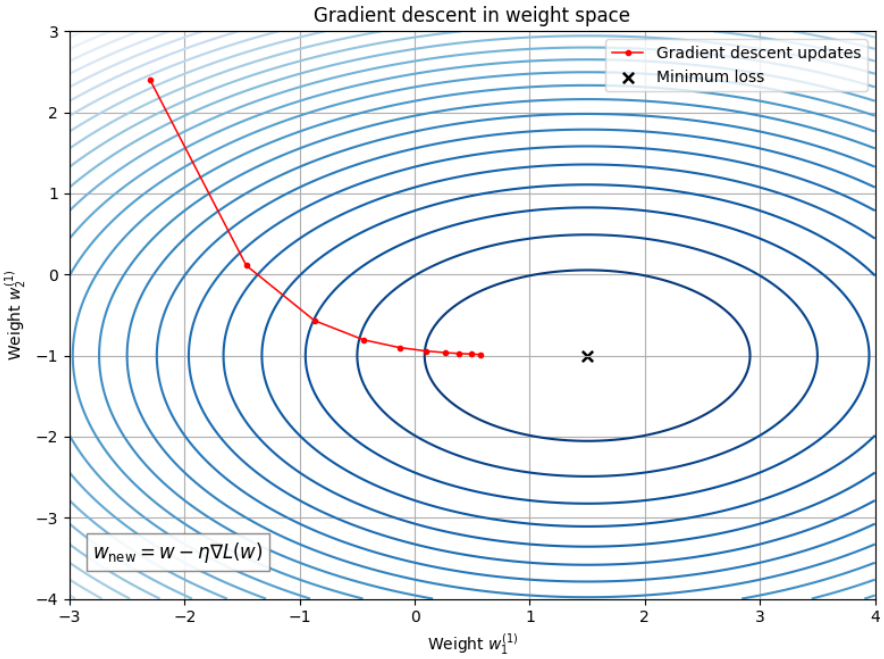


Figure 4.6: The optimization trajectory in the weight space.

In practice, the computation and application of these parameter updates are handled by optimization algorithms, which extend the basic gradient descent method to improve convergence speed, numerical stability, and training efficiency (Goodfellow et al., 2016). Two of the most widely used algorithms are the Stochastic Gradient Decent (SGD) and the Adaptive Moment Estimation (ADAM). These algorithms makes it possible to train the neural network in batches which further improve the computational efficiency. The ADAM optimizer is an improvement to the SGD, using momentum variables to more efficiently converge to a minimum.

4.7 Neural Ordinary Differential Equations

A Neural Ordinary Differential Equation (NODE) is a machine learning framework with roots in differential equation solvers (Worsham & Kalita, 2025). The most foundational numerical solver to update a state of variables h with respect to some trajectory t is the Forward Euler scheme. The problem is defined given an initial trajectory t_0 and an initial state $h(t_0)$, and from this map to the state of the final trajectory t_f, $h(t_f)$. This is called the *initial value problem* (IVP). The required information is the evolution of the state over the trajectory. The Forward Euler method defines this information as the update of the current state $h(t)$ by its derivative with respect to the trajectory $\dot{h}(t)$ for an arbitrary step size Δt. The full derivation of the IVP is layed out in Derivation 4.7.1.

Derivation 4.7.1: Initial Value Problem (IVP) of the neural ODE

In the Forward Euler scheme, the state update from the initial state to the next one is defined as

$$h(t + \Delta t) = h(t) + \dot{h}(t)\,\Delta t.$$

This can be interpreted as approximating the integral over one time step Δt and adding it to the initial value

$$h(t + \Delta t) = h(t) + \int_t^{t+\Delta t} \dot{h}(t)\,dt.$$

For very a very small time step the derivative can be assumed approximately constant, letting $\Delta t \to 0$ to arrive at the definition

$$\lim_{\Delta t \to 0} \frac{h(t + \Delta t) - h(t_0)}{\Delta t} = \dot{h}(t).$$

For a neural ODE the derivative $\dot{h}(t)$ is approximated by a neural network defined as the function $f_\theta(t, h(t))$, where θ are trainable parameters. The IVP can then be written as

$$\frac{dh(t)}{dt} = f_\theta(t, h(t)), \qquad h(t_0) = h_0.$$

Integrating the resulting vector field from the initial trajectory t_0 to the final trajectory t_f gives

$$h(t_f) = h(t_0) + \int_{t_0}^{t_f} f_\theta(t, h(t))\,dt.$$

For the integration scheme a numerical solver have to be chosen for this task. A good example is the 4th order Rugne-Kutta (RK4) scheme.

The NODE utilizes a neural network to map the initial state to the final state through a continuous function $f_\theta(t, h(t))$ instead of the derivative of each discretized step $\dot{h}(t)$. The IVP mapping this evolution of the state can be formulated as the initial value added to the integration over the full trajectory, the expression for the IVP is presented in Equation 4.15.

$$h(t_f) = h(t_0) + \int_{t_0}^{t} f_\theta(t, h(t))\,dt \tag{4.15}$$

The computation of the loss function for a NODE model is done by defining the loss for each state of the trajectory (Worsham & Kalita, 2025). The loss function can also be defined as just comparing the terminal state of the problem, which is done by only using the final predicted value as the condition for learning. An additional $\mathcal{L}_2$ regularization (ridge) term is also beneficial to NODE models to penalize large weights mitigating larger prediction values. In this thesis the full trajectory is used for the loss which is more computationally heavy, but more often leads to more precise predictions. The expression of the loss function is given in Equation 4.16.

$$\mathcal{L}_{MSE}(t, h(t)) = MSE\left(\int_{t_0}^{t} f_\theta(t, h(t))dt,\ \ h(t)_{true}\right) \tag{4.16}$$

The definition of the IVP in Equation 4.15 converts a discretized ODE bound to some step size into a continuous ODE (Worsham & Kalita, 2025). To update the vector field into the actual state evolution it has to be integrated numerically. This

can essentially be done using any numerical solver presuming that the gradient field has been predicted correctly. The studied model employs the 4th order Rugne-Kutta solver which evaluates the derivative as the weighted average of four increment slopes over the interval. The full formulation of the solver is presented in Equations 4.17.

$$k_1 = f_{NN}\left(t, h(t)\right), \tag{4.17a}$$

$$k_2 = f_{NN}\left(t + \frac{\Delta t}{2}, h(t) + \frac{\Delta t}{2} k_1\right), \tag{4.17b}$$

$$k_3 = f_{NN}\left(t + \frac{\Delta t}{2}, h(t) + \frac{\Delta t}{2} k_2\right), \tag{4.17c}$$

$$k_4 = f_{NN}\left(t + \Delta t,\ h(t) + \Delta t\, k_3\right), \tag{4.17d}$$

$$h(t+1) = h(t) + \frac{\Delta t}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right). \tag{4.17e}$$

Using a model like the NODE makes it possible to make predictions on a full evolution of states while making the model generalizable (Worsham & Kalita, 2025). As the neural network itself, in the form of a MLP, only predicts a single derivative at the time instead of a full trajectory, the effects of overfitting is somewhat mitigated. As the NODE is essentially an ODE solver integrated with machine learning, it requires only a single trajectory as input. Because of this, the model can be viewed as a reduced order model (ROM) of the high fidelity problem it is applied on.

5 Machine Learning Models

In this chapter the different models used in the thesis will be presented. The Artificial Neural Network (ANN) will be presented first, as it is the underlying predictive driver for all the models. The network is built on the theory from Sections 4.3 - 4.4. The details from the architectures are described here and in what way they are combined to produce other models. The Neural ODE (NODE) solver for example uses the ANN to predict the derivatives for the solver to integrate.

5.1 Design of Multi-layered Perceptron Predicting the Out-of-Plane Peak Force Capacity

The objective of the Multi-layered Perceptron (MLP) network model is to predict the OOP peak force capacity of the URM wall, denoted as Q_{peak}, from a set of geometrical, boundary and material input features. The model where developed as a data-driven surrogate for the nonlinear FE simulations. The relationship between the wall properties and the OOP capacity is learned directly from the generated simulation dataset (Goodfellow et al., 2016). In this work, the input vector $\mathbf{x}$ contains parameters describing the wall geometry, material properties, boundary and loading conditions, while the output is the predicted peak force capacity Q_{peak}.

The input parameters $\mathbf{x}$ were selected from the parameter space defined in Section 3.4 based on their importance and correlation with the peak force. In addition, a trial-and-error approach was applied during the selection of the input parameters. The created subset of parameters are engineered composite features made to represent the masonry in full. These features are listed below as the input vector for the MLP.

$$\mathbf{x} = [\lambda, K_s, F_0, E_m, f_k]^T \tag{5.1}$$

Selected and Engineered features

- λ is the relative wall thickness: $\frac{t}{H}$
- K_s is the boundary spring stiffness
- F_0 is the pre–compressive force
- E_m is the composite Young's modulus of masonry
- f_k is the characteristic compressive strength of masonry

Here the characteristic compressive strength is given by the equation defined in Eurocode 6 (EC6) (European Committee for Standardization, 2006), also given in Equation 2.1. The composite modulus of elasticity is given as a the expression of springs in series with weighing according to the corresponding volumes of each

material in the specific URM wall construction. The full expression of Young's modulus for masonry is given by Equation 5.2.

$$E_m = \frac{1}{\frac{V_{brick}}{E_{brick}} + \frac{V_{mortar}}{E_{mortar}}} \tag{5.2}$$

Variables

- E_{brick} is Young's modulus for brick
- E_{mortar} is Young's modulus for mortar
- V_{brick} is the volume of brick in the wall
- V_{mortar} is the volume of mortar in the wall

The selected subset of parameters is then evaluated on their correlation to one another using a Spearman correlation analysis presented in section 3.4.6 (Spearman, 1904). The analysis is used as a feature-selection tool together with the manual selection of variables with known mechanical relevance to the structural response of the masonry wall from Section 3.4.6. This ensures that the input matrix includes both statistically important variables and parameters known to influence the structural behavior of the wall. The Spearman correlation matrix is highlighted in Figure 5.1.

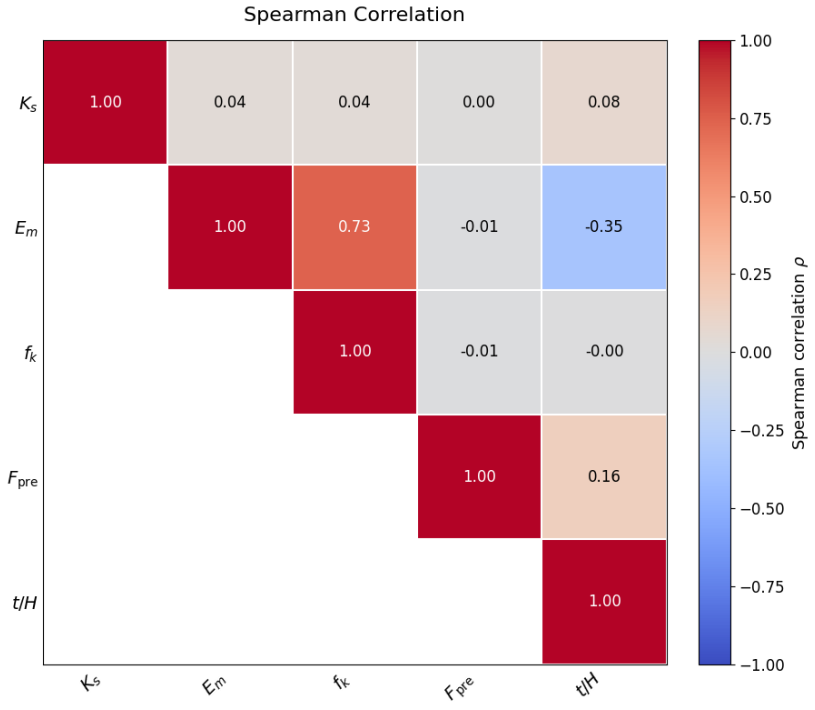


Figure 5.1: The Spearman correlation matrix for the subset of parameters chosen from the parameter space.

Before training the MLP, the input and output variables were scaled to contribute more evenly during optimization (Goodfellow et al., 2016). The input matrix $\mathbf{x}$, containing the selected input features, was standardized using the standard scaling

procedure in Equation 4.9. The target variable was scaled separately using the min–max scaling defined in Equation 4.8.

The MLP was implemented as a feed-forward multilayer perceptron with five input features and one scalar output corresponding to the predicted OOP peak force capacity (Nielsen, 2015). The network consists of two hidden layers, each containing 16 neurons. A rectified linear unit (ReLU) activation function was applied after each hidden layer to introduce nonlinearity into the model, illustrated in Figure 4.2. The final layer is a linear output layer with one neuron, producing the value of Q_{peak}.

Table 5.1: Summary of the MLP architecture used for prediction of Q_{peak}.

Component	Description
Input layer	5 input features
Hidden layers	2
Neurons per hidden layer	16
Activation function	ReLU

The MLP was trained for 2500 epochs using Adaptive Moment Estimation (ADAM) as the optimization algorithm. The learning rate was set to 5×10^{-3}, which controls the magnitude of the parameter updates during training. The relatively large number of epochs was chosen to allow the network sufficient iterations to converge, while the validation performance was monitored to assess whether the model generalized to unseen data.

Table 5.2: Summary of the MLP training setup.

Training parameter	Value
Optimizer	ADAM
Learning rate	5×10^{-3}
Epochs	2500
Loss function	MSE
Target variable	Q_{peak}

To evaluate how the selected input parameters influenced the prediction of Q_{peak}, a SHAP analysis was performed on the trained MLP model. The SHAP values quantify the contribution of each input feature to the model prediction and therefore provide an interpretation of how the network considers the geometrical and material parameters (Hamilton & Papadopoulos, 2024). The results are presented using a beeswarm plot, where each point represents one simulation case and the horizontal position indicates whether the feature contribution increases or decreases the predicted peak force and the color of each point represents the corresponding feature value. This analysis was used to assess whether the trained model had learned physically reasonable dependencies between the input parameters and the OOP peak force capacity.

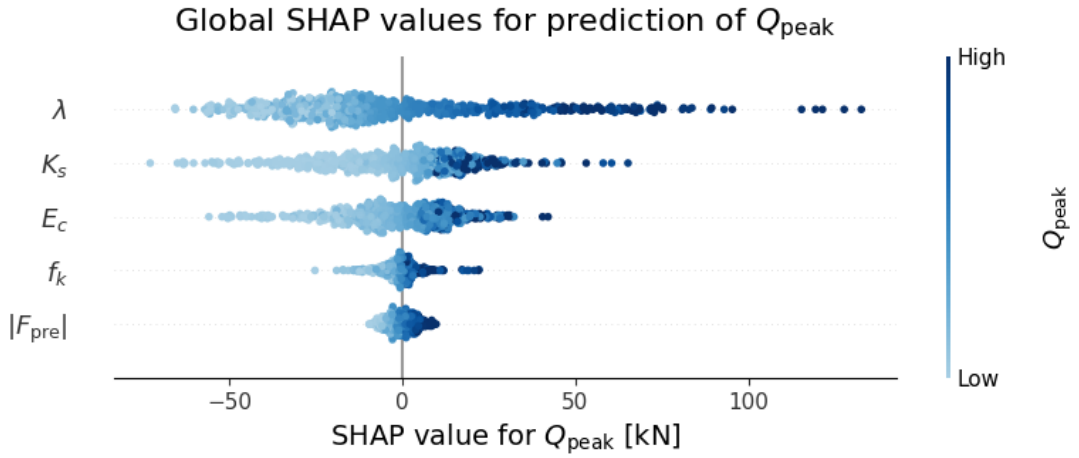


Figure 5.2: SHAP analysis for parameter evaluation toward target value.

The SHAP analysis provides a physically interpretable indication of how the selected input parameters influence the predicted peak force capacity. The largest contribution is associated with the relative wall thickness, λ. An increase in λ corresponds to a stockier wall geometry, which increases the OOP force capacity. The spring stiffness K_s also shows a clear positive influence on the prediction. Higher spring stiffness allows a larger increase in vertical thrust during OOP deformation. The characteristic compressive strength of masonry gives an increase in the predicted response for higher values. E_{m} has a weaker but still visible influence and F_0 shows the smallest global contribution in this analysis which indicates that the model relies more strongly on geometry, boundary stiffness, and masonry strength when predicting Q_{peak} within the sampled parametric range.

5.1.1 Results of the MLP

The final Multi-layered Perceptron model is evaluated purely on static data, mapping the geometric, material and boundary input parameters to the scalar output of the peak force capacity. The training regime for this network measured on Mean Square Error Loss and trained for a duration of 2500 Epochs. The resulting Loss vs Epochs evolution is presented in Figure 5.3. The total training time had a reoccurring duration of approximately 29.5 seconds.

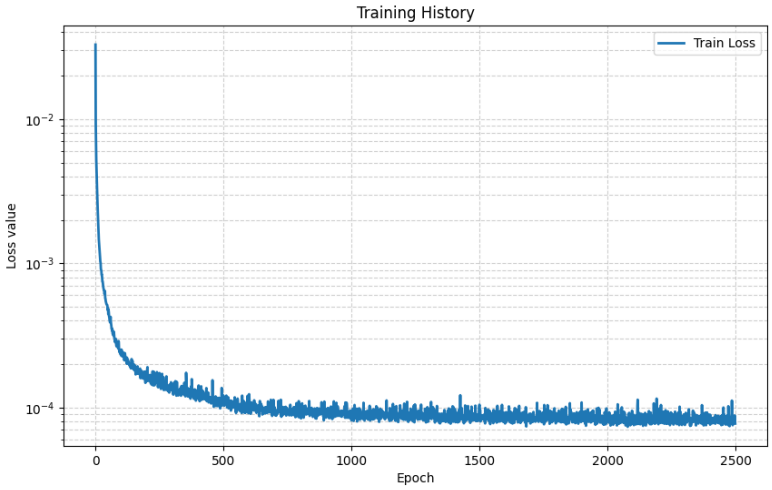


Figure 5.3: Evolution of training loss for the MLP model.

For the specific training regime the MLP model is training to fit the training data which is represented by the loss in Figure 5.3. The model is then evaluated on the the same training data and the withheld 20% of the data used for validation. The parity between the true target values Q_{peak}^{Abaqus} and the predicted target values Q_{peak}^{NN} is presented in the parity evaluation in Figure 5.4.

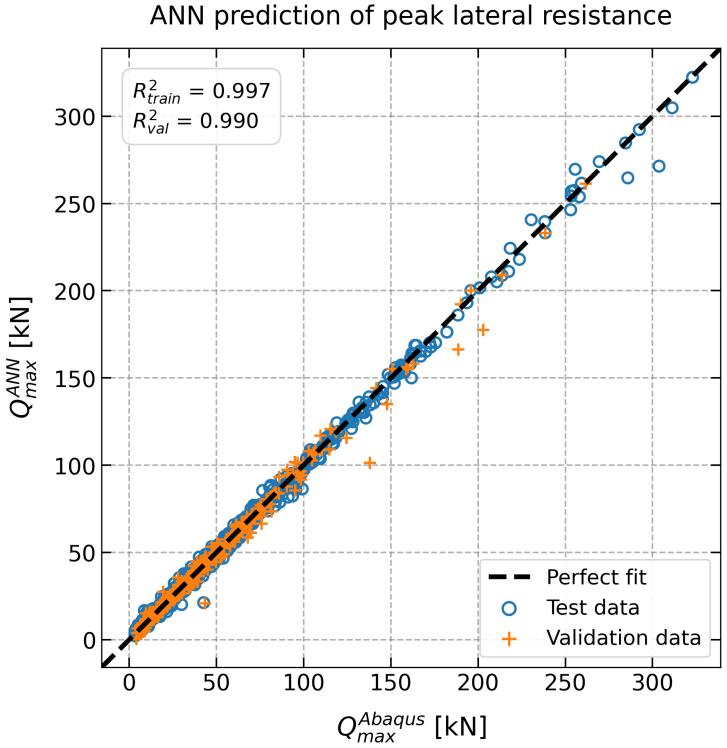


Figure 5.4: Parity plot of validation and training for the MLP model.

The results of the predictions of the Abaqus data are also measured using the stan-

dard performance metrics presented in Section 4.2. The calculation of these metrics is performed on all training and validation predictions and is summarized in Table 5.3.

Table 5.3: Summary of prediction performance for the MLP model.

Metric	Training	Validation
MSE	7.09	15.90
RMSE	2.66	3.98
MAE	–	–
R^2	0.996	0.989

To further test the model on unseen data, the results from the RISE physical experimental tests where used to evaluate the model on all the failure modes. The evaluation will test the models ability to extrapolate to unseen structural behavior. The results are presented in Figure 5.5.

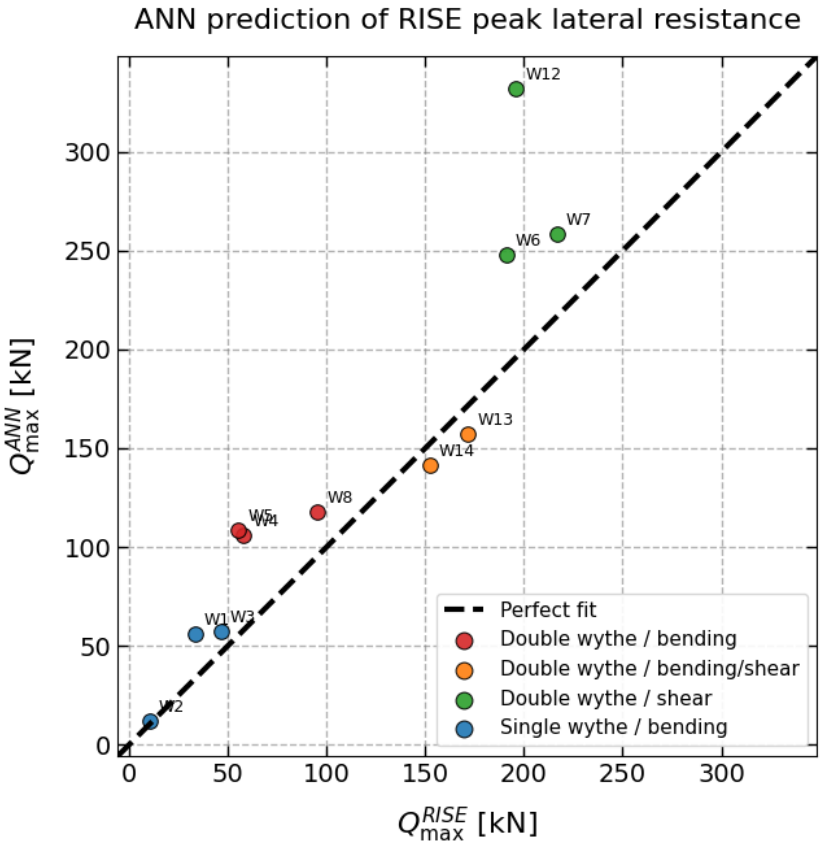


Figure 5.5: Evaluation of the MLP against the experimental tests.

Studying the result parity plot it is evident that the model does predict the single-wythe bending failure specimens well. The extrapolation to the double-wythe models have the best results for the bending/shear failure specimens with less accurate

predictions for the pure bending double-wythe specimens. The specimens subjected to shear failure is harder for the model to predict which is presumed. The summary of performance metrics is for the RISE test data is shown in Table 5.4.

Table 5.4: Summary of prediction performance for the MLP model on RISE data.

Metric	Value
MSE	922.52
RMSE	30.37
MAE	22.84
R^2	0.82

Overall, the MLP model shows strong predictive performance on the simulated Abaqus data and is able to recover the peak force capacity with high accuracy. The evaluation on the RISE experiments indicates that the model also captures parts of the physical test response, especially for bending-dominated specimens (W1, W2, W3, W4, W5, and W8). However, the reduced accuracy for shear-dominated cases highlights that the model is mainly valid within the structural behavior represented in the training data.

5.1.2 Discussion of the MLP prediction results

By observing the predictions made by the MLP on the Abaqus data it is clear that the model can predict the peak force capacity well. It is also observable that predictions on cases where the resulting peak force is on the lower end of the spectrum are predicted with higher accuracy. The density of data points is also greater in this region, which explains the increase in error for predictions of higher peak force because of the increased scarcity of the data.

The observed prediction on the physical test data from the RISE four point bending tests can be discussed in two parts. Firstly, the prediction on the specimens with features that fall within the space of the dataset, For walls W2 and W3, are predicted with high accuracy which is expected. Wall W1 that also is within the dataset is the outlier as the validation seen in Section 3.3 shows that ABAQUS could not fully capture the behavior of the experimental wall. Since the machine learning model is trained on the numerically generated dataset, any discrepancies or limitations present in the ABAQUS simulations are inherently transferred to the ML predictions. Secondly, with respect to the extrapolation aspect, the predictions are more scattered. The double-wythe specimens subjected to pure bending failures is over predicted by the model which indicates difficulties for the model to fully extrapolate. For the bending/shear cases the prediction is better, but still under-predicted, which could be explained by the combination failure of the walls. If there are shear cracks present in the wall when bending failure occurs, the overall capacity of the wall could be lower at the time of failure. For the specimens failing in shear, the capacity is over-predicting showing that the model is incapable of capturing the early failure of the shear. It can be argued that the bending resistance of the wall

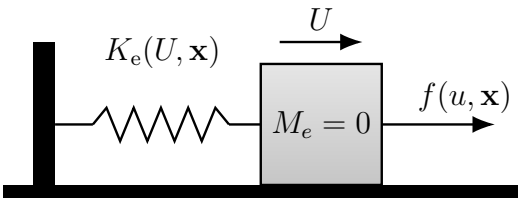
would be closer to the level of the predicted value if the internal components of the wall was stronger, leading to a bending failure.

5.2 Neural ODE model predicting the out-of-plane force capacity

The NODE model presented in this section is designed to capture the evolution of the OOP force capacity of an URM wall subjected to an OOP load. The model resides in only taking into account a mid-height displacement and static variables, mapping these to the total reaction force of the OOP by considering a reduced order model (ROM). The design of the overarching architecture of the structural system and the neural ODE model is presented in Subsection 5.2.1. Following this, the observed results of the training and predictions are presented in Subsection 5.2.2.

5.2.1 Design and Architecture of the Neural ODE Model

To capture the behavior of the URM wall in a computationally cost effective way, the NODE is interpreted as a ROM, from the logic of Subsection 2.2.3 reducing the high fidelity problem setup in ABAQUS to a single degree of freedom. The model maps the mid height displacement U_{mid}, together with the static variables $\mathbf{x}$ (presented in Equation 5.1), to the force OOP force capacity $Q(U)$. The coordinate point of reference is resisted by an equivalent stiffness $K_e(U, \mathbf{x})$, in the form of a non-linear spring. The generated force acts as the OOP resistance $f(U, \mathbf{x})$ of the URM wall which is equal to the force capacity. A schematic of the logic of the reduced model is presented in Figure 5.6.



$$\begin{cases} U_{\text{mid}} = U \\ K_{\text{e}}(U, \mathbf{x})\, U = f(U, \mathbf{x}) \\ f(U, \mathbf{x}) = Q(U) \end{cases}$$

Figure 5.6: Reduced order modeling (ROM) of force-displacement system at the mid-height of the wall strip.

The inputs to the NODE model are firstly the governing trajectory of the problem which in a quasi-static loading case is U. Secondly the state which to update over the trajectory is injected as the current force capacity $Q(U)$. Lastly, the static variables are used as a static encoding of the current URM wall. To make the training of the NODE feasible, the trajectories from ABAQUS are interpolated down to 25 data points per trajectory. The defined trajectory and state is listed below.

$$t = U \qquad h(t) = [Q(U)]$$

The NODE model is then designed to learn the evolution of the OOP resistance of the wall along the mid height displacement path. The state update function produced by the neural network is the vector field between the current state of the force capacity and the next state. The output is by that definition expressed as the derivative of $Q(U)$ along the displacement field U, giving the output shown in Equation 5.3.

$$\frac{dQ}{dU} = f_{NN}\left(U, Q, \mathbf{x}\right) \tag{5.3}$$

The mapping of the current state of the force capacity through the vector field is then performed by solving the initial value problem (IVP) in the same way as shown in Equation 4.15. Were the trajectory is the mid height displacement U and the state is the force capacity $Q(U)$. The statc input features $\mathbf{x}$ are included as a conditioning variable allowing the same NODE to represent several wall configurations within the sampled parameter space. The model specific IVP is expressed in Equation 5.4.

$$Q(U) = Q_0 + \int_{U_0}^{U} f_\theta(U, Q(U), \mathbf{x})\, dU \tag{5.4}$$

The loss function used for the predicted states along the displacement path and the true states is computed using the mean squared error (MSE). The error is checked for each state update and is formulated using the result from the IVP $Q(U)$ to the true state of the Abaqus curve $Q_{Abaqus}(U)$. An additional $\mathcal{L}_2$ regularization term is also applied to penalize large derivative predictions. As training is done in batches the error is also normalized with respect to the batch size. The complete expression of the loss function is presented in Equation 5.5.

$$\mathcal{L}_{MSE}(U, Q(U), \mathbf{x}) = \frac{1}{N}\, MSE\left(\int_{U_0}^{U} f_\theta(U, Q(U), \mathbf{x})dt,\;\; Q_{ABAQUS}(U)\right) \tag{5.5}$$

The derivative function f_θ is represented by a Multi-layered Perceptron (MLP). The MLP has 7 input features given as the current displacement, force capacity and the static features. The output feature is the derivative of the force capacity with respect to displacement $\frac{dQ}{dU}$. The MLP is equipped with 2 hidden layers and 32 neurons in each layer. The nonlinear activation function is the Tanh function presented in Subsection 4.3. The predicted derivative is then integrated numerically using the 4th order Rugne-Kutta (RK4) scheme presented in Equation 4.17. A summary of the model architecture is presented in Table 5.5.

Table 5.5: Summary of the NODE architecture and framework design.

Component	Description
State variable	Reaction force, Q
Integration variable	Mid-height displacement, U_{mid}
Conditioning variables	Static wall parameters, $\mathbf{x}$
Network output	Force derivative, $\mathrm{d}Q/\mathrm{d}U$
Solver	Fourth-order Runge–Kutta, RK4
Response range	Initial state to peak force

With the architecture and model design completed, the NODE can be put through training and and evaluation against the ABAQUS data.

5.2.2 Results of the Neural ODE on the Out-of-Plane Force Capacity

The NODE model is trained on the Abaqus dataset using 80% of the cases for training with the rest withheld for evaluation purposes. The resulting model is trained during 1800 epochs, giving a training time of 516 seconds. The optimizer used for gradient decent is the "Adam" optimizer resulting in the training shown in the loss against epochs plot in Figure 5.7.

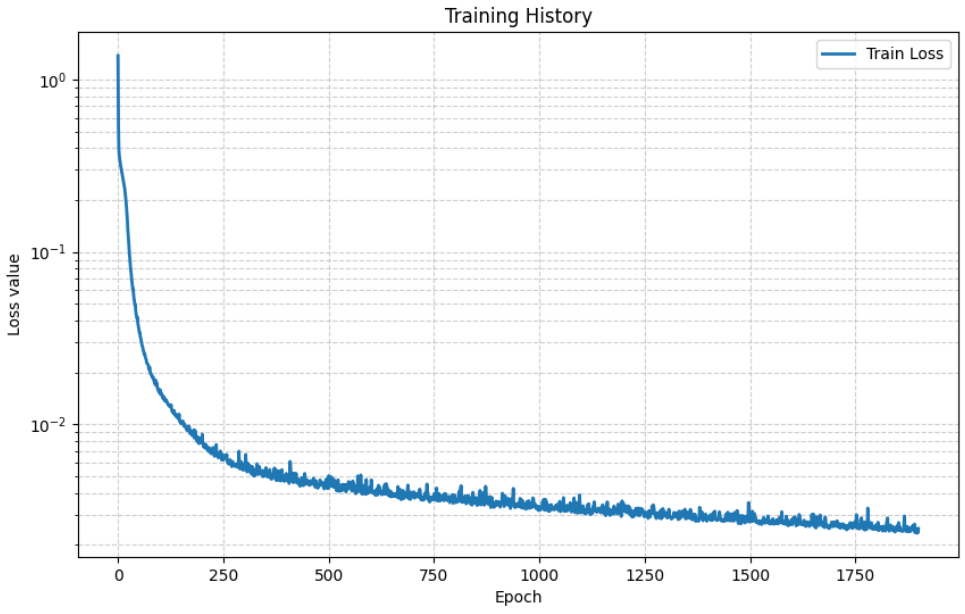


Figure 5.7: The Loss vs. Epochs during the training of the NODE.

The evaluation of the reduced order NODE model is done against three different cases in the evaluation data subset. These correspond to different boundary condi-

tions defined by the stiffness in the boundary spring K_s. The boundary conditions are: simply supported, semi-fixed and fully-fixed. The cases have been selected so that all other parameters are roughly similar, resulting in the spring stiffness being the parameter of influence. The predictions of the force-displacement fields for all three cases are shown in Figure 5.8 . The results from the NODE prediction are compared to the original un-interpolated force-displacement curves from ABAQUS.

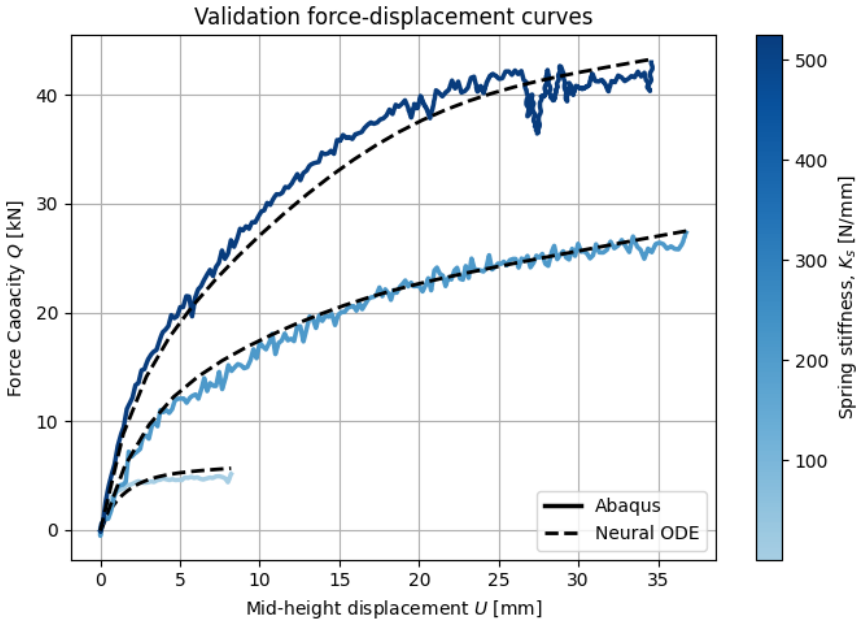


Figure 5.8: Predicted trajectory of the force-displacement trajectories in the validation subset for different values of the boundary spring K_s.

The peak force capacity Q_{peak}, across all cases is produced by evaluating the final states of the trajectories. The resulting parity plot is presented in Figure 5.9.

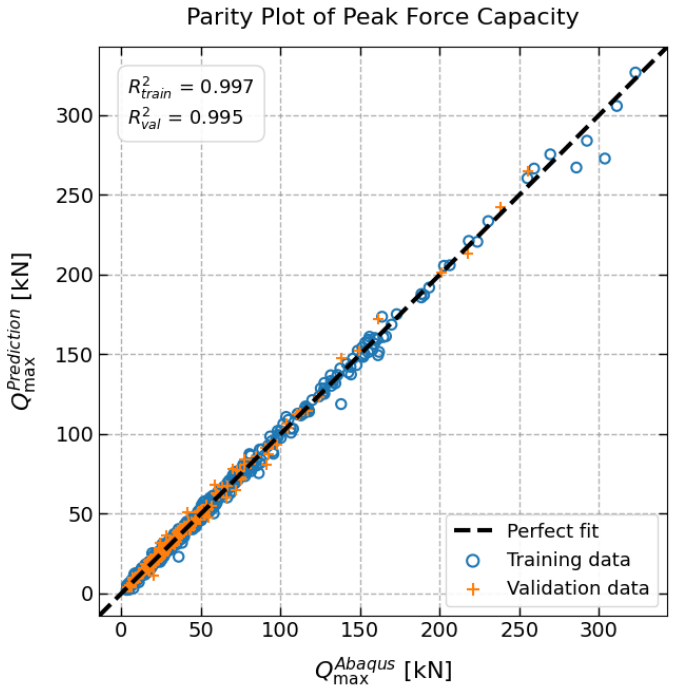


Figure 5.9: Parity plot of the peak force capacity Q_{peak} between Abaqus data and the predicted value for all training and validation cases.

Further evaluation includes real experimental data on the force-displacement fields from the four point bending tests at RISE. The first predictions are performed on specimens W1, W2 and W3, with material and geometrical parameters residing within the dataset. The predictions are presented in Figure 5.10.

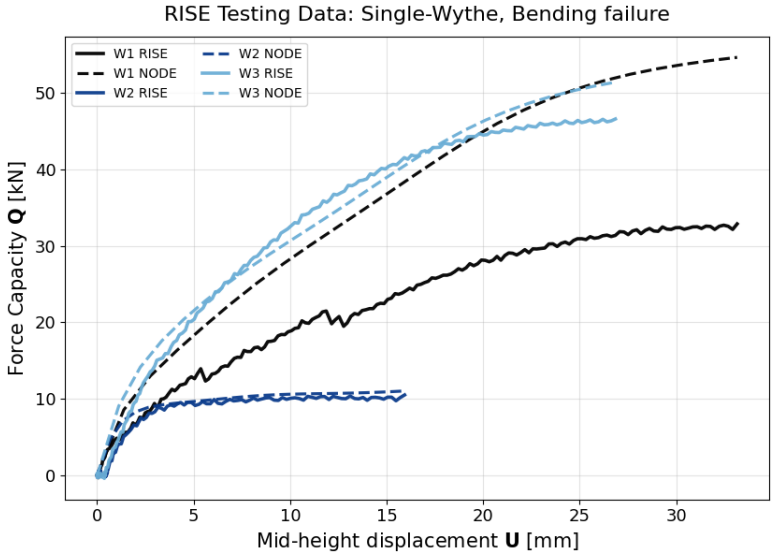


Figure 5.10: Force-Displacemend field prediction of the physical RISE experiment data for Specimens W1, W2 and W3.

Other specimens W4, W5 and W8 tested in the RISE experimental tests are of double wythe geometry, which puts them out of the scope of the dataset geometrically. Figure 5.11 illustrates the prediction for double-wythe walls subjected to bending failure.

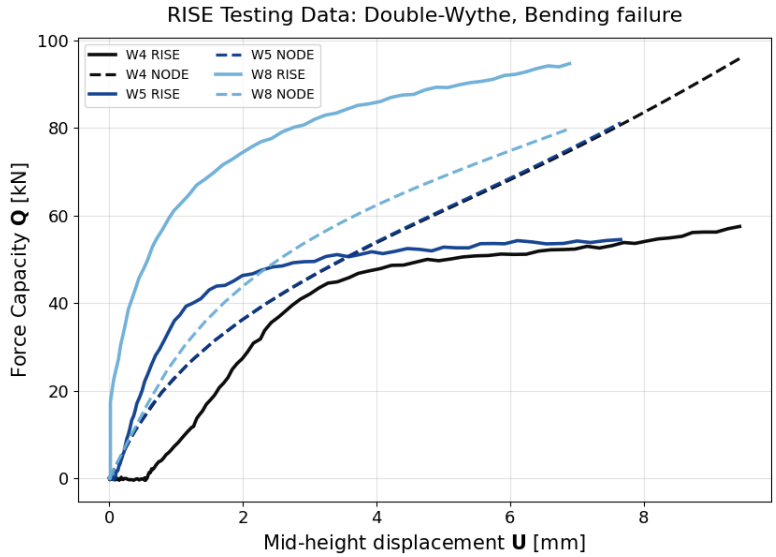


Figure 5.11: Force-Displacemend field prediction of the physical RISE experiment data for Specimens W4, W5 and W8. Double-wythe bending failure.

To further test the extrapolation predictions, double-wythe specimens W13 and W14 subjected to shear/bending failure modes are also predicted. The prediction of the force-displacement field is shown in figure 5.12.

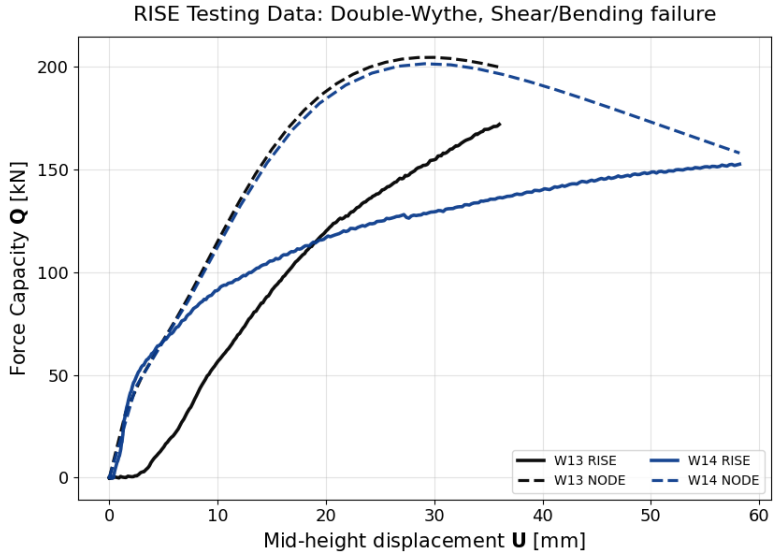


Figure 5.12: Force-Displacemend field prediction of the physical RISE experiment data for Specimens W13 and W14.

As a final prediction the specimens double-wythe W6, W7 and W12 subjected to shear failure is tested on the model. The force-displacement prediction is presented in Figure 5.13.

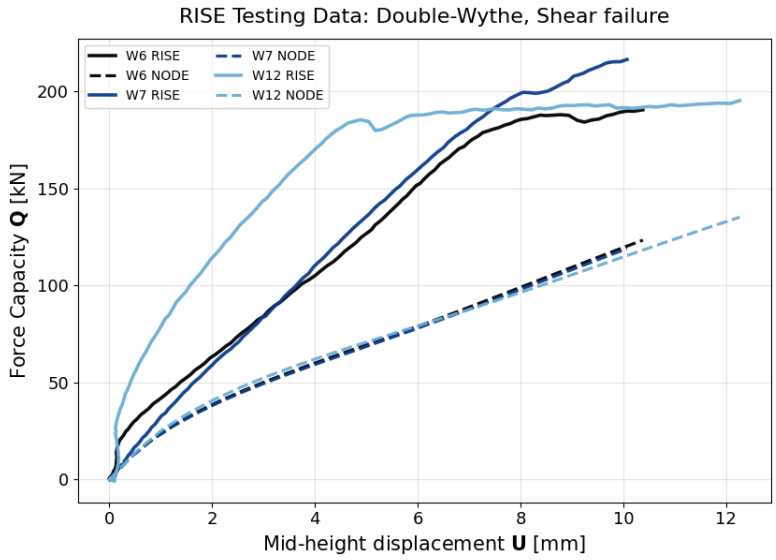


Figure 5.13: Force-Displacemend field prediction of the physical RISE experiment data for Specimens W6, W7 and W12.

As such the model have been tested on the full set of experimental data, where the performance was best on the data within the parametric range of the dataset. Further discussions will be brought up in the next section.

5.2.3 Discussion of the NODE prediction Results

Predictions from the reduced order NODE model can be discussed in terms of accuracy and extrapolation. The model predictions on the Abaqus dataset perform well with respect to accuracy. It can predict the evolution of the force capacity along the displacement path well even though the structural problem has been reduced to only capture the mid height behavior. The prediction capabilities performs well across all boundary conditions which in turn gives a well align parity when comparing the peak force capacity.

The predictions of the Abaqus data carries over well to the physical test data of some RISE specimens. For specimen W2 and W3, which parameters are within the data set space, the model performs very well. The models limitations are obvious when tested on specimens with parameters outside the space of the dataset. The extrapolation to other geometries and failure modes shows that it is incapable of predicting cases with these features as the prediction fail on both the shape and the peak force prediction.

5.3 Symbolic Regression

Symbolic Regression (SR) is a data-driven method for searching mathematical expressions that describe the relationship between input variables and an output variable (Kronberger et al., 2025). Most modern SR approaches use Genetic Programming (GP), an evolutionary computation technique inspired by natural selection. Candidate solutions are iteratively evolved through operations such as selection, crossover, mutation, and replacement. For the purpose of finding an analytical expression predicting the peak force capacity the model PySR presented by Cranmer, 2023 is adopted, it utilizes certain aspects beyond the typical GP to evaluate the analytical expression based on complexity and adaptation of the expression.

5.3.1 Design and utilization of the PySR model

The PySR model utilizes feature selection based on a gradient boosting tree algorithm, fitting the input to the output with a simple ML-model for selecting most important features, and imported into the main search loop. The natural selection of the GP is based on a tournament like selection of equations based on fitness (loss). The best performing equations are ranked and used to generate new candidate expressions through mutation and crossover operations. Over time, weaker equations are replaced by improved expressions. The fitness of an equation is based on a prediction loss and its complexity, this fitness factor is explained by Equation 5.7, where l_{pred} is a user defined performance metric on the predictive accuracy.

$$l(E) = l_{pred}(E) \cdot exp(frecency[C(E)]) \tag{5.6}$$

The second term involves the *"frecency"* of C(E), which combines the frequency and recency of expressions of a given complexity (Cranmer, 2023). Expressions with complexity values that occur frequently and recently are penalized more strongly, discouraging the symbolic regression search from over-fixating on a single complexity regime or local minimum. As a result, the complexity penalty is adaptively updated during the search process, encouraging search across different levels of model complexity. The user defined *"parsimony"* parameter controls the overall strength (or scaling) of this penalization rather than prescribing a fixed complexity penalty. PySR utilizes something denoted as populations which can be compared to as islands. Islands evolve expressions independently from each other with each of there own tournaments to create expressions. Occasionally through the course of SR the best expressions found on each of the islands gets shared between the islands. These shared expressions may undergo mutation and crossover enabling useful building blocks to be utilized in different islands. Islands prevents early convergence in local minima and improves the efficiency of the process.

As previously stated, the predictive loss function, as well as the binary and unary operators available to the SR, are user defined. Certain operators may be more suitable depending on the field of study, as some mathematical relationships are more commonly encountered within certain fields of study. For the present model, only

basic operators were considered to avoid unnecessary complexity.

A summary of the selected user-defined parameters is provided in Table 5.6. The model selection criterion was set to *best*, corresponding to the expression that provides the most suitable trade-off between predictive accuracy and model complexity. This choice was made to prioritize equations that remain practical and relatively easy to interpret. The *max size* parameter defines the maximum allowed complexity of an expression by limiting the number of nodes in the symbolic tree. Furthermore, the loss function was selected as MSE, as it penalizes large errors heavily.

Table 5.6: Summary of user defined parameters in PySR.

Parameter	Value
Iterations	1000
parsimony	1×10^{-3}
model selection	best
Binary operators	$+$,-,$\cdot$,/,$\widehat{}$
Unary operators	square , sqrt
Max size	15
loss	MSE

The input variables consisted of all varied material parameters together with the engineered features developed in Section 5.1. The output target was Q_{peak}, representing the most straightforward approach for SR to operate on directly. However, early experimentation, inspired by the analytical formulations presented in Section 2.2, indicated that the thrust force could serve as an informative parametric quantity, as several existing analytical expressions incorporate thrust as a key variable.

A preliminary test was therefore conducted in which the resulting thrust force was included as an input to the PySR framework. This yielded accurate predictions, as illustrated in Figure 5.14.

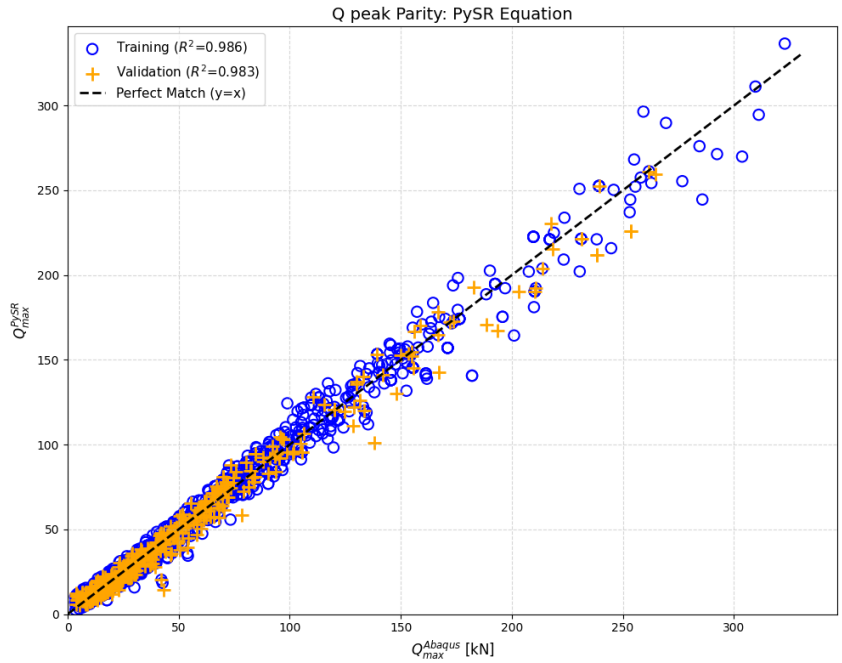


Figure 5.14: Q_{peak} parity plot analytically searched with the resulting *Thrust*.

The discovered equation corresponding to the parity plot is presented in Equation 5.7. Notably, the expression depends on only one material parameter, while the remaining terms are global or geometrical parameters. However, since thrust is used here as an input, while also being similarly difficult to predict as the force capacity, a symbolic expression for thrust was first derived using PySR and subsequently used as an input for constructing an expression to estimate the peak force capacity,which is presented in Subsection 5.3.2.

$$Q_{peak} = 3.2\frac{t}{H}(\sqrt{E_m} - F_0 + T) \tag{5.7}$$

A final consideration regarding SR is that, in its standard form, it does not account for the physical units of the variables. From the perspective of the model, all variables are treated as numerical values, with no regard for dimensions. This, in turn, can result in expressions that may not remain valid during unit conversions. Research has been conducted on SR models to introduce consistent units (Udrescu & Tegmark, 2020), where such implementations have eased the process of finding already known expressions. As the current procedure is not intended to rediscover an already developed equation, a search field as wide as possible has been chosen to serve the purpose of developing this analytical expression. Therefore, the metric prefixes introduced for the numerical model will be the valid ones, and unit conversion is not applicable.

5.3.2 Results of the Symbolic Regression

The results is in two pieces, one where an equation for the Q_{peak} is directly searched, presented in Appendix G. The second gathers an expression for *thrust* with the help of PySR and implements the equation as an input to predict the force capacity. The input variables are defined in SI units, consistent with the model formulation described in Section 3.2. The model output is reported in kN regardless of internal unit conversions.

For a search for an expression of the *thrust* at the moment of the maximum peak force capacity, results in a parity with strong correlation between the simulated thrust and the found equation which can be observed in Figure 5.15 below.

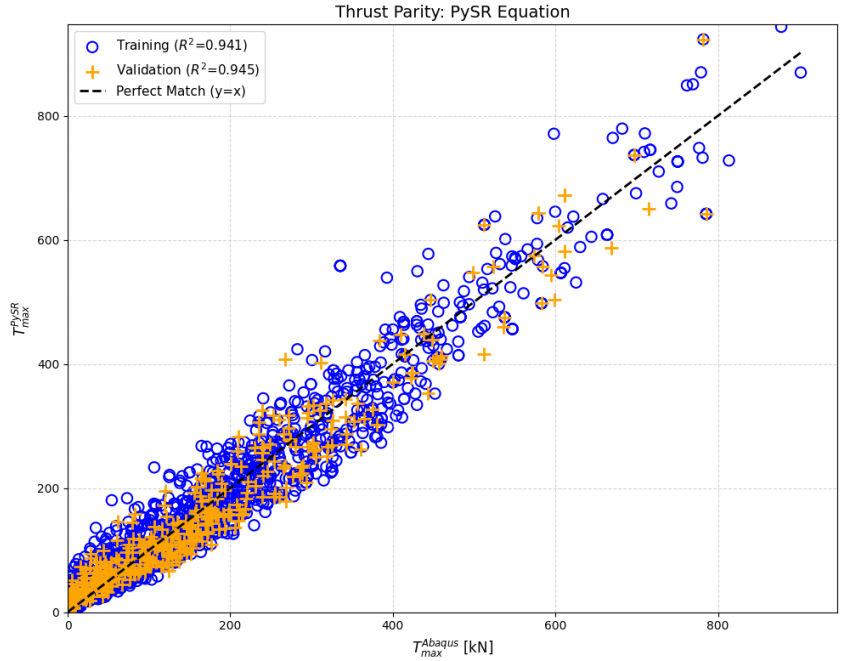


Figure 5.15: Thrust equation found using PySR compared with ABAQUS results.

The proposed equation is presented in Equation 5.8, where the engineered values of the masonry elastic modulus and compressive strength are employed alongside the spring stiffness and relative wall thickness.

$$T_Q = (E_m f_k K_s)^{0.46}(\lambda - 0.023) \tag{5.8}$$

With the newly derived expression for the *thrust*, T_Q, a revised force-capacity parity plot incorporating this additional variable was developed. The resulting parity plot is presented in Figure 5.16. An improved correlation is observed compared with the expression developed in the first iteration.

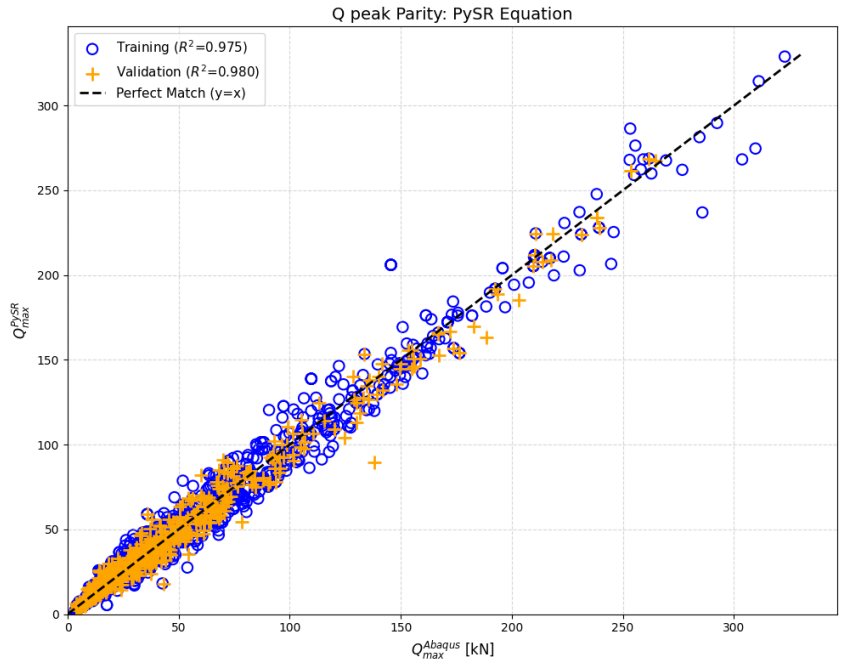


Figure 5.16: Force capacity with equation of newly developed thrust force expression.

The equation developed can be seen in Equation 5.9, where the newly developed equation T_Q is utilized as well as the pre compressive force, slenderness and spring stiffness.

$$Q_{peak} = -4.2(F_O - T_Q) \cdot (\lambda - 3.42 \cdot 10^{-5} K_s) \tag{5.9}$$

The developed force capacity expression, utilizing the newly derived *thrust* relationship, was evaluated against the EC6 expression presented in Section 2.2. A parity plot comparing both expressions against the generated dataset is shown in Figure 5.17, where EC6 predictions are represented by blue markers and the PySR predictions by green markers. The developed expression demonstrates substantially improved accuracy and reliability compared to the EC6 formulation. Data points located above the *Perfect Match* line indicate overestimation, whereas points below indicate underestimation. According to the parity plot, the EC6 equation tends to overestimate the force capacity for walls with low force capacity, particularly simply supported walls, while underestimating walls with higher force capacities. This trend suggests that the EC6 formulation is unable to consistently capture the variation in force capacity across different boundary conditions represented in the dataset.

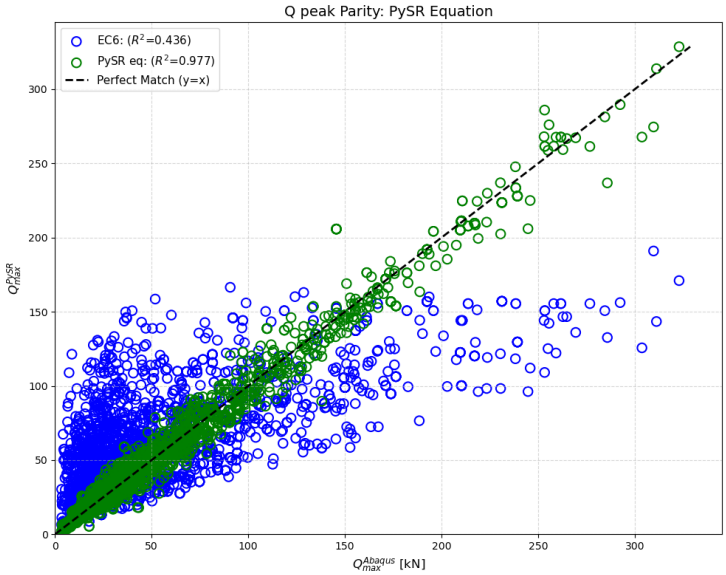


Figure 5.17: Comparison of the expression for force capacity in EC6 (blue) and PySR (red).

Beyond the ABAQUS-generated dataset, the developed expressions were also evaluated against the experimental tests, as presented in Figure 5.18. Although the dataset only includes single-wythe wall geometries, the developed expression was additionally applied to double-wythe walls to assess its broader applicability. As shown in the comparison, the newly developed expression performs better than the EC6 formulation across all considered cases, for both single- and double-wythe walls.

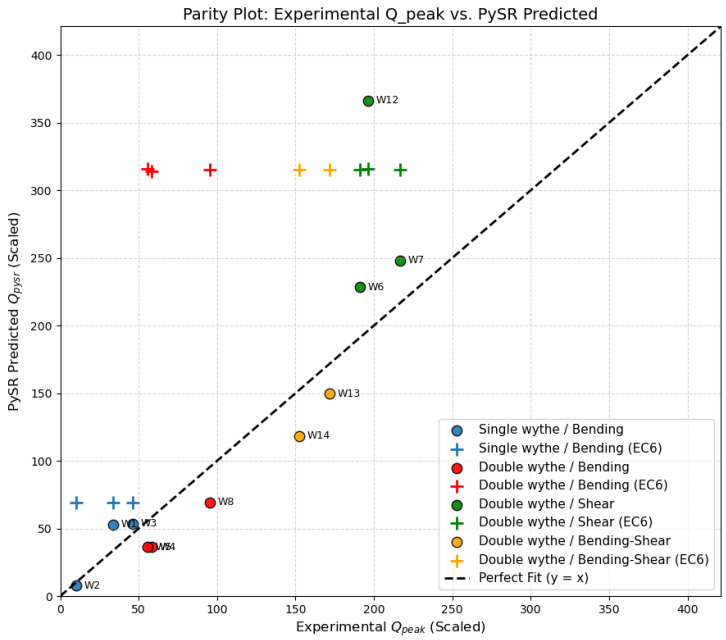


Figure 5.18: Comparison of EC6 (+) and PySR (dots) expressions against the experimental data.

It should, however, be noted that the applicability of the expression is limited to walls governed by bending failure. This limitation is evident in the prediction of wall W12, which experienced a shear-dominated failure mode and is therefore severely mis-predicted. Since the developed expression is derived from a dataset bounded by bending dominated behavior, material degradation and failure mechanisms associated with shear are not captured and are consequently outside the scope of the formulation.

5.3.3 Discussion of the PySR Prediction Results

The development of an expression that outperforms the analytical EC6 formulation has been successfully achieved, particularly for single-wythe walls governed by bending failure. For walls W2 and W3, the developed expression demonstrates very good agreement with the experimental results, exhibiting only minor deviations. Wall W1, however, presents the same discrepancy previously discussed in Section 5.1.2. As the ABAQUS model was unable to fully capture the experimental response of W1, this limitation also influences the confidence in the developed expression.

Extrapolation to double-wythe walls generally yields satisfactory results. For walls exhibiting bending or combined bending–shear failure, the predictions show improved agreement with experimental observations compared to EC6, although the force capacity is consistently underestimated.

6 Conclusion & Future Work

This chapter discusses the results of the predictive assessment of the force capacity of URM walls using ML, and compares these findings with analytical expressions available in the literature and design standards. Since a substantial part of the work was also dedicated to the development of a micro-model in ABAQUS, the obtained numerical results are also discussed, including their significance, development, and influence on the ML models. The discussion further establishes the basis for the conclusions of this thesis and identifies areas where additional research is required to further improve the prediction of force capacity.

The numerical model developed has as presented been an 2D micro model, containing the bricks and mortar with an interface in between their layers. For thin walls, as the single-wythe wall, computational time is still in an reasonable range of about 10-30 minutes per iteration. An increased wall thickness and a more complicated bond structure as that of a flemish bond in the double-wythe wall complicates the model immensely, even with compromises to the stability of the model, the models average time for the validation wall with a short height of 1.5 m was 3 hours. This became a bottle neck in the development of a dataset of the double wythe wall. Because of this a dataset with double-wythe walls where not able to be created, as not enough results where able to be produced.

Beyond the computational time associated with the double-wythe walls, the correct failure behavior was never fully captured, as demonstrated in Appendix E. Although the force capacity could be reproduced, this was only achieved by calibrating the tensile strength of the bricks to enforce a failure mechanism similar to that observed experimentally. As discussed in Section 3.2.2, ABAQUS employs a modified Drucker–Prager yield criterion through the CDP model. This constitutive formulation governs the interaction between compressive and tensile behavior within a shared stress space and may therefore play a significant role in the development of shear failure. In its current form, the model adopts a standard CDP formulation, and it is possible that modifying the internal CDP parameters could facilitate a more representative failure mechanism, rather than relying on a reduction in brick tensile strength. However, how such calibration should be performed remains uncertain, as does the extent to which it is physically meaningful or applicable across different brick materials. This also raises questions regarding the viable parametric range over which such modifications remain valid. From an engineering perspective, the resulting expressions or parameter ranges would likely become highly dependent on these calibrated variables, which may be difficult to estimate in practice. Consequently, extensive material testing may be required to accurately predict shear failure in masonry walls.

For the purpose of creating a dataset suitable for ML, results must be as accurate and noise-free as possible. For the single-wythe wall, this has largely been achieved and validated against a limited set of experimental tests, where two out of three specimens were able to reproduce the peak force capacity. A larger number of ex-

perimental tests, particularly for walls with varying heights, would further increase confidence in the numerical model, provided that the modeling approach is correctly formulated. This would subsequently strengthen the reliability of the ML models and derived expressions, as ML models are inherently dependent on the quality of the feature data used for training.

Comparing the MLP model results in Subsection 5.1.1 and the SR expression results in Subsection 5.3.2 shows that both have similar predictions accuracy. With the MLP model being a bit more accurate for the extrapolated experimental tests. As both models need values of compressive strength and elastic modulus for both materials, the need for an analytical expression deemed unnecessary, as it gives no additional benefit if both models are available. As the EC6 expression in Section 2.2 utilizes the compressive strength as well it could be argued the newly developed expression is an improvement even if it needs the elastic modulus also. From an engineering perspective, however, the applicability of the developed expression depends on a relatively large set of experimentally determined material parameters for both brick and mortar. Since these properties must be obtained through material testing, the practical implementation of the expression may be challenging. If experimental testing where conducted both compressive strength and elastic modulus would be needed to be retrieved.

The Neural ODE, while modeled after a reduced order model (ROM), can capture the behavior of the force-displacement evolution well if the properties of the studied specimen lies within the boundaries of the dataset. It is apparent that the model suffer more difficulties in extrapolation exercises than other models. This stems from the fact that shapes of the force-displacement curves changes between failure modes while the NODE model expects to capture all behaviors within the same bending failure regime. The main cause of the prediction difficulties is as discussed previously, that other geometries and failure modes failed to be modeled in ABAQUS and therefore the model cannot be trained to predict these behaviors. If such data is obtained however, the model have shown great potential to predict behaviors present in the dataset. The same model outline could therefore be used in further purposes to capture a bigger variety of specimens. As the NODE model is very sensitive to varying shapes for the same feature inputs, further development of the model can include an integrated classification model to alert the NODE of the failure mode beforehand. This would support the main NODE model to predict a more accurate force-displacement evolution despite its varying shape due to altering failure modes. This also becomes more apparent if the model is used to predict past the point of the peak force. As failure modes dictates the degradation and final failure, a classification module would be even more necessary if applied to such tasks. Further, the model could be expanded on for more dynamical loading applications such as blast loading. By re-defining the NODE to mirror the dynamic ROM, the model could be applied to predict the dynamical behavior of URM walls.

Expanding the work to incorporate the shear failure in the models is the next step to fully capture the OOP reaction force capacity of URM walls. Certain regards for the models might be needed, parameters currently used and developed have been

proven important for the bending failure of the walls and with a dataset including shear these might need to be expanded to incorporate other mechanical properties that dictate the failure, such as tensile strength.

In regards to the research questions asked in Section 1.3, it has been shown that the OOP peak force capacity can be predicted with the help of ML for bending failure in all three support conditions. With a slimmed down amount of parameters to what has been parametrized in the numerical model. Thus making the model easier and more useful for the assessments for OOP bending failures.

The ability to predict the force-displacement evolution in an URM wall can be deemed partly successful. For URM wall constructions with properties residing within the successfully observed data the Neural ODE model have shown great predictive abilities. Another conclusion comes with the fact of the difficulties of predicting the force-displacement curve for a wider range of geometries and failure modes. Although this is not because of the model architecture itself but on the fact of missing data. But for a specific type of URM wall construction with single-wythe geometry subjected to bending failure, the NODE model is able to predict the evolution of the force capacity until it reaches the peak force.

Lastly it has also been shown that an analytical expression can be created with the help of symbolic regression models capturing all the kinds of support conditions applicable. With an improved explanation how to handle the vagueness of the rigidness of the top support. Even if unit conversion is not applicable to the models an prediction of the OOP peak reaction force can be made. Beyond the OOP reaction force an expression for the thrust force has been able to be developed, with a lower accuracy but as a mean to predict the OOP peak force.

References

Abrams, D. P., Eeri, M., Angel, R., & Uzarski, J. (1996). *Out-of-plane strength of unreinforced masonry infill panels* (tech. rep.).

Anderson, .-. C. (1984). *Arching action in transverse laterally loaded masonry wall panels* (tech. rep.).

Arksey, H., & O'Malley, L. (2005). Scoping studies: Towards a methodological framework. *International Journal of Social Research Methodology*, *8*(1), 19–32. https://doi.org/10.1080/1364557032000119616

Avasthi, R., Kumar, S., & Rai, D. C. (2026). Behavior of masonry arch bridges under seismic load conditions and ecc strengthening. *Soil Dynamics and Earthquake Engineering*, *206*, 110258. https://doi.org/10.1016/j.soildyn.2026.110258

Binici, H., Binici, F., Akcan, M., Yardim, Y., Mustafaraj, E., & Corradi, M. (2020). Physical–mechanical and mineralogical properties of fired bricks of the archaeological site of harran, turkey. *Heritage*, *3*(3), 1018–1034. https://doi.org/10.3390/heritage3030055

Bokan Bosiljkov, V., Padovnik, A., & Turk, T. (Eds.). (2023). *Conservation and restoration of historic mortars and masonry structures: Hmc 2022* (Vol. 42). Springer. https://doi.org/10.1007/978-3-031-31472-8

Bonate, P. L. (2001). A brief introduction to monte carlo simulation. *Clinical Pharmacokinetics*, *40*(1), 15–22. https://doi.org/10.2165/00003088-200140010-00001

Cabané, A., Pelá, L., & Roca, P. (2022). Anisotropy and compressive strength evaluation of solid fired clay bricks by testing small specimens. *Construction and Building Materials*, *344*, 128195. https://doi.org/10.1016/j.conbuildmat.2022.128195

CAE Assistant. (2025). *Abaqus standard vs explicit explained: Pick the right one* [Accessed: 2026-03-23]. https://caeassistant.com/blog/abaqus-standard-or-abaqus-explicit/

Ciarlet, P., & Lunéville, E. (2023). *The finite element method: From theory to practice*. ISTE Ltd; John Wiley & Sons, Inc.

Costigan, A., Pavia, S., & Kinnane, O. (2015). An experimental evaluation of prediction models for the mechanical behavior of unreinforced lime-mortar masonry under compression. *Journal of Building Engineering*, *4*, 283–294. https://doi.org/10.1016/j.jobe.2015.10.001

Cranmer, M. (2023). Interpretable machine learning for science with pysr and symbolicregression.jl. *arXiv preprint arXiv:2305.01582*. https://doi.org/10.48550/arXiv.2305.01582

D'Altri, A. M., Sarhosis, V., Milani, G., Rots, J., Cattari, S., Lagomarsino, S., Sacco, E., Tralli, A., Castellazzi, G., & de Miranda, S. (2020). Modeling strategies for the computational analysis of unreinforced masonry structures: Review and classification. *Archives of Computational Methods in Engineering*, *27*, 1153–1185. https://doi.org/10.1007/s11831-019-09351-x

Dassault Systèmes SIMULIA. (2017). *Using python and the abaqus scripting interface*. Retrieved January 23, 2026, from https://abaqus-docs.mit.edu/2017/English/SIMACAECMDRefMap/simacmd-m-IntPythonandacl-sb.htm

Dassault Systèmes SIMULIA. (2024). *Abaqus nonlinear analysis guide*. Retrieved January 23, 2026, from https://docs.software.vt.edu/abaqusv2024/English/TsoUserMap/tso-c-usr-solver-abaqus-nonlinguide.htm

Dassault Systèmes Simulia Corp. (2024). *Abaqus documentation, version 2025* [Available at https://help.3ds.com]. Dassault Systèmes.

de Borst, R. (1991). Simulation of strain localization: A reappraisal of the cosserat continuum. *Engineering Computations*, *8*(4), 317–332. https://doi.org/10.1108/eb023828

Edri, I. E., & Yankelevsky, D. Z. (2018). Analytical model for the dynamic response of blast-loaded arching masonry walls. *Engineering Structures*, *176*, 49–63. https://doi.org/10.1016/j.engstruct.2018.08.053

Eurocode 1: Actions on structures. (2002). European Committee for Standardization (CEN).

European Committee for Standardization. (2006). *Eurocode 6: Design of masonry structures – part 1-1: General rules for reinforced and unreinforced masonry structures* (EN 1996-1-1). European Committee for Standardization. Brussels.

Federal Emergency Management Agency. (1997, October). *Nehrp guidelines for the seismic rehabilitation of buildings* (tech. rep. No. FEMA 273) (Prepared for the Building Seismic Safety Council by the Applied Technology Council (ATC-33 Project)). Federal Emergency Management Agency (FEMA). Washington, D.C.

Fiala, J., Mikolas, M., Fiala, J. J., & Krejsova, K. (2019). History and evolution of full bricks of other european countries. *IOP Conference Series: Materials Science and Engineering*, *603*, 032097. https://doi.org/10.1088/1757-899X/603/3/032097

Friesem, D., Boaretto, E., Eliyahu-Behar, A., & Shahack-Gross, R. (2011). Degradation of mud brick houses in an arid environment: A geoarchaeological model. *Journal of Archaeological Science*, *38*, 1135–1147. https://doi.org/10.1016/j.jas.2010.12.011

Garijo, L., Zhang, X., Ruiz, G., Ortega, J. J., & Yu, R. C. (2018). Mechanical behavior of natural hydraulic lime mortars. In *Sustainable construction and building materials*. IntechOpen. https://doi.org/10.5772/intechopen.80852

Godio, M., & Flansbjer, M. (2025a). *Blast capacity of masonry walls – modelling of static and impact loads* (tech. rep. No. CFORT Work Report 2025:04). Centrum för Fortifikatorisk Kompetens (CFORT).

Godio, M., & Flansbjer, M. (2025b). Experimental response of unreinforced brick masonry walls to moderate-velocity point-load and line-load impacts. *International Journal of Impact Engineering*, *206*, 105461. https://doi.org/10.1016/j.ijimpeng.2025.105461

Godio, M., Flansbjer, M., & Portal, N. W. (2023a). Single- and double-wythe brick masonry walls subjected to four-point bending tests under different support

conditions: Simply supported, rigid, non-rigid. *Construction and Building Materials*, *404*. https://doi.org/10.1016/j.conbuildmat.2023.132544

Godio, M., Flansbjer, M., & Portal, N. W. (2023b). Low-velocity out-of-plane impact tests on double-wythe unreinforced brick masonry walls instrumented with optical measurements. *International Journal of Impact Engineering*, *178*. https://doi.org/10.1016/j.ijimpeng.2023.104597

Godio, M., Williams Portal, N., Flansbjer, M., Magnusson, J., & Byggnevi, M. (2021). Experimental and numerical approaches to investigate the out-of-plane response of unreinforced masonry walls subjected to free far-field blasts. *Engineering Structures*, *239*, 112328. https://doi.org/10.1016/j.engstruct.2021.112328

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* [http://www.deeplearningbook.org]. MIT Press.

Gustafsson, T. (2023). *Murat byggande*. AB Svensk Byggtjänst.

Hamilton, R. I., & Papadopoulos, P. N. (2024). Using shap values and machine learning to understand trends in the transient stability limit. *IEEE Transactions on Power Systems*, *39*(1), 1384–1397. https://doi.org/10.1109/TPWRS.2023.3248941

Hillerborg, A., Modéer, M., & Petersson, P.-E. (1976). Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements. *Cement and Concrete Research*, *6*(6), 773–782. https://doi.org/10.1016/0008-8846(76)90007-7

Johansson, M., & Laine, L. (2012). *Bebyggelsens motståndsförmåga mot extrem dynamisk belastning del 1: Last av luftstötvåg*. Myndigheten för samhällsskydd och beredskap.

Kallas, J., & Napolitano, R. (2024). Understanding critical masonry building attributes shaping vulnerability to blast loads: Data-driven insights from the 2020 beirut explosion. *International Journal of Disaster Risk Reduction*, *110*. https://doi.org/10.1016/j.ijdrr.2024.104640

Karaman, S., Ersahin, S., & Gunal, H. (2006). Firing temperature and firing time influence on mechanical and physical properties of clay bricks. *Journal of Scientific and Industrial Research*, *65*(2), 153–159.

Kaushik, H. B., Rai, D. C., & Jain, S. K. (2007). Stress-strain characteristics of clay brick masonry under uniaxial compression. *Journal of Materials in Civil Engineering*, *19*(9), 728–739. https://doi.org/10.1061/(ASCE)0899-1561(2007)19:9(728)

Kim, D.-K., & Lim, W.-Y. (2025). Correlation between brick and mortar material properties and brick masonry prism compressive strength and elastic modulus. *Journal of Building Engineering*. https://doi.org/10.1016/j.jobe.2025.113561

Kronberger, G., Burlacu, B., Kommenda, M., Winkler, S. M., & Affenzeller, M. (2025). *Symbolic regression* (1st ed.). CRC Press. https://doi.org/10.1201/9781315166407

Lee, G., Lee, K., Park, J. H., Lee, C. H., & Lee, S.-M. (2024). Mechanical and structural investigation of traditional masonry systems with diverse types of bricks and hydrated lime mortars. international journal of architectural

heritage. *International Journal of Architectural Heritage*. https://doi.org/10.1080/15583058.2023.2197405

Li, J., Lu, W., Zhang, X., & Zhang, Z. (2025). Experimental investigation on the mechanical properties of bricks subjected to triaxial compression. *Journal of Building Engineering*, *104*, 112386. https://doi.org/10.1016/j.jobe.2025.112386

Lourenço, P. B. (1996). *Computational strategies for masonry structures* [Doctoral dissertation, Delft University of Technology] [Ph.D. thesis]. https://resolver.tudelft.nl/uuid:4f5a2c6c-d5b7-4043-9d06-8c0b7c9f1f6f

Lourenço, P. B., Rots, J. G., & Blaauwendraad, J. (1995). Two approaches for the analysis of masonry structures: Micro and macro-modeling. *HERON*, *40*(4), 313–340.

Malathy, R., Shanmugam, R., Dhamotharan, D., Kamaraj, D., Prabakaran, M., & Kim, J. (2023). Lime based concrete and mortar enhanced with pozzolanic materials – state of art. *Construction and Building Materials*, *390*, 131415. https://doi.org/10.1016/j.conbuildmat.2023.131415

Manoharan, A., & Umarani, C. (2022). Lime mortar, a boon to the environment: Characterization case study and overview. *Sustainability*, *14*(11), 6481. https://doi.org/10.3390/su14116481

Martens, D. R. W., Vermeltfoort, A. T., Martens, D. R. W., & T, A. (2017). *Arching effect in laterally loaded urm walls* (tech. rep.). www.tue.nl/taverne

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

Mitchell, T. M. (1997). *Machine learning* [https://www.cs.cmu.edu/~tom/mlbook.html]. McGraw-Hill Science/Engineering/Math.

Moumni, B., Oulmekki, A., Kizinievic, O., Kizinievic, V., Eliche-Quesada, D., Charroud, M., El Moudden, N., & Benmoussa, H. (2024). Clay influence on lightweight brick's properties: Investigating the impact of waste's nature and amount as secondary variables. *Construction and Building Materials*, *438*, 136844. https://doi.org/10.1016/j.conbuildmat.2024.136844

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.

Najafgholipour, M. A., & Zohrei, M. (2026). Predictive models for the in-plane strength and drift capacity of unreinforced masonry walls with various types of failure modes using machine learning algorithms. *Bulletin of Earthquake Engineering*, *24*, 2661–2684. https://doi.org/10.1007/s10518-026-02389-w

Natário, P., Silvestre, N., & Camotim, D. (2014). Web crippling failure using quasi-static fe models. *Thin-Walled Structures*, *84*, 34–49. https://doi.org/10.1016/j.tws.2014.05.003

Nazimi, K., Castro, J. J., Omi, S., & Stanikzai, M. A. (2024). Investigating the compressive strength of clay brick masonry: A case study of nangarhar, afghanistan. *Buildings*, *14*(12), 3882. https://doi.org/10.3390/buildings14123882

Nielsen, M. A. (2015). *Neural networks and deep learning*. Determination Press. http://neuralnetworksanddeeplearning.com

Rezaie, A., Godio, M., Achanta, R., & Beyer, K. (2022). Machine-learning for damage assessment of rubble stone masonry piers based on crack patterns. *Au-*

tomation in Construction, *140*, 104313. https://doi.org/10.1016/j.autcon.2022.104313

Shi, Y., Wang, N., Li, Z.-X., & Ding, Y. (2021). Experimental studies on the dynamic compressive and tensile strength of clay brick under high strain rates. *Construction and Building Materials*, *272*, 121908. https://doi.org/10.1016/j.conbuildmat.2020.121908

Singh, S. B., & Munjal, P. (2017). Bond strength and compressive stress-strain characteristics of brick masonry. *Journal of Building Engineering*, *9*, 10–16. https://doi.org/10.1016/j.jobe.2016.11.006

Soleymani, A., Najafgholipour, M. A., & Johari, A. (2022). An experimental study on the mechanical properties of solid clay brick masonry with traditional mortars. *Journal of Building Engineering*. https://doi.org/10.1016/j.jobe.2022.105057

Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, *15*(1), 72–101. https://doi.org/10.2307/1412159

Swedish Institute for Standards (SIS). (2012). *Eurocode 6: Part 1-1: General rules for reinforced and unreinforced masonry structures*. European Committee for Standardization.

Udrescu, S.-M., & Tegmark, M. (2020). Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, *6*(16), eaay2631. https://doi.org/10.1126/sciadv.aay2631

Wohlin, C., Kalinowski, M., Romero Felizardo, K., & Mendes, E. (2022). Successful combination of database search and snowballing for identification of primary studies in systematic literature studies. *Information and Software Technology*, *147*, 106908. https://doi.org/10.1016/j.infsof.2022.106908

Worsham, J. M., & Kalita, J. K. (2025). A guide to neural ordinary differential equations: Machine learning for data-driven digital engineering. *Digital Engineering*, *6*, 100060. https://doi.org/10.1016/j.dte.2025.100060

Yadav, A., & Pal, S. (2023). The impact of mortar thickness and strength on the brick masonry prism: An investigation. *Materials Today: Proceedings*, *93*, 552–559. https://doi.org/10.1016/j.matpr.2023.10.035

Yucel, G. (2025). Effect of macro- and meso-scale modeling approaches on the numerical analysis of historical masonry minarets: A comparative case study. *Structures*, *82*, 110783. https://doi.org/10.1016/j.istruc.2025.110783

A Appendix 1 - Mass scaling

To reduce computational time, an artificial increase in mass is utilized (CAE Assistant, 2025), which is a common approach in quasi-static analyses (Natário et al., 2014). By artificially increasing the material density of parts of the model, the stable time increment increases, thereby reducing the number of increments required to complete a simulation. The relationship between density and the stable time increment is shown in Equations A.1.

$$\Delta t = \frac{L_e}{c_d} \tag{A.1a}$$

where L_e is the characteristic element length and c_d is the dilatational wave speed of the material.

$$c_d = \sqrt{\frac{E}{\rho}} \tag{A.1b}$$

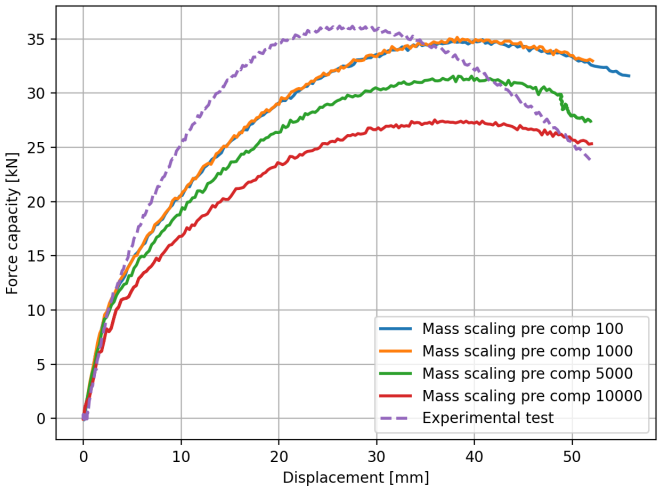
Where E is Young's modulus and ρ is the material density. By increasing the density by a factor f, the wave speed decreases by a factor of $\sqrt{f}$, while the stable time increment increases by the same factor. However, caution must be exercised when applying mass scaling, as it also affects inertial forces and may lead to inaccurate results. The following sections will further examine the mass scaling applied in the current model.

A.1 Mass scaling

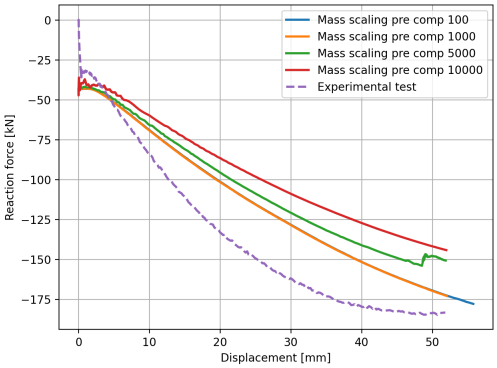
Since the model consists of two steps, the influence of mass scaling must be evaluated separately for each to understand its effect on the model's output.

To assess this, all variables were kept constant while the mass scaling factor was varied. For the pre-compressive step, four iterations were evaluated, with mass scaling values ranging from 100 to 10,000. The corresponding effects on force capacity, reaction force, and uplift are presented in Figure A.1.

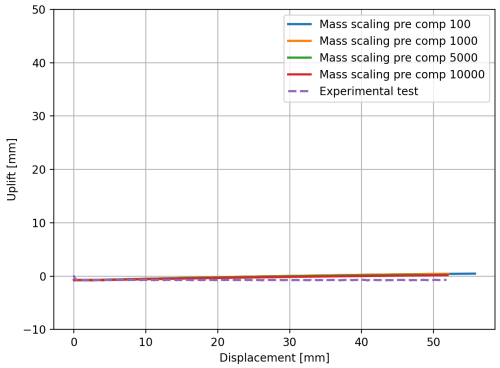
The results show that mass scaling factors of 100 and 1000 have negligible influence on the model response. However, at a scaling factor of 5000, both the force capacity and reaction force begin to decrease, with further reductions observed at higher scaling values. The uplift remains very small across all cases, and no noticeable differences can be observed between the different iterations.



(a) OOP reaction force vs horizontal displacement in pre compressive step.



(b) Thrust force vs horizontal displacement in both steps.



(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure A.1: Mass scaling increase effect in the pre compressive step with the validation of wall W3.

Based on the results, the mass scaling factor should not exceed 1000. The computational time associated with the different mass scaling values is presented in Table A.1, where an increase in mass scaling leads to a reduction in simulation time.

Table A.1: Computational time of different mass scaling in the pre compressive step

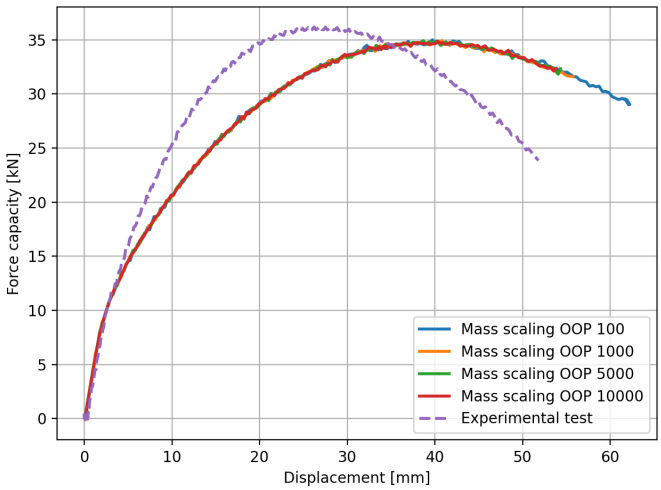
Mass scaling	100	1000	5000	10 000
Time [s]	2375	1485	1334	1318
Time [min]	39.6	24.8	22.2	22.0

Increasing the mass scaling factor from 100 to 1000 reduces the computational time by approximately 15 minutes. A mass scaling factor of 1000 provided the best approximation of the experimental response. However, in the development of Figure A.1, the experimental value of the mortar elastic modulus had to be reduced to

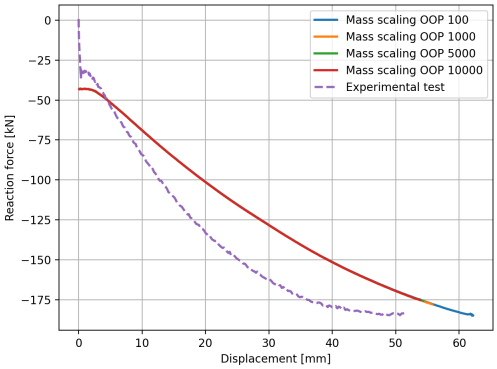
achieve agreement with the results. For the creation of the dataset, a mass scaling factor of 10000 was used together with the experimental value of the mortar elastic modulus. Although mass scaling affects the results, the model is considered to perform as intended.

A.2 Out of plane step mass scaling

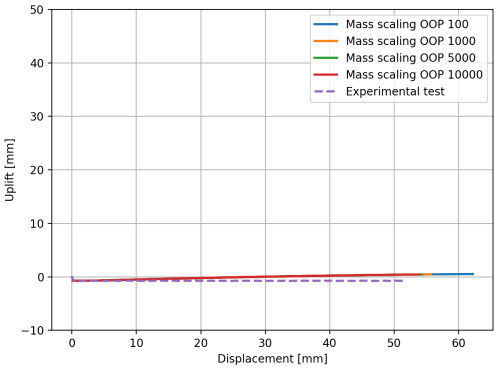
The second step out of plane (OOP) is examined based on the same increment in mass scaling. A negligible difference between the iterations is observed in Figure A.2, showing that an increase does not change the results for these parameters.



(a) OOP reaction force vs horizontal displacement in pre compressive step.



(b) Thrust force vs horizontal displacement in both steps.



(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure A.2: Mass scaling increase effect on the out of plane step with the validation of wall W3.

However, a decrease in computational time can be observed with increasing mass scaling, as shown in Table A.2. Increasing the mass scaling factor from 100 to

5000 reduces the computational time by 51.6 minutes. A further increase results in an additional reduction of only 5 minutes. However, since different parameters will be used, such an increase is not considered sufficiently beneficial to justify its implementation.

Table A.2: Computational time of different mass scaling in the out of plane step

Mass scaling	100	1000	5000	10 000
Time [s]	4736	2377	1640	1417
Time [min]	78.9	39.6	27.3	23.6

Therefore an mass scaling of 1000 - 5000 is determined to be utilized in the OOP step.

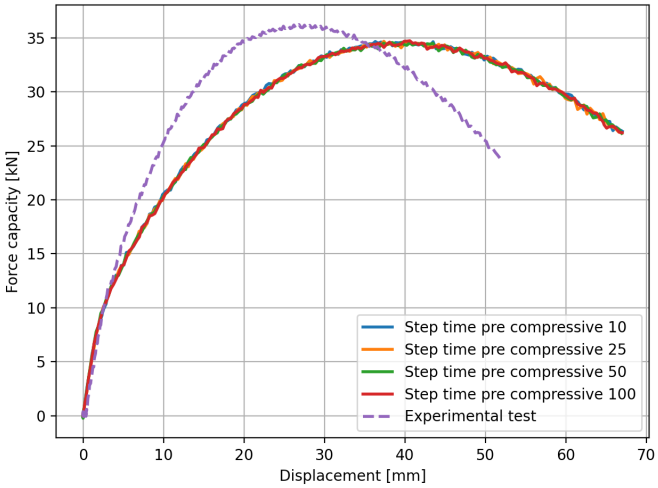
B Appendix 2 - Time step

A quasi static analysis in ABAQUS explicit requires a certain types of tactics to ensure the model stays static in its solution (Dassault Systèmes Simulia Corp., 2024). The explicit solver is developed as a way to model high speed impact events where inertia plays a large role, for a static solution the inertia is the opposite and is insignificant in the solution. Modeling the entire time period of the quasi static tests would be extremely computational time draining and therefore the time steps is reduced to a manageable scale while still keeping the inertia effect of the model insignificant. The following sections will go through the size of the time steps and what effect they have on the model.

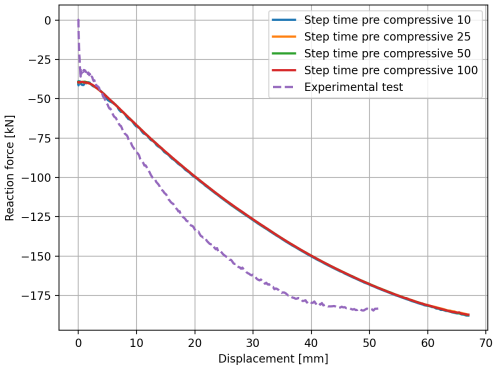
For a quasi-static solution, the external work should remain equal compared to the internal energy of the model (Dassault Systèmes Simulia Corp., 2024). With a commonly used criterion is that the ratio of kinetic energy to internal energy should not exceed 5% to ensure quasi-static conditions. Since this validation is performed on only one wall, and the parameterization may influence the results, a lower ratio is targeted to provide a greater margin of confidence.

B.1 Pre - compressive time step

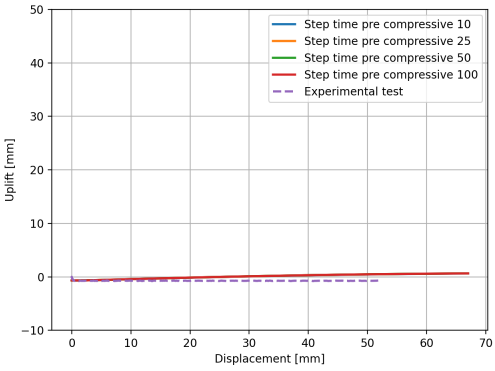
For the pre compressive time step the load applied at the top of the wall is implemented with a smoothly over the entire period. As the time step increases the slower the force is applied. As such the time steps selected is from 10-100 seconds and the results is pictured in Figure B.1. Based on this the resulting time step of the load has a minimal effect on the results.



(a) OOP reaction force vs horizontal displacement in pre compressive step.



(b) Thrust vs horizontal displacement in both steps.



(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure B.1: Different step times effect on the reaction forces in the pre compressive step.

In the computational time and max ratio in between the kinetic and internal energy observed in Table B.1 below, The kinetic energy is kept low regardless of the duration of the pre compressive time step. An increase is however evident with a larger time step, from 5 minutes to 8 minutes.

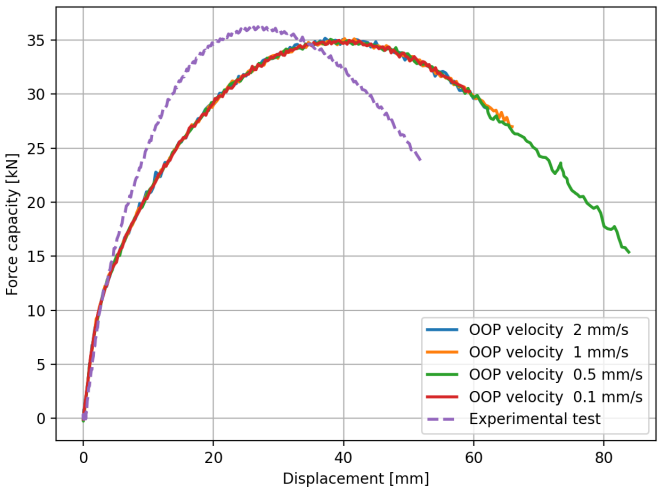
Table B.1: Computational time of different mass scaling in the pre compressive step

Time step	10	25	50	100
Time [s]	305	326	390	494
Time [min]	5.1	5.4	6.5	8.3
Max ratio KE/IE [%]	0.146	0.101	0.099	0.099

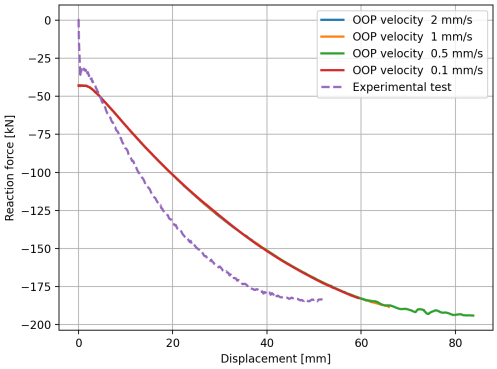
Based on the evaluation an pre compressive time step of 50 is selected, as the applied vertical force in parametrization will be higher and thus an increase of external force has to be taken into account.

B.2 Out of plane time step

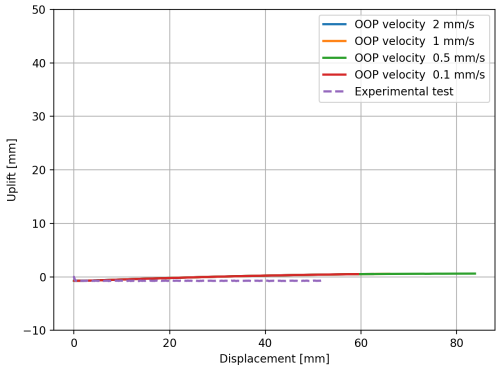
The OOP step has the velocity in the horizontal direction of the wall been iteratively tried. A note that the initial velocity is added in a smooth fashion over a time period of ten seconds to reduce the noise emerging from the initial kinetic energy peak (Natário et al., 2014). Results from the different velocities is illustrated in plots in Figure B.2 below. Based on these the no large deviations can be seen of the different iterations.



(a) OOP reaction force vs horizontal displacement in pre compressive step.



(b) Thrust vs horizontal displacement in both steps.



(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure B.2: Different step times effect on the reaction forces in the out-of-plane step.

The time and the resulting ratio of the kinetic and internal energy is shown in Table B.2, where a faster velocity gives reduces the computational time, while increasing the ratio in the model.

Table B.2: Computational time of different step times of the out-of-plane step, with the max ratio of the kinetic and internal energy.

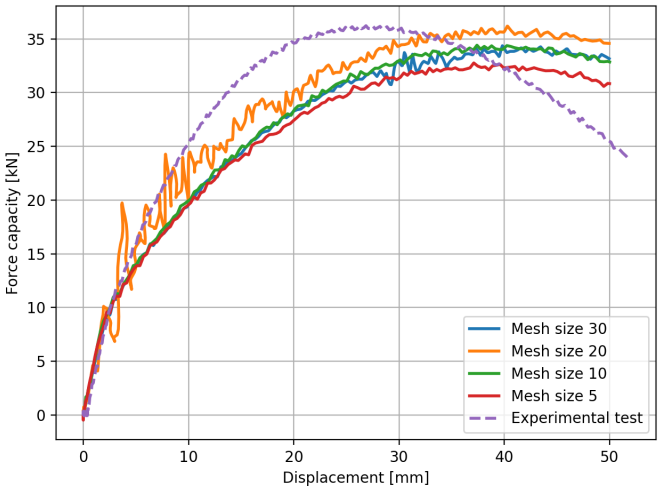
OOP velocity [mm/s]	2	1	0.5	0.1
Time [s]	326	401	458	1031
Time [min]	5.4	6.7	7.6	17.2
Max ratio KE/IE [%]	2.4	1.1	0.4	0.06

A velocity of 0.5 mm/s gives a ratio of 0.4 % and a computational time of 7.6 minutes. Which is deemed to be the most sensible result based on the selected parameters.

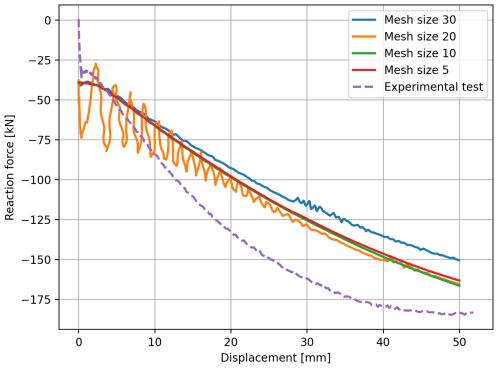
C Appendix 3 - Mesh size

Even though the material models for the mortar and bricks, based on the Concrete Damaged Plasticity (CDP) model, are regularized to reduce mesh dependency, a mesh convergence study is still required to ensure that the chosen mesh discretization does not influence the results (Dassault Systèmes Simulia Corp., 2024).
The mesh is defined under a plane strain assumption, as the thickness of the wall is large compared to its other dimensions. Triangular elements are used in the mortar layers to achieve higher mesh resolution in these regions. The bricks, being larger and containing a greater number of elements, are instead discretized using quadrilateral elements. The element sizes are kept consistent between triangular and quadrilateral meshes. The supports at the top and bottom of the model are represented by single, large quadrilateral elements in plane strain. This approach has been found to perform well for the present setup and reduces the total number of elements in the model significantly.

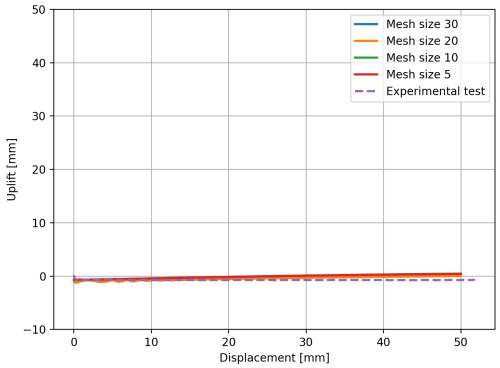
The selected mesh sizes are 30,20,10 and 5 and the results of these can be seen in Figure C.1, in (a) the maximum force capacity with the mesh size of 5 has a slight reduction with an refined mesh of approximately 1 kN. There should also be noted that an decrease from 30 to 20 increases the maximum, while an further decrease lowers it back again.
The reaction force in Figure C.1.b shows an reduction in the thrust for the mesh size of 30 while the other show minimal difference in between themselves. Giving uncertainties in both the mesh size of 30 and 20.



(a) OOP reaction force vs horizontal displacement in pre compressive step.



(b) Thrust force vs horizontal displacement in both steps.



(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure C.1: Different size mesh of the numerical model and the responses of the model.

Computational time and energy ratio are presented in Table C.1. The mesh sizes of 30 and 20 exhibit very low computational times (less than 3 minutes); however, the results are associated with uncertainties. In particular, the mesh size of 20 shows a high energy ratio.

Table C.1: Computational time of different mesh sizes in the numerical model for wall W3

Mesh size [mm]	30	20	10	5
Time [s]	96	141	374	1207
Time [min]	1.6	2.4	6.23	20.1
Max Ratio [KE/IE]	0.5	37	0.4	0.4

Reducing the mesh size to 5 increases the computational time to approximately 20 minutes, which is considered a significant increase compared to the mesh size of 10.

Therefore, a mesh size of 10 is selected as a suitable balance between computational efficiency and solution accuracy.

D Appendix 4 - Pre compressive force

Maximum of pre compression calculation

Geometry

$L := 3\ m$

$H := 3\ m$

$g := 9,81\ \frac{m}{s^2}$ Gravity constant

$W := 1\ m$

$t_f := 0,25\ m$

$t_m := 0,145\ m$

$S := 4$ Amount of floors

Weights

$\rho_{concrete} := 2500\ \frac{kg}{m^3}$

$\rho_{masonry} := 1715\ \frac{kg}{m^3}$

Loads

$q_s := 2\ \frac{kN}{m^2}$ Snow load

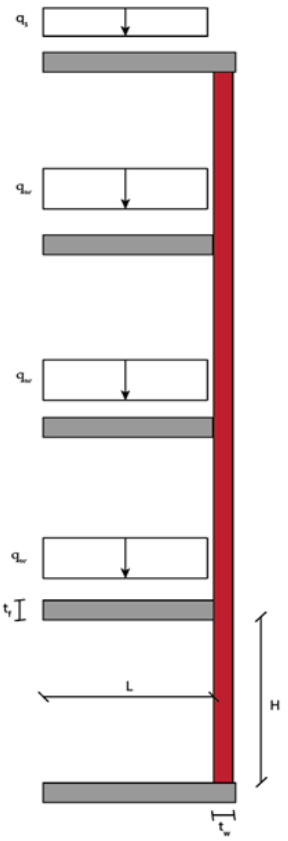
$q_w := 3\ \frac{kN}{m^2}$ Imposed load

$q_f := t_f \cdot L \cdot W \cdot S \cdot \rho_{concrete} \cdot g = 73,575\ kN$

$q_{k,f} := q_w \cdot (S-1) \cdot W \cdot L = 27\ kN$

$q_{s,R} := q_s \cdot W \cdot L = 6\ kN$

$q_W := t_m \cdot H \cdot S \cdot W \cdot \rho_{masonry} \cdot g = 29,274\ kN$



Maximum value of the pre compressive force

$O := q_f + q_{k,f} + q_{s,R} + q_W = 135,849\ kN$

Figure D.1: Simple calculation for the decision of the maximum value of the parametric range of the pre-compressive force

E Appendix 5 - Double wythe numerical model

A numerical model of a double wythe wall, similar to the one presented in Godio et al., 2023a, was developed and tested. However, the computational time required for each simulation became excessively large, making it impractical to generate a sufficiently large dataset for meaningful machine learning applications. Despite numerous attempts to simplify the model, stability issues arose during the parametrization process. In particular, failures involving extensive crushing, such as shear failures, resulted in significant energy releases that caused convergence difficulties. Nevertheless, the model that was successfully developed is presented in the following section.

E.1 Double wythe wall

The double wythe model is constructed to replicate the Flemish bond pattern presented in Godio et al., 2023a for 2D analysis, in order to ensure validation and improve the accuracy of the results. The general geometry of the model is illustrated in Figure E.1. Compared to the single-wythe model, the double wythe model includes additional components such as half-bricks and mortar head joints. However, to reduce numerical interactions and avoid instability, no vertical interactions between these components are defined, as this otherwise leads to model failure.

The gravitational load has been adjusted to reflect the geometric configuration of the wall. Aside from this modification, the model setup remains consistent with that of the single-wythe model.

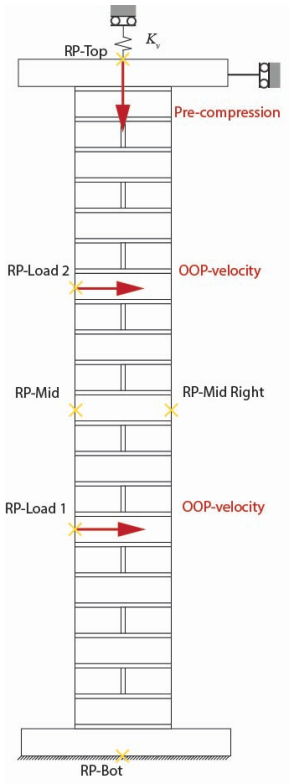
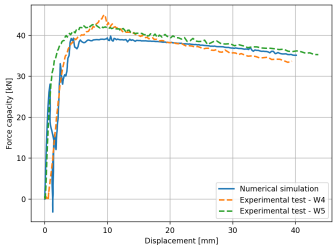


Figure E.1: Double wythe wall as modeled in ABAQUS, with reference points highlighted by yellow crosses and applied forces shown as red arrows.

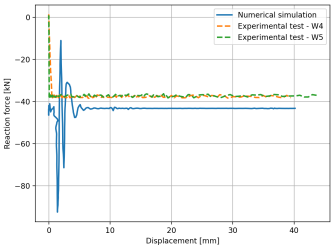
Mass scaling has been modified for the double wythe wall, as keeping the scaling factor resulted in reaching the maximum allowable increment size in ABAQUS. Due to this limitation, and within the parameterization of the model, the mass scaling in the out-of-plane (OOP) step was increased from 1,000 to 10,000 to reduce the computational time to somewhat reasonable.

E.2 Validation of double-wythe wall

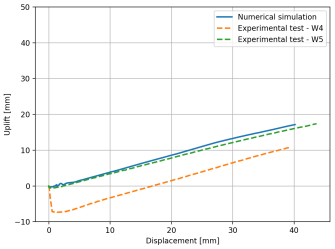
Three different walls where used for validation against the double-wythe wall seen in Godio et al., 2023a and presented in chapter2.1. Two of the walls (W4 and W5) where both simply supported with similar pre compressive force and results from these results are presented in Figure E.2. Within a good relationship between can be seen with both of the walls, small differences between the two experimental walls can be attributed manufacturing differences or material inconsistency, something discussed further in chapters 3.4.2.



(a) OOP reaction force vs horizontal displacement.



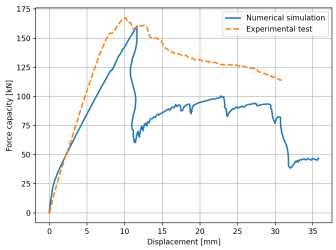
(b) Thrust force vs horizontal displacement



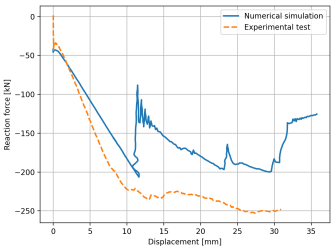
(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure E.2: Results from the numerical model (blue) compared with the experimental data of walls W4 and W5.

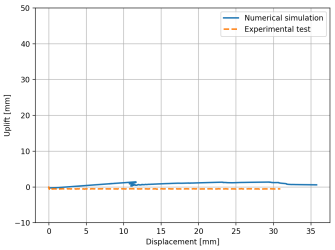
For the shear failure in W7 the tensile strength of the brick had to be reduced to 1/10 of the compressive strength instead of 2/10 of the inital measurement in the experimental tests. In Figure E.3 correlation before failure and at peak force is similar, however the failure of the wall is to brittle afterwards. As the searched for is the failure of peak force is well enough captured the model works as intended but has potential for improvements.



(a) OOP reaction force vs horizontal displacement.



(b) Thrust force vs horizontal displacement



(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure E.3: Results from the numerical model (blue) compared with the experimental tests of wall W7.

A screenshot of the wall at the final step in ABAQUS is displayed in Figure E.4, Shear failure can be observed diagionally starting from the supports. This failure moves through both the brick and the mortar head similar to that seen in the experimental walls seen in Chapter 2.1.

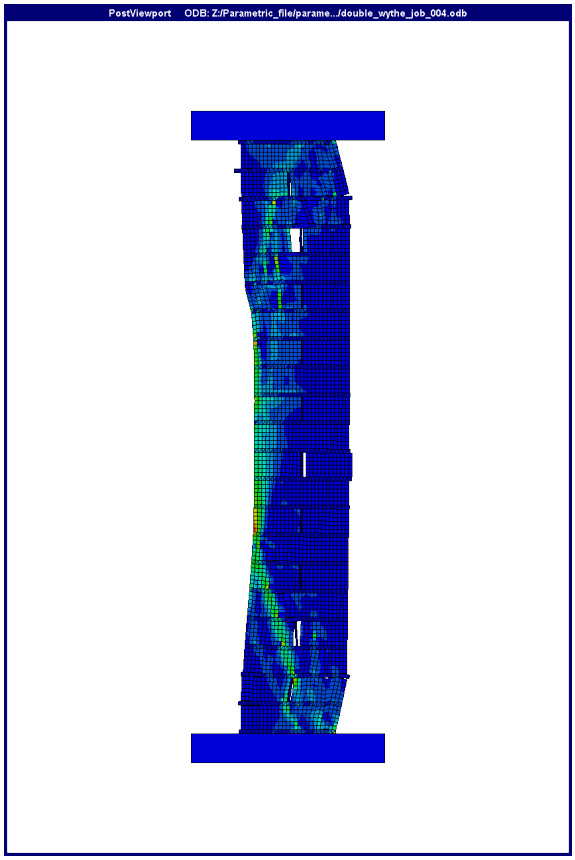
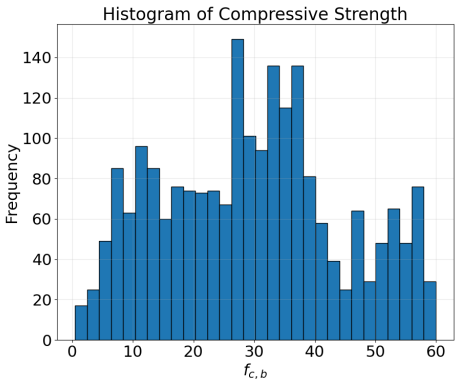


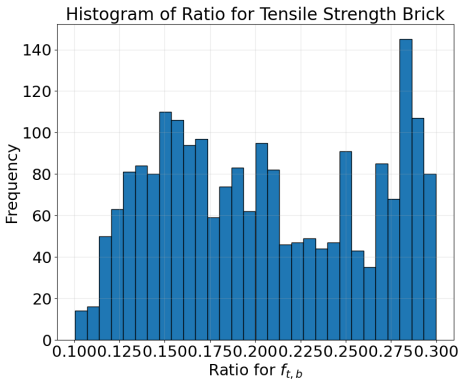
Figure E.4: Final picture of W7 from ABAQUS with shear failure close to the supports.

F Appendix 6 — Distribution of Parametric Values

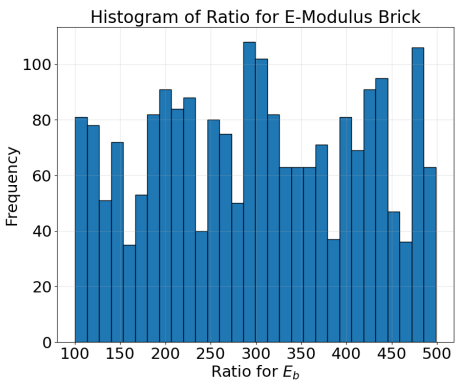
This appendix gathers up all of the parameters that has been varied through out the numerical model and there frequencies as they are occuring. As previously stated in chapter 3.4.6 the parametric range has been changed along with the creation of the dataset and as such some values will be more frequently represented. The Figure F.1 shows the histograms for the parameters that has been varied for the brick material, a fairly evenly spread distrubution.



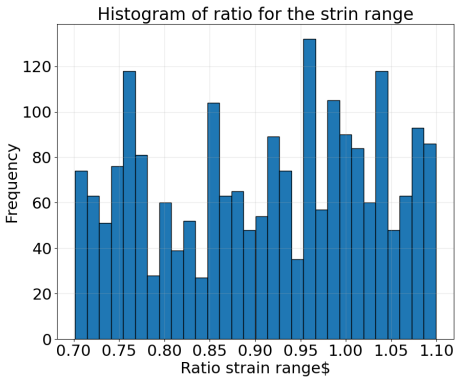
(a) Distribution of the compressive strength



(b) Distribution of compressive/tensile strength ratio



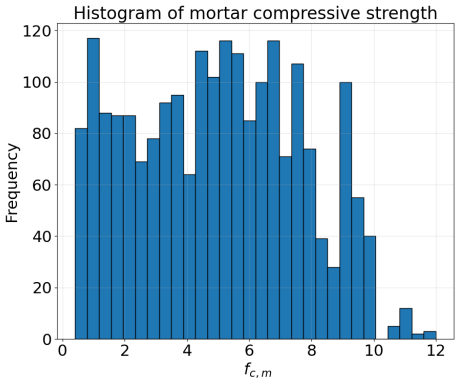
(c) Distribution of elastic ratio



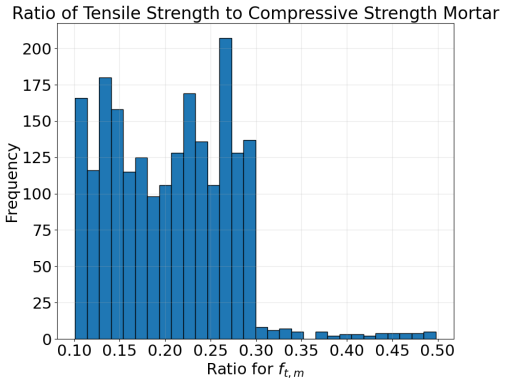
(d) Distribution of the strain range ratio

Figure F.1: Distribution of mechanical parameters affecting brick properties

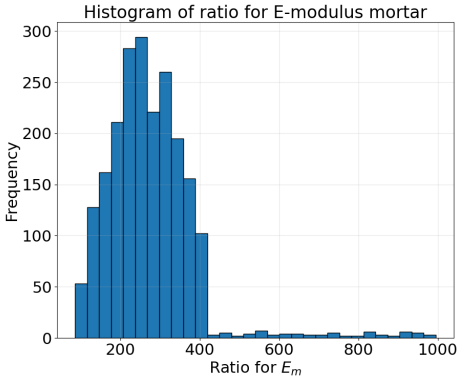
In Figure F.2 below, the histograms for the mortar parameters can be seen and here as well a fairly uniform distribution can be seen and the increase of the parametric range, for the compressive/tensile strength, elastic ratio and mortar height range.. Both mortar and compressive strength has a few values beyond the parametric range.



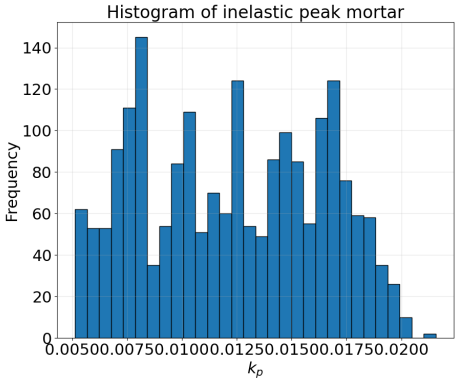
(a) Distribution of the compressive strength



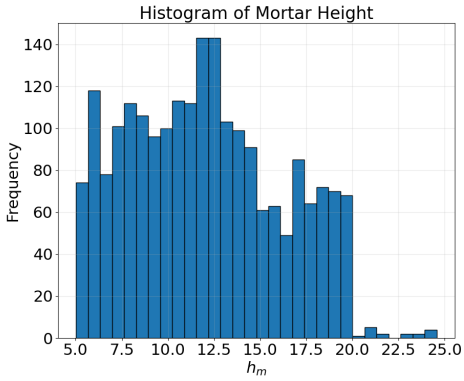
(b) Distribution of compressive/tensile strength ratio



(c) Distribution of elastic ratio



(d) Distribution of the strain range

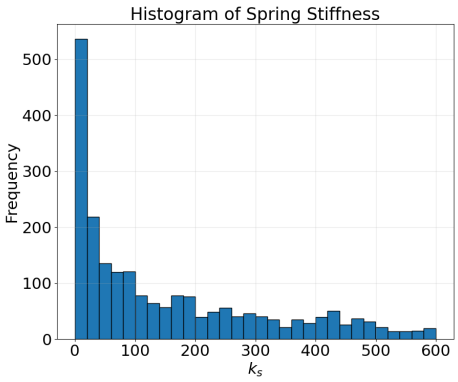


(e) Distribution of the mortar height range

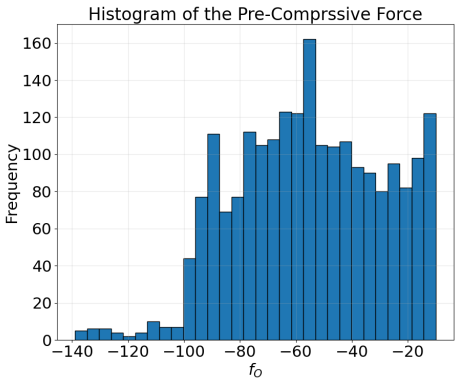
Figure F.2: Distribution of mechanical parameters affecting mortar properties

The last histograms in Figure F.3 represent the global parameters of the model. For the spring stiffness, lower stiffness values are more strongly represented in the dataset in order to better capture the response of simply supported or non-rigid wall behaviour.

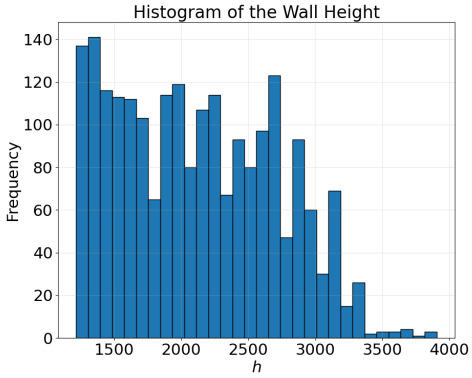
For the pre-compressive force, the expansion of the parametric range resulted in fewer samples with values exceeding 100. In the case of wall height, a decrease in the frequency of samples is observed for taller walls. This is associated with the increased instability of higher walls, where the kinetic/internal energy threshold is exceeded more frequently and sorted out.



(a) Distribution of spring stiffness



(b) Distribution of pre compressive force



(c) Distribution of wall height

Figure F.3: Distribution of global parameters effecting the walls

G Appendix 7 — Direct SR Prediction of the Force capacity

A parity plot for the prediction of the first iteration is found in Figure G.1 below. A correlation between the the expression and the target value indicate that the expression captures the force capacity to large degrees. Scatter in the observed graph is

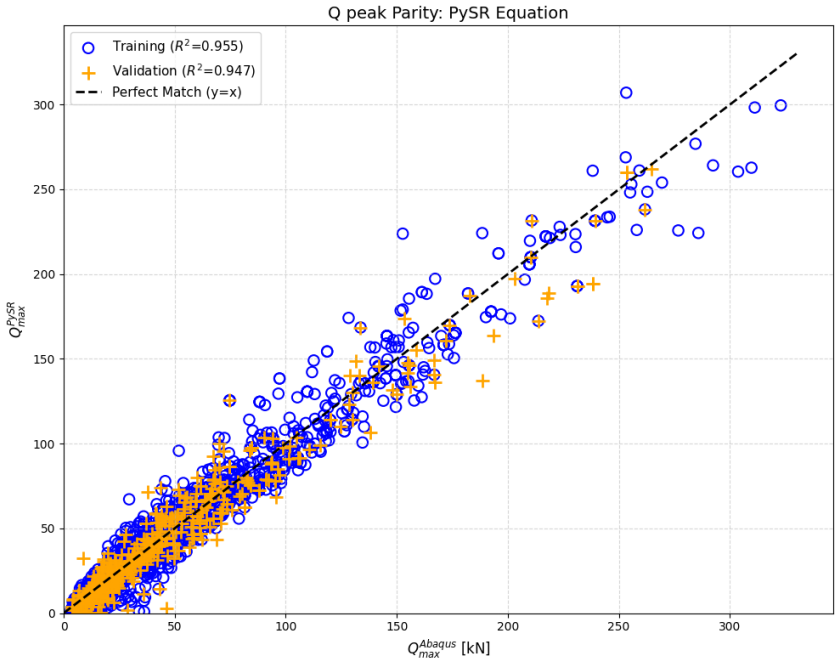


Figure G.1: Parity plot between the ABAQUS results and the developed equation of PySR.

Resulting equation for the peak force capacity is presented in Equation G.1, Of note of this expression is that spring stiffness, K_s, in large parts dictates the behavior of the expression. Therefore loses it accuracy in predictions for simply supported walls especially those of double-wythe.

$$Q_{peak} = 23.8(E_m\sqrt{K_s}(\frac{t}{H})^{4.5})^{0.6} \tag{G.1}$$

This is highlighted in the comparison plot between the experimental data and the developed expression shown in Figure G.2. Walls W2, W4, W5, and W8 which is all simply supported exhibit reduced predictive accuracy, compared to that in Figure G.2.

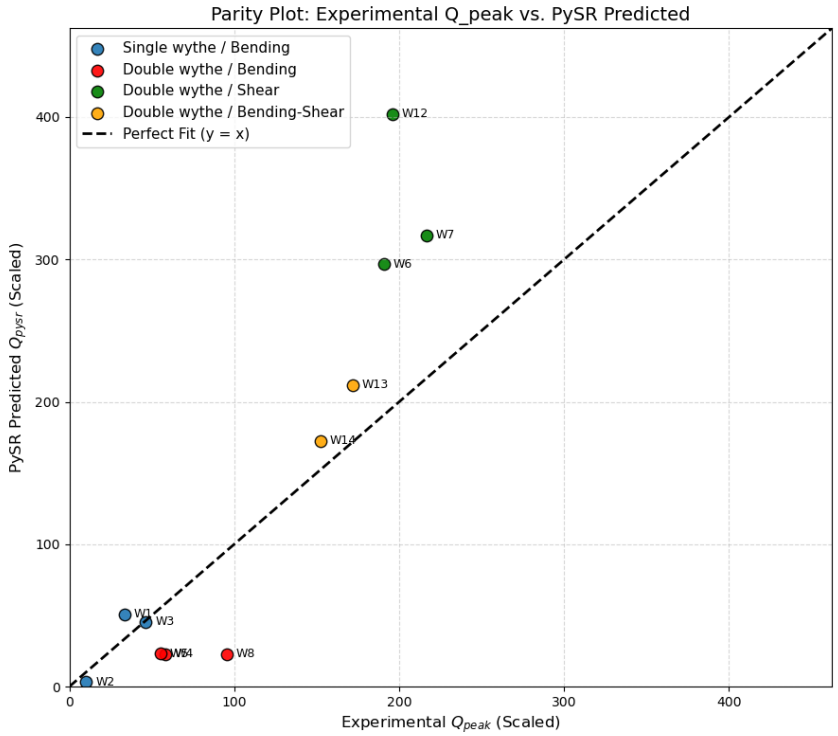


Figure G.2: Comparison of the experimental data and the new expression

As previously mentioned this method loses the accuracy in walls with no spring stiffness therefore an more refined expression is presented in Chapter 5.3.2.

H Appendix 8 — Distributed load compared with four point loading in ABAQUS

To be able to compare the four point bending tests results with that of a wall being subjected to a distributed load a new numerical model was developed in ABAQUS. Instead of applying an velocity to points a pressure was applied over the entire surface as seen in Figure H below.

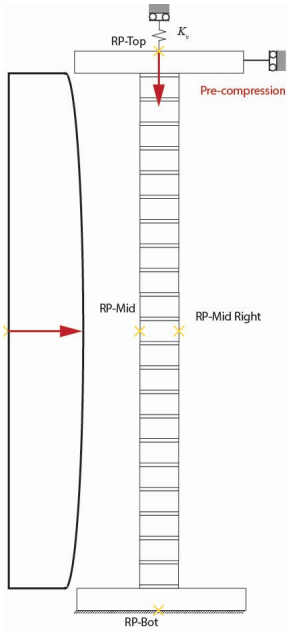
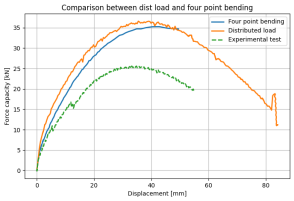


Figure H.1: Distributed load over the single wythe wall

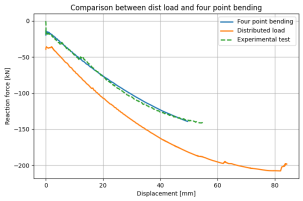
To force the model to not only be pushed of the supports a small deviation was applied on the pressure controlled by the height, where lower pressure was applied closer to the supports, The equation controlling this deviation can be seen in Equation H.1

$$P_d = P(0.9 + 0.1(sin(\frac{\pi y}{H})) \tag{H.1}$$

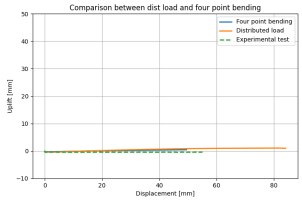
Comparison is made against the four point bending tests both done in ABAQUS and experimental tests. For the wall W1 seen in Figure H.2 the force capacity and uplift see no large deviation between each other, but for the vertical reaction force the distributed force closely resemble the experimental test.



(a) Horizontal reaction force vs horizontal displacement.



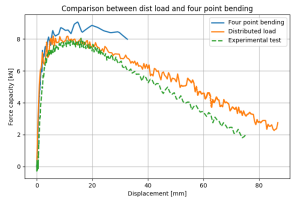
(b) Vertical reaction force vs horizontal displacement



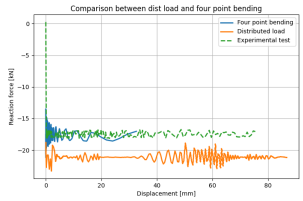
(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure H.2: Comparison between four point bending (orange) with distributed load (blue) and the experimental data (green) for wall W1.

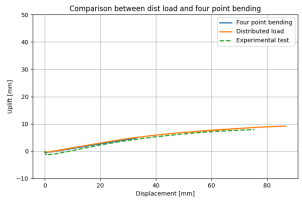
For the simply supported case shown in Figure H.3, the response obtained using distributed loading closely resembles that of the four-point bending configuration across all evaluated results. Although the distributed loading case predicts a marginally higher force capacity, the difference remains limited. Furthermore, thrust development is negligible in both models, which is consistent with the simply supported boundary condition where arching action is not expected to develop.



(a) Horizontal reaction force vs horizontal displacement.



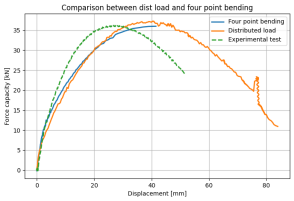
(b) Vertical reaction force vs horizontal displacement



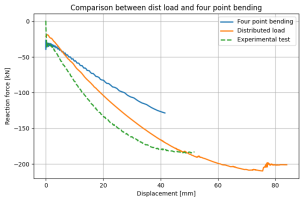
(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure H.3: Comparison between four point bending (orange) with distributed load (blue) and the experimental data (green) for wall W2.

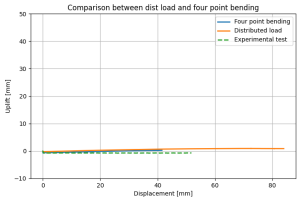
The final wall, W3, shown in Figure H.4, exhibits behavior similar to that of the previously presented walls, with a strong correlation between the four-point bending and distributed loading approaches. Across all three comparisons, both loading methods demonstrate comparable force–displacement behavior and uplift response. In terms of thrust development, however, the four-point bending model shows closer agreement with the experimental results, exhibiting both a more representative response shape and thrust magnitude.



(a) Horizontal reaction force vs horizontal displacement.



(b) Vertical reaction force vs horizontal displacement



(c) Uplift in the top slab vs displacement in the horizontal direction.

Figure H.4: Comparison between four point bending (orange) with distributed load (blue) and the experimental data (green) for wall W3.

Based on the numerical results, four-point bending and distributed loading exhibit comparable bending failure mechanisms, resulting in similar force capacities and, to a large extent, similar displacement responses.

DEPARTMENT OF STRUCTURAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY