IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Bridging Continuum and Discrete Optimization: A Theoretical Framework for Compression-Dominant Masonry Design

[1]Muhamed Mustafa Yousry, [2]Mohamed Ezzeldin, [3]Ayman Assem

[1]Research assistant, [2]Associate Professor of Architecture Engineering, [3]Professor of Architecture Engineering

[1]Department of Architecture,

[1]Faculty of Engineering, Ain shams University, Cairo, Egypt

***Abstract:*** Contemporary masonry design exhibits a fundamental methodological tension between continuum-based optimization techniques and the discrete, modular nature of brick construction. This paper presents a comprehensive theoretical framework that systematizes the integration of parametric geometry generation, structural performance evaluation, density-based topology optimization, and adaptive discretization within a unified conceptual pipeline. Rather than treating topology optimization as a post-hoc analytical tool, the framework positions it as an active design generator embedded within an iterative feedback cycle. The work examines the theoretical foundations of bridging the discrete–continuum gap through recursive parameter coupling, wherein discrete-modular constraints inform continuum optimization objectives, and conversely, topology-driven efficiency targets guide discrete reconfiguration logic. The methodology is organized across two conceptual phases: (1) Morphology Selection, addressing the parametric search for geometrically feasible and structurally competent designs, and (2) Boundary Optimization and Discretization, where continuum material distribution informs discrete assembly rules. This framework addresses a persistent gap in computational masonry literature: the lack of a systematic, generalizable approach to preserve structural performance during the conversion from continuous to discrete representations. The paper formalizes the theoretical logic underlying such integration, elaborates the algorithmic workflow, and discusses implications for future architectural optimization research.

***Keywords:*** Optimization Framework, Topology Optimization, Parametric Masonry, Discrete–Continuum Integration.

## I. Introduction

### 1.1. The Structural Logic of Historical Masonry and Its Digital Displacement

Masonry has historically been inseparable from structural reasoning. The thrust line concept—developed across centuries of architectural practice and formalized through graphical statics—encoded a direct relationship between form, material distribution, and force flow. This reciprocal relationship meant that design iterations naturally accommodated structural logic, with aesthetic and functional concerns arising simultaneously from equilibrium conditions [1].

The computational revolution of the late twentieth century disrupted this synthesis. Digital modeling environments enabled the rapid generation of geometrically complex, organically curved surfaces, decoupled from underlying force paths. NURBS-based parametric systems prioritized formal expressiveness, while structural validation remained a downstream verification step, often resolved through concealed reinforcement rather than inherent geometric logic. This separation—between form-finding and force-finding—has created a persistent methodological challenge: how to re-establish alignment between expressive geometry and structural necessity in computationally mediated design.

### 1.2. The Discrete–Continuum Disconnection

At the core of this challenge lies what may be termed the discrete–continuum disconnection: the incompatibility between the continuous, density-field outputs of topology optimization (TO) and the discrete, modular realities of masonry construction. Standard TO algorithms—particularly the Solid Isotropic Material with Penalization (SIMP) family—operate on continuous design domains and iteratively refine density distributions to minimize strain energy under material volume constraints. The resulting geometries are optimized for stiffness and efficiency at the continuum scale, producing organic, often complex three-dimensional forms.

Masonry, by contrast, is constructed from discrete, standardized units arranged according to explicit bonding rules, joint geometries, and construction sequences. A direct attempt to rationalize continuous TO result into discrete brick modules faces several inherent challenges:

1. **Geometric Incompatibility:** Organic TO boundaries rarely align with regular brick grids. Discretization introduces stair-stepped approximations that can misalign with the optimized load paths.
2. **Stress Concentration at Discrete Boundaries:** The translation from smooth continuum to voxelated brick creates local stress concentrations that may violate the compression-only assumptions underlying masonry design.
3. **Loss of Performance Fidelity:** Structural metrics (safety factors, tensile stress distributions) may degrade significantly during discretization, invalidating the optimization gains predicted in the continuum analysis.
4. **Constructability Constraints:** Beyond structural adequacy, masonry imposes constructability requirements—maximum corbel overhangs, bond pattern continuity, buildable sequences—that standard TO formulations do not explicitly encode.

### 1.3. Research Positioning and Contribution

This paper proposes that the discrete–continuum gap need not be treated as an intractable limitation, but rather as a design problem amenable to systematic methodological treatment. The central hypothesis is that by embedding discretization constraints within the parametric generation phase and creating feedback loops between continuum optimization and discrete reconfiguration, a unified framework can be constructed that simultaneously optimizes for structural efficiency, geometric feasibility, and constructability.

The contribution is fundamentally theoretical and methodological: this paper formalizes the conceptual structure and algorithmic logic of such an integration without presupposing a single optimal discretization strategy. Instead, it elaborates a generalizable framework applicable across varied masonry geometries, loading conditions, and design intents.

## II. Literature Review and Theoretical Background

### 2.1. Historical and Contemporary Approaches to Form-Finding in Masonry.

Form-finding—the practice of deriving structural geometry in equilibrium with applied loads—constitutes the intellectual heritage from which modern computational methods emerge. Historically, techniques such as thrust line analysis and graphical statics allowed designers to verify the stability of arches, vaults, and complex three-dimensional masonry forms through geometric construction and basic statics [1, 2]. These methods were inherently iterative: an initial geometric guess was tested for equilibrium, refined, and re-tested until a stable form emerged. The geometry itself was the primary design variable; its shape encoded the structural solution.

The transition to computational methods began in the mid-twentieth century with the adoption of finite element analysis (FEA). FEA enabled the analysis of complex three-dimensional stress distributions in masonry structures, supporting detailed evaluation of historic structures and complex modern geometries [1]. However, FEA was initially employed primarily as a post-rationalization tool: geometries were designed (often through aesthetic or functional criteria), then analyzed to verify structural adequacy or identify necessary reinforcement.

More recent developments have attempted to re-integrate analysis with design. Block and colleagues [3] introduced Thrust Network Analysis (TNA), a method that extends classical thrust-line concepts through discrete network equilibrium, enabling the generation of three-dimensional compression-only surfaces via linear programming. TNA represents a significant methodological advance by coupling form-finding directly to equilibrium constraints without explicit assumption of a specific geometry.

In parallel, parametric design environments—particularly Grasshopper in Rhinoceros—have enabled the embedding of real-time FEA feedback within the design process. Structural analysis tools such as Karamba3D can be integrated into parametric workflows, allowing designers to evaluate performance of thousands of variants in rapid succession [4]. This represents a qualitative shift: from analysis as verification to analysis as design generator.

## 2.2. Topology Optimization: Theory, Applications, and Limitations.

Topology Topology optimization comprises a family of computational methods that automatically determine optimal material distributions within a design domain to achieve specified objectives (e.g., minimum strain energy, maximum fundamental frequency) subject to constraints (e.g., material volume, displacement bounds, stress limits) [5, 6]. Among these methods, SIMP—Solid Isotropic Material with Penalization—has achieved widespread adoption in both engineering and architectural contexts due to its mathematical rigor, computational efficiency, and intuitive material interpretation [6].

The SIMP methodology iteratively refines a density field ρ across the design domain, penalizing intermediate density values to drive the solution toward binary (solid or void) configurations. The approach minimizes strain energy (compliance) subject to a material volume constraint:

$$\text{Minimize: } C(\rho) = U^T K(\rho) U$$
$$\text{Subject to: } V(\rho) \leq V_{target},\ 0 < \rho \leq 1$$

where U is the displacement vector, K(ρ) is the density-dependent stiffness matrix, and V(ρ) is the current material volume.

In architectural applications, TO has been adopted as a tool for generating structurally efficient, material-conscious designs [7, 8]. The method naturally identifies load paths and produces geometries that concentrate material where structural demand is highest, creating forms that are ostensibly "efficient" and "structurally expressive."

However, TO faces a persistent interpretation barrier in architectural practice: the output is a continuous density field or smooth, organic geometry—not a constructible object. The result provides no information about material units, fabrication constraints, assembly sequences, or modular logic. For masonry in particular, this gap is profound. Standard TO produces smooth, often doubly curved surfaces; masonry consists of discrete rectangular units bonded in specific patterns. Translating one to the other without loss of structural fidelity remains an open methodological problem.

## 2.3. Parametric Design and Digital Masonry.

Parametric design tools have become central to the generation and management of complex masonry systems. These tools encode design rules—relationships between geometric parameters, brick dimensions, bonding logic, and construction constraints—enabling dynamic adaptation of form while maintaining constructive validity [4, 9].

Grasshopper, through its hierarchical data tree structure and component-based workflow, proves particularly well-suited to masonry applications. Rules for bond patterns, geometric constraints (collision detection, dimensional feasibility), and material assignments can be encoded in parametric scripts, enabling exploration of design variants without manual re-detailing. Recent research demonstrates the use of such tools for façade retrofit, complex brick pattern generation, and structural performance optimization [10].

However, most existing parametric masonry workflows exhibit a fundamental one-directional flow: geometry is generated parametrically, then evaluated structurally, with structural insights remaining external to the parametric generation logic itself. In other words, performance feedback is not systematically incorporated back into the parametric rules, limiting the potential for integrated optimization.

### 2.4. Structural Analysis of Masonry Under Compression-Only Constraints.

Masonry structural theory rests on the assumption that masonry cannot sustain significant tensile stress without cracking. This leads to Heyman's no-tension model [2], wherein stability is ensured if compressive stress remains within the kern (middle third of the section) and tensile stress is negligible. In contemporary computational terms, designs are screened for compression-only behavior by evaluating the Percentage of Elements in Tension (PET)—the proportion of the discretized domain exhibiting positive principal stress. Linear elastic analysis (LEA), while mathematically conservative (it assumes infinite tensile strength), serves as an efficient proxy for form-finding validation [2, 11]. Designs with low PET values are presumed to remain stable; high PET values warrant refinement or rejection. The distinction between LEA-based form-finding and nonlinear structural verification is crucial: LEA screens candidate forms efficiently; rigorous validation requires more sophisticated analyses.
This framework has enabled the coupling of FEA with parametric design, allowing rapid iteration toward compression-dominant geometries [4].

### 2.5. Integration of Computational Design and Optimization: State of the Practice.

The contemporary frontier in computational architecture lies at the convergence of three domains:

1. **Parametric geometry generation** (rule-based, data-driven, adaptive)
2. **Structural analysis and optimization** (FEA, topology optimization, evolutionary algorithms)
3. **Fabrication and construction logic** (discrete units, bonding patterns, assembly sequences)

Existing practice tends to address these domains sequentially: parametric models generate geometry, optimization techniques refine it, and then fabrication constraints are accommodated post-hoc. This linear flow inevitably results in compromises, as each downstream phase negotiates with the outputs of the previous one.
Recent research by Preisinger and colleagues [4, 12] has advanced integrated workflows where embedded FEA provides real-time feedback to parametric generation. Similarly, research on fabrication-aware design [13] and machine-learning-enabled variant exploration [14] points toward systems where structural, geometric, and constructive constraints are simultaneously active during design synthesis.

The present research contributes to this trajectory by formalizing a fully integrated, cyclical framework where parametric generation, structural optimization, topology optimization, and adaptive discretization form a closed loop, with information flowing bidirectionally between phases.

## III. Theoretical Framework: A Two-Phase Integrated Optimization Paradigm y

### 3.1. Conceptual Architecture and Phase Organization

The proposed framework is structured as a two-phase system, each decomposed into sequential operational steps. The overarching logic is cyclical: neither phase is truly "final"; rather, they represent levels of refinement that can be revisited as design intent or constraints evolve, as illustrated in figure 1.

**Phase One: Morphology Selection and Structural Baseline**
- **Step 1:** Parametric generation of discrete brick geometries according to explicit tectonic rules
- **Step 2:** Evolutionary optimization to identify structurally competent baseline morphologies

**Phase Two: Boundary Optimization and Discrete Rationalization**
- **Step 3:** Continuum topology optimization within the Phase One baseline domain
- **Step 4:** Adaptive discretization and performance validation

The logical flow is:
Parametric Rules → Variant Generation → Performance Evaluation → Baseline Selection → Continuum Optimization → Discrete Reinterpretation → Validation Feedback
This structure acknowledges that optimization in a discrete domain (Phase One) must precede continuum optimization (Phase Two), because the continuum design domain itself must be structurally reasonable. Otherwise, TO operates on an ill-posed problem with unconstrained freedom for material removal.

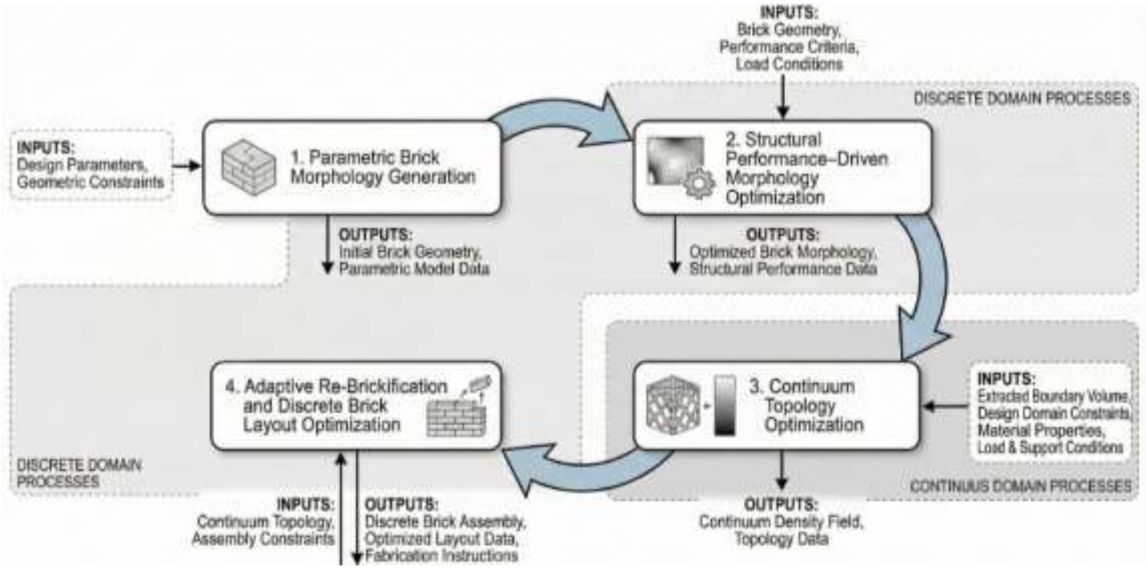

Fig. 1: Integrated computational framework illustrating the inputs & outputs of each step, source: created by the author.

The framework uses a two-phase structure to address optimization at complementary scales. **Phase One** explores geometrically valid, structurally sound configurations in the discrete design domain using parametric rules, while **Phase Two** refines the Phase One baseline in the continuum domain through topology optimization. The table below contrasts their roles, methods, and contributions within the integrated framework.

**Table 1: PHASE ONE VS. PHASE TWO: KEY CHARACTERISTICS AND OUTPUTS**

| Aspect | Phase One: Morphology Selection | Phase Two: Boundary Optimization & Discretization |
|---|---|---|
| **Design Domain** | Discrete parametric space; brick modules are primary design variables | Continuous volumetric domain bounded by Phase One baseline geometry |
| **Optimization Objective** | Identify structurally competent baseline morphologies balancing compression-only behavior (PET), safety (SF), and geometric feasibility | Minimize strain energy (compliance) subject to material volume constraint; identify optimal load paths within baseline domain |
| **Optimization Method** | Genetic Algorithm (GA); population-based, multi-objective search | SIMP (Solid Isotropic Material with Penalization); density-field refinement via iterative FEA |
| **Key Outputs** | 1. Parametric rule library describing brick placement 2. Population of design variants 3. Selected baseline morphology (structurally validated) | 1. Density field $\rho(x,y,z)$ over volumetric domain 2. Extracted, smoothed boundary 3. Material distribution map indicating high/low demand regions |
| **Performance Metrics Evaluated** | PET (Percentage of Elements in Tension), Safety Factor (SF), geometric feasibility constraints, bond continuity, dimensional fit, collision avoidance | Compliance $C(\rho)$, PET, SF, material volume $V(\rho)$, convergence criteria |
| **Design Degrees of Freedom** | Curvature (R), wall height (H), number of courses (HS), brick orientation angle (BOA), bond type (BT), offset ratio (OR) | Density at each voxel element $\rho_e \in [0, 1]$; no explicit control over geometric form (emergent from optimization) |
| **Computational Cost** | ~1,500 parametric variants evaluated; minutes to hours (design iteration feasible) | 100+ iterations of FEA-driven optimization; hours (overnight studies feasible) |

| Aspect | Phase One: Morphology Selection | Phase Two: Boundary Optimization & Discretization |
|---|---|---|
| **Output Format** | Discrete brick assemblies (3D coordinates of each brick); buildable, modular geometry | Continuous density field; requires post-processing (boundary extraction, smoothing) and re-discretization |
| **Constructability Assurance** | Built-in through parametric rules (bond continuity, dimensional fit, corbelling limits enforced algorithmically) | Implicit in base domain; explicit verification required during re-brickification step (Phase Two, Step 4) |
| **Design Intent Integration** | High: designer controls parametric ranges, fitness function weights, and baseline selection | Medium: designer sets volume target V_target and penalization strategy; exact topology emergent from optimization |
| **Iteration Capability** | Can return to Phase One after Phase Two validation if structural fidelity lost; supports iterative refinement | Can loop within Phase Two (adjust V_target, retarget optimization) or feed insights back to Phase One parametric adaptation |

## 3.2 Phase One: Morphology Selection Through Evolutionary Optimization

### 3.2.1 Parametric Rule Definition and Geometric Intent

The framework starts by formalizing tectonic rules—algorithmic relations that convert design intent (curvature, height, bonding pattern) into detailed brick configurations. This is essential because parametric models bridge continuous design geometry and discrete construction logic based on modular bonding rules, as illustrated in figure 2. The parametric model encodes:

1. **Primary geometric parameters:** curvature radius, wall height, number of brick courses, brick dimensions
2. **Secondary parameters:** brick orientation angle, bond type (running, Flemish, stack), offset patterns for staggered joints
3. **Constraint rules:** maximum deviation from ideal brick positioning (accounting for manufacturing tolerance), collision detection between adjacent units, bond continuity checks

The data structure for managing these rules is crucial. A hierarchical data tree—where each branch represents a course and each item represents an individual brick—preserves the logical sequence required for both geometric validity and construction sequence planning.

### 3.2.2 Geometric Feasibility Constraints

Not all parametric combinations yield geometrically valid assemblies. The framework must explicitly enclose feasibility constraints:

- **Bond continuity:** Vertical joints in adjacent courses must be staggered to avoid stress concentration at aligned joints
- **Dimensional fit:** Each brick must occupy its designated position within manufacturing tolerance (typically ±3 mm per masonry standards)
- **Collision avoidance:** On high-curvature surfaces, adjacent bricks may intersect at inner corners if the radius of curvature is too tight relative to brick dimensions
- **Corbelling limits:** Overhanging courses must remain within traditional masonry practice constraints to ensure constructability

These constraints are encoded as Boolean penalties in the parametric model: a design that violates any constraint receives a high penalty value in subsequent fitness evaluation.

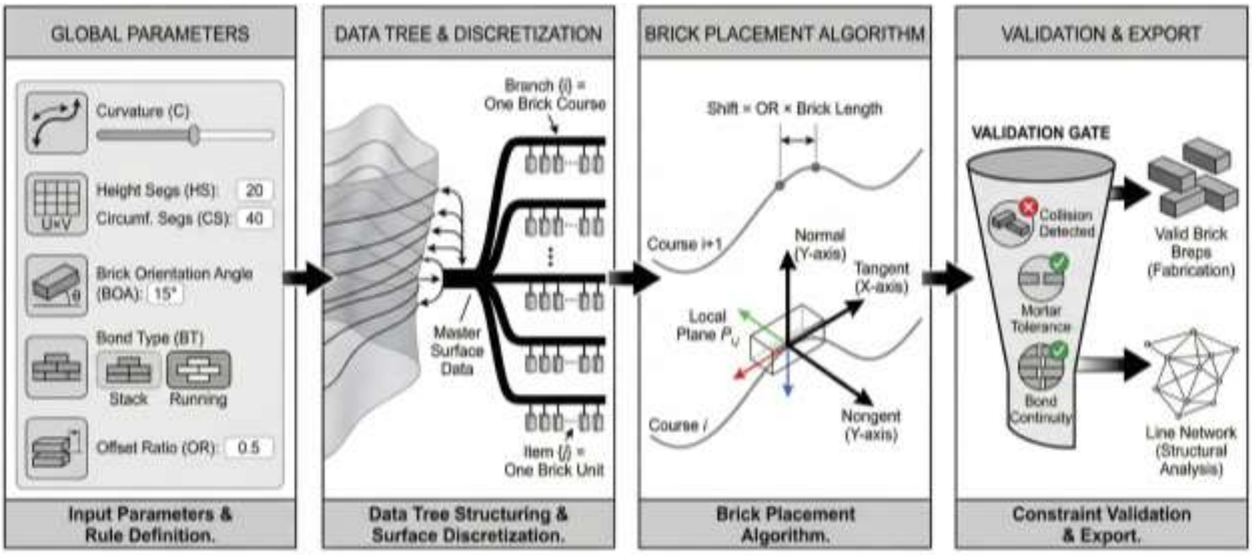


Fig. 2: Parametric brick wall generation workflow from input parameters through geometric generation and constraint solving to the final assembly, source: created by the author

### 3.2.3 Performance-Driven Evolutionary Search

With parametric rules and constraints in place, the framework employs evolutionary algorithms—specifically genetic algorithms (GAs)—to explore the design space. Rather than attempting to solve an optimization problem analytically, GAs maintain a population of candidate designs, evaluate each against a fitness function, and iteratively breed high-fitness designs while introducing controlled mutations to maintain exploration.
The fitness function combines multiple objectives:

$$\mathbf{F = w_1 \cdot PET\_normalized + w_2 \cdot SF\_inverse + w_3 \cdot Constraint\_penalties}$$

where:
- **PET_normalized** quantifies tension (lower is better for compression-only masonry)
- **SF_inverse** penalizes low safety factors (lower is better)
- **Constraint_penalties** enforce geometric and constructive feasibility

The weighting factors ($w_1$, $w_2$, $w_3$) reflect design philosophy. By emphasizing PET (tension minimization) with $w_1 = 0.4$, the framework prioritizes the fundamental masonry principle of compression-only behavior; safety factor receives $w_2 = 0.3$, and geometric constraints $w_3 = 0.3$.

**Rationale for Evolutionary Search Over Gradient-Based Optimization:**
Evolutionary algorithms are preferred over gradient-based methods in this context for several reasons:
1. **Discontinuous constraint space:** The feasibility constraints are Boolean and discontinuous; gradient-based methods struggle with such search spaces
2. **Multiple competing objectives:** The fitness function aggregates distinct metrics with incommensurable units; evolutionary algorithms naturally accommodate multi-objective optimization
3. **Discrete parameter exploration:** Many parameters (bond type, number of courses) are discrete; evolutionary algorithms handle discrete and continuous variables simultaneously
4. **Exploration vs. exploitation balance:** GAs explicitly balance population diversity (exploration) with convergence (exploitation), reducing the risk of premature convergence to local optima

### 3.2.4 Structural Evaluation via Linear Elastic Analysis

Each candidate design is structurally evaluated using linear elastic analysis (LEA) embedded within the parametric environment. LEA provides fast, deterministic stress distributions, enabling high-throughput variant evaluation.
The parametric geometry is automatically discretized into a tetrahedral mesh. Material properties are homogenized (masonry treated as a single, isotropic material), a necessary simplification for computational tractability in iterative optimization. Boundary conditions are standard: base fixed, top free, representing a cantilever wall condition.
Loading includes self-weight (automatically computed from geometry and material density) and lateral load (e.g., 5 kPa wind pressure or equivalent seismic acceleration).

The output of LEA is a nodal stress field from which three key metrics are extracted:

1. **Percentage of Elements in Tension (PET):** The proportion of mesh elements with positive principal stress, used as the primary compression-only screening metric
2. **Maximum Principal Stress ($\sigma_1$,max):** Used to compute global safety factor as f_c / $\sigma_1$,max
3. **Maximum Displacement:** Checked against serviceability limits

These metrics directly populate the fitness function, creating a tight coupling between parametric form and structural performance.

### 3.2.5 Baseline Selection and Transition to Phase Two

The evolutionary search produces a population of structurally viable variants. Multiple selection strategies are possible:

- **Single optimum:** Select the design with lowest fitness value (Pareto dominance)
- **Pareto frontier:** Identify non-dominated designs that represent different efficiency–robustness trade-offs
- **Intermediate selection:** Choose a design with modest optimization—enough to demonstrate improvement, but with conservative margins for subsequent refinement

The selected baseline serves as the design domain for Phase Two continuum optimization. Critically, this baseline is already structurally competent (it meets PET < 5%, SF > 2.5, and other constraints). Subsequent TO operates within a validated domain, rather than attempting to impose optimization on an under-specified or infeasible starting geometry.

## 3.3 Phase Two: Boundary Optimization and Discrete Reinterpretation

### 3.3.1 Transitioning from Discrete to Continuum: Domain Definition

The Phase One baseline geometry defines a volumetric design domain for Phase Two continuum optimization. This domain is characterized by:

- **Boundary:** The outer surface of the Phase One optimized wall
- **Interior:** A voxelated representation with uniform element size (e.g., 0.05 m)
- **Support conditions:** Same boundary conditions as Phase One (base fixed, top free)
- **Loads:** Identical static loads (self-weight + lateral pressure)

This volumetric domain represents all the "material" available for optimization. The TO algorithm will selectively remove (or rather, set to low density) regions where structural demand is low, concentrating material where stress is high.

### 3.3.2 Continuum Topology Optimization via SIMP

The continuum optimization problem is formulated classically:

**Minimize:** $C(\rho) = U^T K(\rho) U$

**Subject to:**

$$K(\rho) U = F \text{ (equilibrium constraint)}$$
$$V(\rho) \leq V_{target} \text{ (volume constraint)}$$
$$0 < \rho_e \leq 1 \text{ (density bounds)}$$

The SIMP penalization scheme is applied:

$$k_e = \rho_e^p \cdot \bar{k}_e, \text{ where } p \in [2, 4]$$

Penalization exponent p = 3 is standard; this value forces the optimizer to drive intermediate densities toward binary (0 or 1) rather than settling on a "grey" gradient. Physically, this reflects the reality of masonry: bricks are solid or not; partial density "quasi-brick" elements are not constructible.

The volume constraint $V(\rho) \leq V_{target}$ is set to target approximately 20% material reduction ($V_{target} = 0.80 V_0$). This value balances efficiency with constructability:

- **V < 0.75:** Results in highly fragmented, disconnected voids that cannot be rationalized into buildable brick patterns
- **V ≈ 0.80:** Produces distinct, architecturally interpretable voids while achieving meaningful material savings

- **V > 0.85:** Yields minimal material efficiency gains relative to additional computational cost

The optimizer iteratively refines the density field over 100+ iterations until convergence (change in objective < 0.1% over 5 consecutive iterations).

#### 3.3.3 Boundary Extraction and Smoothing

The output of SIMP is a density field—a voxelated representation where each element has a density value between 0 and 1. To convert this to an architectural geometry, an isosurface is extracted at a threshold density $\rho_{th} = 0.50$, capturing the transition between high-density (likely to be solid in the final design) and low-density (likely to be void) regions.

The extracted isosurface is inherently jagged due to the voxelated representation. Laplacian smoothing is applied iteratively to remove these artifacts while preserving topological features (voids, protrusions, buttressing). This produces a smooth, continuous boundary that can be used to regenerate discrete brick assemblies in Step 4.

**Theoretical Significance of Boundary Extraction:**

The boundary extraction step is not merely post-processing; it represents a conscious re-interpretation of the continuum optimization result. The isosurface at $\rho_{th} = 0.50$ represents a design decision: material with density > 0.50 is assumed solid, density < 0.50 is assumed void. This threshold is somewhat arbitrary but conventionally justified as representing the "most likely" material distribution. Alternative thresholds (e.g., 0.30, 0.70) would produce different interpretations of the same density field, highlighting that topology optimization provides information about material efficiency, not a unique "optimal" shape.

#### 3.3.4 Theoretical Basis for Adaptive Discretization

The transition from continuum optimization to discrete brick assembly cannot be mechanical; it requires adaptive logic that interprets the optimized geometry through the lens of discrete constructive rules. The framework formalizes this adaptation as a rule-based reinterpretation process:

**Given:**

- An optimized continuum boundary B_opt (from boundary extraction)
- Discrete unit dimensions (brick length, height, depth)
- Bonding rules (running, Flemish, stack)
- Constructability constraints (maximum corbel overhang, etc.)

**Find:**

- A discrete brick assembly A_disc that:
  - Approximates the boundary B_opt (geometric fidelity)
  - Maintains structural performance (PET, SF, stress distribution)
  - Respects constructability constraints
  - Preserves masonry tectonic logic

The challenge is that these objectives may conflict. Perfect geometric fidelity to B_opt might require excessive brick cutting or corbelling; strict adherence to constructability limits might require relaxing geometric fidelity. Rather than prescribing a single resolution, the framework formalizes the discretization as an algorithmic procedure guided by multiple heuristics, with iterative validation and optional refinement.

#### 3.3.5 Adaptive Re-Brickification: Algorithmic Logic

The re-brickification process operationalizes discretization through the following logic:

**Step 1: Geometric Analysis of Optimized Boundary** Analyze the B_opt boundary to extract:

- Height variation along the wall
- Circumferential arc length at each height level
- Identify void regions and their extent

**Step 2: Parametric Rule Adaptation** Adapt the parametric generation rules from Phase One to the new boundary:

- Compute number of bricks per course (circumferential segments) based on local arc length
- Adjust height segments if necessary to maintain consistent course height
- Identify regions where bricks must be cut or rotated to approximate the optimized boundary

**Step 3: Discrete Brick Generation** Regenerate the brick assembly on the optimized boundary using the adapted parametric rules, producing a discrete assembly A_disc as shown in figure 3.

**Step 4: Iterative Validation and Refinement** Validate A_disc against structural and constructability criteria:
- Perform FEA on A_disc to compute PET, SF, stresses
- Compare metrics to the continuum baseline; if discretization error exceeds thresholds (e.g., $\Delta PE > 2\%$, $\Delta SF > 5\%$), adjust parameters and regenerate
- Continue until convergence

This process is not deterministic; different parameter choices in Steps 2–3 yield different discrete assemblies, all of which may be structurally valid and constructable. Rather than claiming a single "optimal" discretization, the framework acknowledges multiple valid solutions and supports designer choice based on aesthetic or functional preferences.

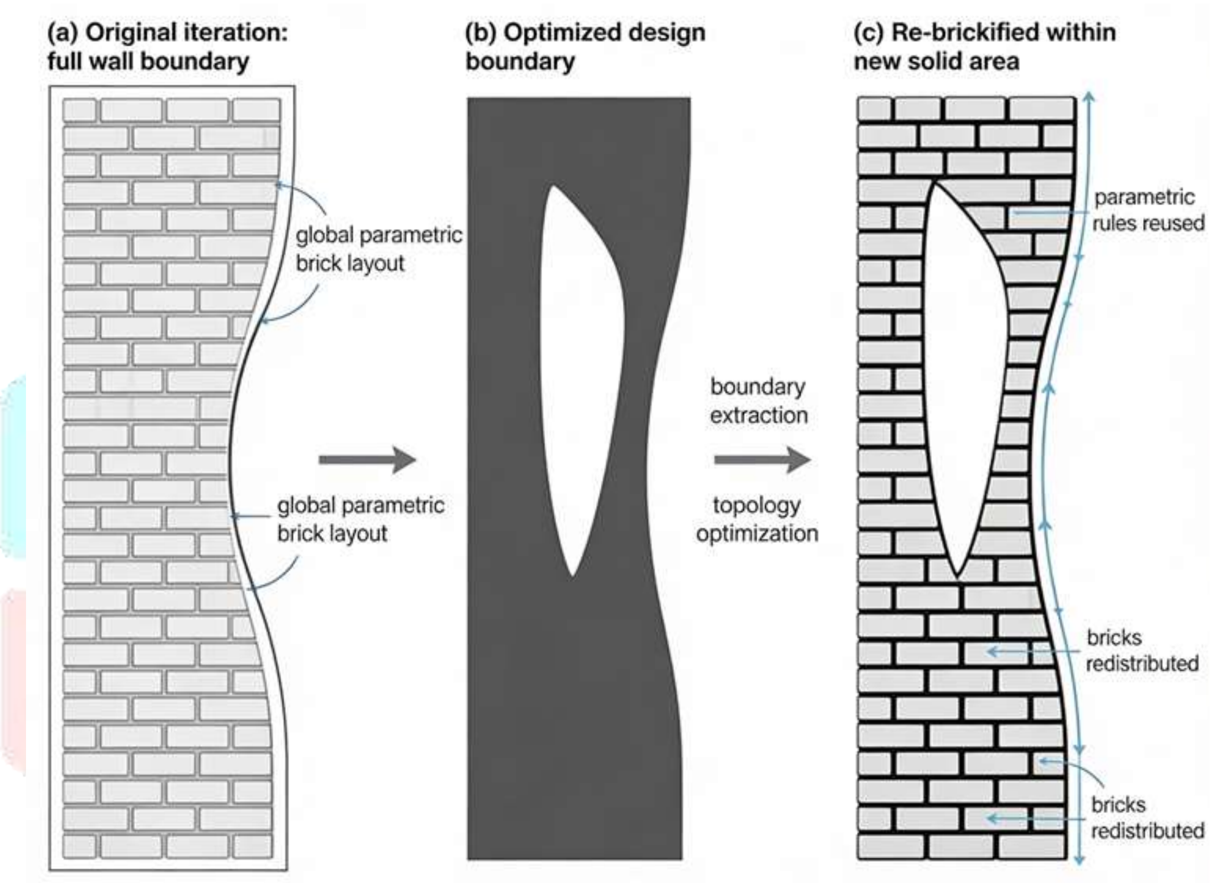


Fig. 3: Conceptual Re-brickification workflow, source: created by the author

### 3.4 Integration Logic and Feedback Mechanisms

The framework's power derives from its cyclical, feedback-driven architecture. Information flows bidirectionally:

1. **Phase One → Phase Two:** The baseline morphology from Phase One defines the design domain and boundary constraints for Phase Two continuum optimization

2. **Phase Two → Discretization:** The optimized continuum topology informs the re-brickification rules, suggesting which regions of the design have low structural demand (candidate for discretization refinement)

3. **Discretization → Validation:** The discrete assembly is validated against performance targets; if structural fidelity is lost, feedback is provided to adjust parametric parameters or refine the optimization

4. **Validation → Iteration:** Results from validation may motivate a return to Phase One to explore alternative baseline morphologies

This cyclical structure addresses a fundamental limitation of sequential optimization: by creating feedback loops, the framework allows downstream phases to inform upstream decisions, reducing the cumulative loss of performance fidelity across the pipeline.

The framework's theoretical formulation rests upon a series of intentional assumptions and design choices, each addressing distinct aspects of the discrete–continuum integration problem. The following table summarizes these core assumptions, their scientific justification, and their implications for framework applicability. Understanding these assumptions is essential for applying the framework to novel contexts or extending it toward different material systems or loading conditions.

**Table 2: FRAMEWORK ASSUMPTIONS AND JUSTIFICATIONS**

| Framework Assumption | Scientific Justification | Applicability & Scope | When to Reconsider |
|---|---|---|---|
| **Linear Elastic Analysis (LEA) as form-finding proxy** | LEA is mathematically conservative (assumes infinite tensile strength), providing safe screening for compression-only behavior [2]. Global stress distributions match micro-scale models within ~2% error [15]. | Appropriate for conceptual form-finding and variant screening in masonry under static loads. Not suitable as final design verification. | When nonlinear effects (cracking, rocking, sliding), time-dependent behavior (creep), or dynamic loads are significant. |
| **Homogenized material properties (brick + mortar as single isotropic material)** | Computational tractability for iterative optimization. Extensive research validates homogenized approach for global stress distribution assessment [15]. | Valid for form-finding and topology-driven design. Sufficient fidelity for structural logic exploration. | For final design verification, micro-scale brick–mortar interface analysis required. For anisotropic materials or heterogeneous systems. |
| **Static self-weight + lateral pressure loading only** | Simplifies initial framework development; covers primary masonry loading scenarios (wind, basic seismic equivalent). | Appropriate for initial exploration and design research. Establishes baseline structural logic. | When seismic demands are critical, dynamic response matters, or time-dependent effects (creep, shrinkage thermal cycling) are dominant. |
| **Compression-only design as primary organizing principle ($w_1$ = 0.4 in fitness function)** | Fundamental to masonry theory [2]. PET-based screening efficiently identifies tension-dominant regions. | Ensures forms respect masonry's no-tension behavior. Aligns with traditional practice and safety principles. | For reinforced or post-tensioned masonry where limited tension is acceptable. For hybrid material systems. |
| **SIMP topology optimization with p=3 penalization and V_target = 0.80 $V_0$** | Standard penalization drives intermediate densities toward binary (0 or 1) [5, 6], reflecting masonry's discrete nature. V = 0.80 balances efficiency (meaningful material savings) with constructability (interpretable void patterns). | Produces architecturally meaningful topologies with ~20% material reduction. Avoids over-fragmented, unbuildable patterns. | For extreme efficiency demands ($V < 0.70$), robustness-critical applications ($V > 0.85$), or multi-material systems. |

| Framework Assumption | Scientific Justification | Applicability & Scope | When to Reconsider |
|---|---|---|---|
| **Isosurface extraction at ρ_th = 0.50 for boundary definition** | Threshold = 0.50 conventionally represents the "most likely" material distribution separating solid from void [6]. | Standard practice in TO applications. Produces reasonable geometric interpretation. | Alternative thresholds (0.30, 0.70) yield different discretizations of same density field. Choice should align with design intent. |
| **Parametric brick generation with discrete, full-sized units (no cutting)** | Reflects traditional masonry constructability constraints [16]. Preserves tectonic logic and simplifies fabrication. | Ensures designs are buildable with standard brick modules. Maintains vernacular aesthetic integrity. | For 3D-printed or robotically assembled masonry, cutting/placement tolerance increases. Rules may adapt. |
| **Phase One evolutionary search before Phase Two continuum optimization** | Establishes structurally reasonable baseline domain. Prevents TO from operating on ill-posed problem with unconstrained material removal freedom. | Ensures Phase Two optimization inherits valid starting geometry. Reduces final discretization error. | If starting with hand-designed or external geometry, Phase One can be bypassed. Trade-off: loss of integration benefits. |

## IV. Methodological Considerations and Implementation

### 4.1 Computational Efficiency and Tractability

A key practical concern is computational cost. The framework involves multiple optimization phases, each computationally intensive:

- Phase One evolutionary search: ~1,500 parametric variants, each requiring FEA evaluation (minutes of cumulative runtime)
- Phase Two SIMP optimization: 100+ iterations of density-field refinement (hours of cumulative runtime)
- Discretization and validation: Iterative FEA of discrete assemblies (hours cumulative)

Total runtime on contemporary hardware is typically 3–5 hours for a complete exploration. This is practically feasible for design iteration (overnight studies, multi-day parametric explorations), supporting the role of the framework as a design research tool rather than real-time design assistant.

Computational efficiency could be substantially enhanced through:

- **Machine learning acceleration:** Train neural networks to predict performance metrics without explicit FEA
- **Surrogate modeling:** Build reduced-order models capturing the relationship between parametric variables and performance
- **Parallelization:** Distribute variant evaluation across multiple processors
- **GPU acceleration:** Implement FEA solvers on graphics processors for 10–100× speedup

### 4.2 Homogenized vs. Micro-Scale Material Modeling

The framework employs homogenized material properties for masonry (treating brick + mortar as a single isotropic material with average properties). This simplification is necessary for computational tractability but introduces approximations:

- **Brick–mortar interface effects** (which may initiate cracking at lower stress than predicted by homogenized analysis)
- **Anisotropy** (actual masonry exhibits directionally dependent behavior depending on bond pattern)
- **Joint failure modes** (sliding along mortar joints is not explicitly modeled)

Homogenized modeling is justified by extensive prior research [15] showing that global stress distributions are captured within ~2% error relative to micro-scale models that explicitly represent brick and mortar layers.

For form-finding (the intended use), this fidelity is sufficient; for final design verification, micro-scale analysis would be required.

### 4.3 Linear Elastic Analysis as Form-Finding Proxy

The framework uses LEA for structural screening and topology optimization, rather than nonlinear analysis accounting for masonry cracking, rocking, or sliding. LEA is mathematically conservative (assumes infinite tensile strength) and thus functions as a screening tool: designs that pass LEA compression-only and safety factor criteria are presumed sound, but detailed verification requires nonlinear analysis.

This distinction—between form-finding screening and rigorous verification—must be maintained throughout application of the framework. The framework generates candidate forms; final structural validation is external.

### 4.4 Design Intent vs. Algorithmic Determinism

A philosophical question underlying the framework is the role of designer intent versus algorithmic determinism. The framework can be applied in two modes:

1. **Exploratory mode:** Designer specifies parametric ranges and design priorities (weights in the fitness function), then allows the algorithm to explore and report on promising designs without prescribing a final answer
2. **Prescriptive mode:** Designer specifies target performance metrics (e.g., "minimize material use subject to PET < 1%") and tasks the algorithm with finding designs meeting those criteria

Both models are valid; the framework supports both. However, the framework is not deterministic—multiple runs with identical parameters may yield slightly different results due to stochastic elements in genetic algorithms. This stochasticity is not a bug but a feature: it reveals the breadth of the design space and prevents overfitting a single "optimal" solution.

## V. Theoretical Implications and Design Philosophy

### 5.1 Reintegrating Form and Force in Digital Architecture

The framework addresses a fundamental question in contemporary architectural practice: can computational form-generation be meaningfully coupled to structural logic, rather than treated as an aesthetic concern resolved through post-hoc reinforcement?

Historical masonry design answered this question affirmatively—form and force were inseparable. The framework proposes that digital design can recover this integration by embedding structural evaluation and optimization within the parametric generation process itself. Rather than generating form geometrically and then evaluating structure, the framework allows structure to actively participate in form generation.
This represents methodological reorientation: topology optimization is not treated as a specialized analysis tool for aerospace or mechanical engineering, but as a core architectural design instrument capable of producing expressive, tectonic geometries that embody structural logic.

Figure 4 illustrates the evolution of design methodology in masonry. (Left) Historical form-finding: structural logic and aesthetic expression emerged simultaneously through equilibrium. (Center) Contemporary digital workflows: form generation (geometric modeling) and structural verification (FEA) remain separate, leading to post-hoc reinforcement. (Right) Proposed integrated framework: computational design re-establishes direct coupling between form generation and structural optimization through cyclical feedback loops, recovering the historical synthesis of form and force."

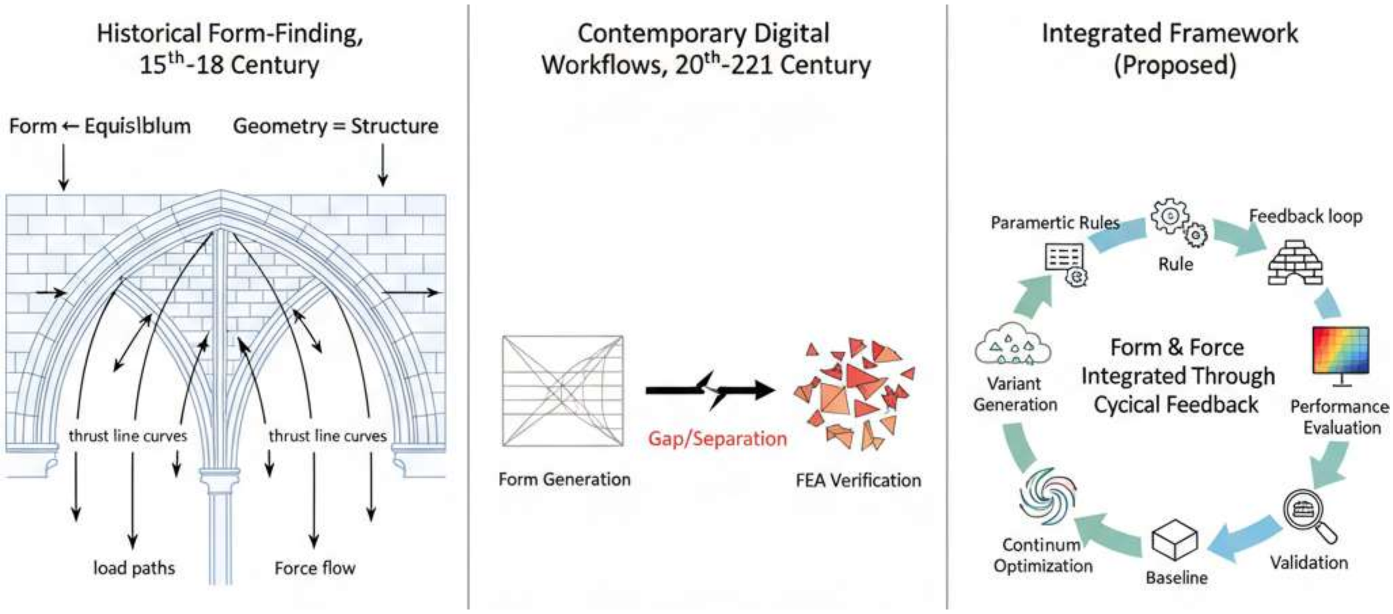


Fig. 4: Historical vs. Digital Form-Finding: Recovering Integration Through Computational Feedback, source: created by the author

### 5.2 Discrete Modularity as Design Constraint, Not Limitation

Contemporary optimization literature often treats discrete constraints as obstacles to overcoming. The framework proposes an alternative perspective: discrete modularity (the fact that masonry is built from discrete bricks) is not a constraint to be minimized but a constitutive feature of the tectonic language to be celebrated and expressed.

Rather than attempting to force continuous optimization results into brick grids, the framework works with the grain of discrete construction. Parametric brick generation is a primary design medium in Phase One; discrete re-brickification in Phase Two respects and amplifies discrete logic rather than fighting it.
This philosophical stance has practical implications: it reduces the tension between optimization and constructability, positioning them as aligned rather than opposed objectives.

### 5.3 Multiple Optima and Design Freedom

Standard optimization seeks a single optimum—the configuration that minimizes (or maximizes) an objective function. The framework acknowledges that in architectural design, multiple valid solutions often exist, representing different resolutions of competing criteria (structural efficiency, aesthetic expression, constructability).
The framework supports designer exploration of the space of valid solutions, rather than prescribing a unique "optimal" design. This approach aligns with contemporary understandings of design as a value-laden, culturally situated practice rather than a pure technical optimization.

### 5.4 Compression-Only Design as Organizing Principle

Throughout the framework, compression-only behavior (quantified via PET) serves as a primary organizing principle. This principle is not novel—it is fundamental to masonry theory—but the framework's systematic embedding of PET constraints throughout multiple optimization phases represents a deliberate choice to center this principle in computational design practice.
By making PET a primary fitness criterion in Phase One ($w_1 = 0.4$) and a validation target in Phase Two, the framework effectively says: "We are designing for compression-only masonry, and all other optimization objectives are secondary to this fundamental structural requirement."
This principle-driven approach provides philosophical clarity and supports the generation of forms that are genuinely compression-dominant rather than merely satisfying post-hoc compression criteria.

Figure 5 illustrates the hierarchical structure of framework objectives. Compression-only behavior (quantified by Percentage of Elements in Tension, PET) serves as the primary organizing principle, with a target PET < 5%. Secondary objectives (structural efficiency, geometric feasibility, constructability) are optimized subject to this primary constraint. This principle-driven hierarchy ensures that all resulting designs respect masonry's fundamental material characteristic—inability to sustain significant tensile stress—before pursuing efficiency or aesthetic goals."

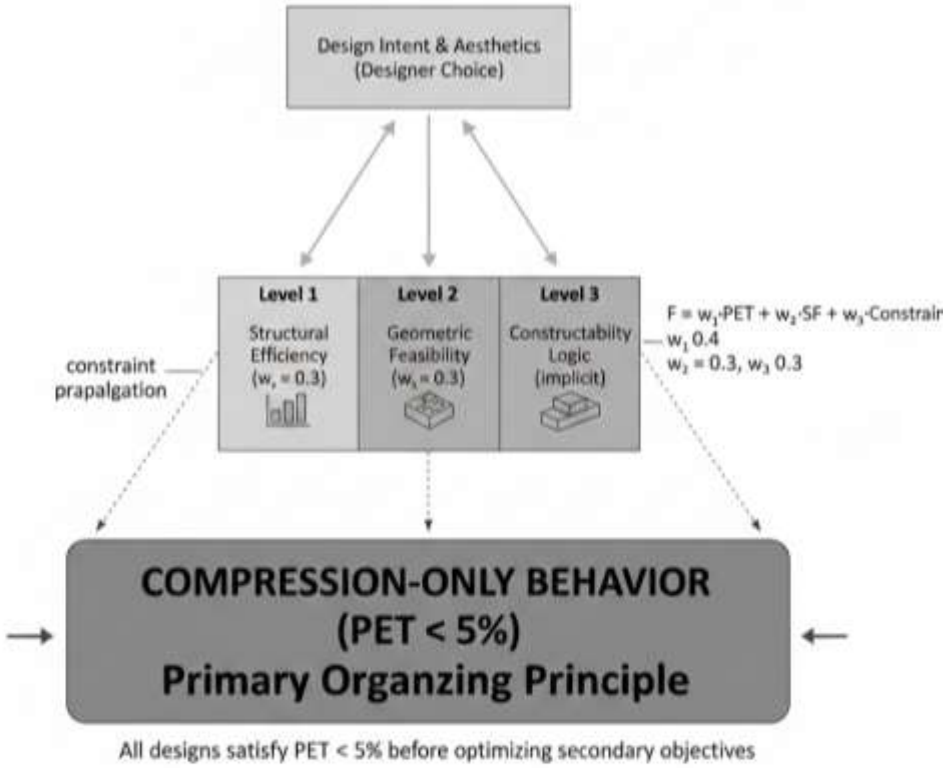


Fig. 5: Compression-Only Design as Organizing Principle: Hierarchical Integration of Framework Objectives,
source: created by the author

## VI. Discussion: Implications for Architectural Theory and Practice

### 6.1 Computational Design as Knowledge Formalization

The framework can be understood as a formalization—in algorithmic and computational terms—of tacit knowledge embedded in masonry tectonic tradition. Historical masons intuitively understood load paths, optimal material distribution, and compression-dominant behavior. The framework makes this intuitive knowledge explicit through mathematical formulation and algorithmic implementation.

This formalization serves both epistemological and practical purposes: it clarifies what we believe we know about masonry design (revealing assumptions and potential errors) and operationalizes that knowledge in computational tools accessible to contemporary practitioners.

### 6.2 Bridging Analogue and Digital Design Wisdom

The framework is not proposed as a replacement for traditional masonry knowledge but as a bridge enabling dialogue between traditional and contemporary methods. Historical form-finding techniques (thrust line analysis, graphical statics) provided different insights than topology optimization; each reveals different aspects of structural behavior. The framework supports integration of these perspectives rather than replacement.

### 6.3 Democratization of Optimization-Informed Design

Topology optimization has historically been an esoteric tool, accessible primarily to specialized researchers and large firms with computational resources. By embedding TO within parametric design workflows using accessible software (Grasshopper, open-source solvers), the framework contributes to democratization, making optimization-informed design available to a broader community of architects and designers.

### 6.4 Toward a Constructive Optimization Paradigm

The framework contributes to an emerging paradigm in computational design: "constructive optimization," wherein optimization is not a downstream analytical step but an active participant in design synthesis from inception. Rather than designing first and optimizing later, the framework integrates optimization throughout the design process, producing geometries that are simultaneously expressive, efficient, and constructable.

### 6.5 Limitations and Scope

The framework's theoretical contributions must be contextualized within explicit limitations:

1. **Static loading only:** The framework addresses static self-weight and lateral loads. Dynamic loads, time-dependent effects (creep, shrinkage), and thermal stresses are not explicitly incorporated.
2. **Homogenized material model:** Masonry is treated as a single isotropic material. Brick–mortar interface effects, anisotropy, and joint failure modes are not explicitly modeled.
3. **Compression-only assumptions:** The framework prioritizes compression-only behavior, which may be overly restrictive for some applications where limited tension (with appropriate reinforcement) is acceptable.
4. **Geometric restrictions:** The framework is illustrated for curved wall geometries. Its applicability to shells, vaults, or more complex three-dimensional forms requires further investigation.
5. **Constructability heuristics:** Constructability constraints (maximum corbel overhang, joint thickness) are based on traditional practice and may require recalibration for novel geometries or fabrication methods.
6. **Software environment:** Implementation assumes parametric design tools (Grasshopper, Karamba3D, Millipede) and may not be straightforwardly translating tools to alternative platforms.

### 6.6 Future Directions and Extensions

The framework opens several promising research directions:

1. **Multi-Objective and Multi-Constraint Optimization** Future work should systematically explore multi-objective formulations where multiple design criteria (material efficiency, aesthetic expression, constructability complexity, cost) are optimized simultaneously, yielding Pareto-optimal solution sets rather than single optima.
2. **Seismic Design and Dynamic Load Cases** Extending the framework to address seismic loads and dynamic response would require incorporation of modal analysis, response spectrum methods, or time-history analysis. This would enrich the framework's applicability to real-world scenarios where dynamic effects are significant.
3. **Temporal and Environmental Effects** Incorporating creep, shrinkage, and thermal effects would enable the framework to address long-term structural behavior and environmental durability—critical concerns in masonry design.
4. **Multi-Material Optimization** Extending the framework to combine masonry with reinforced concrete, steel inserts, or other materials in strategically optimized zones could substantially expand design possibilities while maintaining masonry's tectonic language.
5. **Fabrication Integration and Robotic Assembly** Direct integration with robotic assembly systems or 3D-printing technologies would close the loop from computational design to physical realization, enabling automated fabrication of topology-optimized discrete masonry assemblies.
6. **Machine Learning Acceleration** Training machine learning models on datasets of parametric designs and their performance metrics could enable rapid prediction of performance without explicit FEA, accelerating variant exploration by orders of magnitude.
7. **Historical Validation and Precedent Analysis** Retrospective application of the framework to historically significant masonry structures (Gothic cathedrals, Roman vaults, Guastavino tile vaults) could provide validation that the framework's compression-only and efficiency assumptions align with intuitions embedded in historically successful geometries.

## VII. Conclusion

This paper has presented a comprehensive theoretical framework for integrating topology optimization with discrete, modular masonry construction through iterative parametric feedback. The framework formalizes an approach to the discrete–continuum gap—a persistent methodological challenge in computational masonry design—by coupling parametric brick generation, evolutionary morphology selection, continuum topology optimization, and adaptive discretization within a unified, cyclical pipeline.

The key theoretical contributions are:

1. **Formalization of discrete–continuum integration:** The framework systematizes the translation from continuous optimization results to discrete constructible assemblies, accounting for both structural fidelity and constructability constraints.

2. **Principle-driven design:** By embedding compression-only design (quantified through PET metrics) as a primary organizing principle throughout multiple optimization phases, the framework demonstrates that structural logic can be a generative force in form-finding, not merely a post-hoc verification criterion.
3. **Cyclical feedback loops:** By creating bidirectional information flow between parametric generation, continuum optimization, and discrete validation, the framework enables more holistic exploration of design spaces that simultaneously satisfy structural, geometric, and constructive criteria.
4. **Multiple optima and designer agency:** Rather than prescribing a single "optimal" design, the framework supports exploration of solution spaces, enabling designers to navigate efficiency–aesthetics–constructability trade-offs according to project-specific values.

The framework is positioned as a methodological contribution applicable across varied masonry geometries, loading scenarios, and design intents. Its theoretical underpinnings are generalizable, though specific implementation details would require adaptation to applications.

Beyond its immediate technical contributions, the framework embodies a philosophical position: that computational design, when carefully structured, can recover and amplify the integration of form and force that characterized historical masonry practice. In an era where digital tools often amplify disconnections between aesthetic intent and structural logic, this framework demonstrates that integrative, principle-driven design remains possible and even advantageous.

Future research should extend the framework along the directions outlined above—seismic design, environmental effects, multi-material optimization, fabrication integration, and machine learning acceleration. Each extension would deepen the framework's applicability to real-world architectural challenges while maintaining its theoretical coherence.
The fundamental thesis remains topology optimization, when appropriately constrained and discretized, can re-establish the historic alignment between architectural form and structural necessity, producing designs that are simultaneously expressive, efficient, and constructable.

## References

**[1]** Heyman, J. (1995). The Stone Skeleton: Structural Engineering of Masonry Architecture. Cambridge University Press.
**[2]** Heyman, J. (1969). The safety of masonry arches. International Journal of Mechanical Sciences, 11(4), 363–385.
**[3]** Block, P., & Ochsendorf, J. (2007). Thrust network analysis: A new methodology for three-dimensional equilibrium. Journal of the International Association for Shell and Spatial Structures, 48(3), 167–173.
**[4]** Preisinger, C., & Schimek, H. (2015). Karamba3D—A toolbox for parametric structural design. Structural Engineering International, 25(2), 186–197.
**[5]** Bendsøe, M. P., & Sigmund, O. (2003). Topology Optimization: Theory, Methods, and Applications (2nd ed.). Springer-Verlag.
**[6]** Sigmund, O., & Maute, K. (2013). Topology optimization approaches: A comparative review. Structural and Multidisciplinary Optimization, 48(6), 1031–1055.
**[7]** Oxman, N. (2010). Structurally informed parametric design. Architectural Design, 80(4), 52–59.
**[8]** Fernandez-Jimenez, M., & Larson, C. (2015). Topology optimization in architectural design. Journal of the International Association for Shell and Spatial Structures, 156(1), 33–42.
**[9]** McNeel Associates. (2024). Rhinoceros 3D: Computational Design Environment. Retrieved from https://www.rhino3d.com
**[10]** Amorim, D., Lopes, A., & Menges, A. (2021). Towards a design-to-fabrication framework for complex masonry. Advances in Architectural Geometry, 2021, 234–255.
**[11]** Block, P., DeJong, M., & Ochsendorf, J. (2006). As hangs the flexible line: Equilibrium, accuracy, and computation of anticlastic masonry vaults. Nexus Network Journal, 8(2), 7–20.
**[12]** Piker, D., & Tynan, M. (2014). Kangaroo: Form Finding with Computational Physics. Advances in Architectural Geometry.
**[13]** Rafizadeh, S., Arvin, H., & Doroudgar, A. (2025). Fabrication-aware synthetic dataset generation for robotic brick assembly. Robotics and Autonomous Systems, 168, 104–121.

**[14]** Dörrie, B., Günther, D., & Brell-Cokcan, S. (2024). Learning to lay: Machine learning approaches for robotic masonry assembly. IEEE Access, 12, 48923–48941.

**[15]** Rots, J. G., Nauta, P., Kusters, G. M. A., & Blaauwendraad, J. (1985). Concrete mechanics—A finite element perspective. Journal of Engineering Mechanics, 111(7), 966–990.

**[16]** EN 1996-1-1:2005. (2005). Eurocode 6: Design of Masonry Structures—Part 1-1: General Rules for Reinforced and Unreinforced Masonry Structures. European Committee for Standardization.

**[17]** EN 1991-1-4:2005. (2005). Eurocode 1: Actions on Structures—Part 1-4: General Actions—Wind Actions. European Committee for Standardization.

**[18]** Ochsendorf, J. A. (2010). Guastavino Vaulting: The Art and Science of Vaulting. Princeton Architectural Press.

**[19]** Kneel, L., & Moseley, H. (2022). Optimization-based form-finding and structural design in contemporary architecture. Architectural Research Quarterly, 26(1), 45–62.

**[20]** Kumar, P., & Schmidt, R. (2018). Parametric structural optimization in masonry design: A review. Journal of Architectural Engineering, 24(3), 04018015.