# *Structural optimization of 3D masonry buildings*

**Citation:** Whiting, Emily, Hijung Shin, Robert Wang, John Ochsendorf, and Frédo Durand. "Structural Optimization of 3D Masonry Buildings." ACM Transactions on Graphics 31, no. 6 (November 1, 2012): 11 pages.

**As Published:** http://dx.doi.org/10.1145/2366145.2366178

**Publisher:** Association for Computing Machinery

**Persistent URL:** http://hdl.handle.net/1721.1/86239

**Version:** Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

# Structural Optimization of 3D Masonry Buildings

Emily Whiting Hijung Shin Robert Wang John Ochsendorf Frédo Durand
Massachusetts Institute of Technology

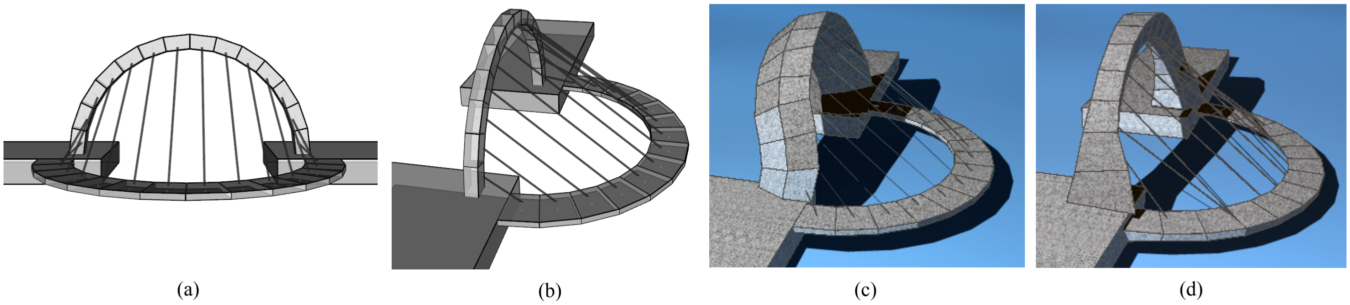


**Figure 1:** *We present a method to compute the gradient for the stability of a structure composed of rigid blocks, and demonstrate how we enable the optimization of stable structures. For example: (a) A cable bridge structure originally infeasible. (b) Side view of the input model. (c) Output feasible model. The horizontal arched walkway and cables are fixed, only the vertical arch is optimized. (d) An alternative feasible output. The horizontal arch and cable joints are free to deform with the constraint that top faces (walking surface) remain horizontal.*

## Abstract

In the design of buildings, structural analysis is traditionally performed after the aesthetic design has been determined and has little influence on the overall form. In contrast, this paper presents an approach to guide the form towards a shape that is more structurally sound. Our work is centered on the study of how variations of the geometry might improve structural stability. We define a new measure of structural soundness for masonry buildings as well as cables, and derive its closed-form derivative with respect to the displacement of all the vertices describing the geometry. We start with a gradient descent tool which displaces each vertex along the gradient. We then introduce displacement operators, imposing constraints such as the preservation of orientation or thickness; or setting additional objectives such as volume minimization.

**Keywords:** Statics, structural stability, architecture, optimization

## 1 Introduction

While computer graphics and computer-aided-design (CAD) have dramatically broadened the range of shapes available for architectural design, structural considerations have often been ignored. Structural analysis of a building is usually performed after the aesthetic design has been determined and has little influence on the overall form. An architect designs the shape, which is passed to structural engineers to make the building stable through the use of appropriate material and reinforcement. Existing structural analysis software, such as finite element analysis, is a powerful method for analyzing a given structure, but does not directly suggest ways to improve the geometry in order to reduce internal forces and required material. In contrast, we seek to propose modifications to the geometry that enhance structural soundness. We focus on masonry structures because their stability is the direct result of their geometry, but we argue that the central principle of sound masonry design – minimization of non-axial forces – extends to other materials.

The input to our method is a building geometry described as a set of blocks specified by their vertex coordinates. The central component of our approach is the notion of a structural gradient, which expresses, for each vertex, the displacement direction that maximally improves structural soundness. The gradients can be used in a steepest-descent manner. Alternatively, constraints can be introduced to modify the gradient direction, such as preservation of horizontal and vertical directions, or constant thickness of blocks. Objectives can also be added such as volume minimization to reduce material usage. We explore a number of gradient modifications and show that they enable variations in structurally sound models.

We base our notion of structural soundness on static analysis [Livesley 1978; Livesley 1992] and focus on masonry materials, comprising stone and brick structures. Masonry is the dominant material for traditional architecture and is also used in modern architecture, especially in developing countries. In contrast to contemporary steel or reinforced concrete, traditional masonry relies on forms which are inherently stable, because the material resists only axial compressive forces [Allen and Zalewski 2009]. Though we focus on the case of masonry, our approach can be used to minimize non-axial forces in general. Even with materials that resist tension, such as reinforced concrete or steel, a good structural form with reduced non-axial force requires less material, leading to cheaper, more environmentally-friendly, and robust buildings. In addition, we extend our approach to enable the treatment of cables as tension-only elements, using the same principles of static analysis and resistance to axial forces.

The heart of our approach is to compute the gradient of a stability metric with respect to geometry modification. First, we show that previous expressions of masonry instability [Whiting et al. 2009] do not lead to appropriate structural gradients because they are based on forces rather than torque. We use a stability metric defined by a quadratic program minimizing both tension and compression forces and subject to feasibility constraints. We compute the gradient of this metric with respect to all the vertex coordinates. To do this, we transform active inequality constraints into equalities, resulting in a linear system that we use to derive a closed-form expression for the optimum. We then analytically derive the Jacobians of the transformed feasibility constraints with respect to vertex coordinates. Together with the closed form expression of the optimum, this allows us to derive the final structural gradient.

We introduce geometry modification tools that leverage the gradient of stability. Each tool relies on a number of user-specified constraints and objectives, such as the preservation of vertical or horizontal directions or the minimization of material usage.

**Contributions** The following contributions are presented:

- We introduce a new stability metric that accurately quantifies infeasibility of a structure by incorporating torque imbalance.
- We provide a closed form derivation of the gradient of stability with respect to geometry modification.
- We describe a parameterization for geometry manipulation that respects planarity of the block primitives.
- We present an extension to tension elements that enables the inclusion of cable structures.
- We modify the gradient according to constraints and objectives to enable the user-guided improvement of stability.

## 2 Related Work

**Architectural Modeling** Attar et al. [2009; 2010] apply physics to generative design of architectural models, but do not address the creation of structurally feasible forms. Schiftner and Balzer [2010] use statics in mesh layouts, but applied at the initialization phase rather than a constraint on the geometry. Optimization has been used in architecture for modeling free-form surfaces that meet fabrication criteria [Eigensatz et al. 2010; Pottmann et al. 2008; Pottmann et al. 2007; Liu et al. 2006]. However, these examples do not consider structural feasibility constraints. Most related to our method, Whiting et al. [2009] integrate structural soundness with procedural modeling, but are limited to low-dimensional parameterized models. Our approach addresses the creation and modification of structurally-sound masonry structures defined as generic quad-mesh models.

**Optimization for Design** Shape design is often formulated as a numerical optimization. Delfour and Zolesio [2001] and Laporte and Le Tallec [2003] provide the mathematical foundations for optimization problems over a geometric domain. A range of surface modeling techniques incorporate geometry optimization [Gal et al. 2009; Kilian et al. 2008; Sorkine and Alexa 2007]. Most similarly, Welch and Witkin [1992] solve a constrained variational optimization for interactive modeling of free-form surfaces. Analogous to our approach they formulate the surface energy as a quadratic program, and incorporate user-controlled constraints such as fixed points. Harada et al. [1995] optimize constrained layout designs with physically based user interaction. Li et al. [2004] introduce constraint editing for guided optimization in image segmentation applications.

**Structural Design** Many commercial CAD modeling systems, such as CATIA (www.3ds.com/products/catia) and Revit (usa.autodesk.com/revit-architecture), integrate finite element analysis into the modeling package to shorten the pipeline between model creation and structural analysis. However, these systems do not provide guidance on how to modify designs to improve stability. The EifForm application [Shea 2000] integrates FEM with generative structural design methods, but differs from our technique by use of a simulated annealing algorithm.

Allaire et al. [2004] describe structural optimization combining shape derivatives with a level-set method, though their method is based on elasticity rather than rigid material. Smith et al. [2002] developed automatic optimization of truss structures. In trusses only axial forces are considered. Further, the solution space is typically restricted to two levels of joints, compared to the arbitrary stacking of blocks in masonry architecture. The Thrust Network Analysis method for the design of 3D masonry vaults considers equilibrium of compression-only structures [Block and Ochsendorf 2007; Vouga et al. 2012], however, the approach is specific to topologies that can be projected onto a 2D plane, while we support arbitrary topology and full 3D models.

**Computer Graphics Applications** Stability analysis has been applied to a wide range of model subjects in computer graphics applications with the notion that physical realism translates to a more realistic appearance. Shi et al. [2007] use static equilibrium as a constraint for determining plausible character poses. Static analysis has also been used in creating realistic tree structures [Hart et al. 2003]. Many geometric modeling applications integrate physical constraints and simulation [Martin et al. 2010; Xu et al. 2009; Terzopoulos and Fleischer 1988; Terzopoulos et al. 1987], but these systems target realistic deformations rather than stability.

## 3 Background: Static Analysis

This section reviews the feasibility conditions for a structurally sound masonry model, where the forces must satisfy static equilibrium, friction constraints and be in compression. Next, for infeasible structures, we review the method introduced by Whiting et al. [2009] for incorporating tension penalty forces. We later extend our feasibility method to cables (Section 7).

### 3.1 Contact Forces

We model structures as assemblages of rigid blocks. We discretize the force distributions at the interfaces between these blocks, positioning a 3D force $\mathbf{f}^i$ at each vertex of the interface (Figure 2). Each force $\mathbf{f}^i$ is decomposed into three components with respect to the local coordinate system of the interface: an axial component $f_n^i$ perpendicular to the face, and two orthogonal in-plane friction components, $f_u^i$ and $f_v^i$, where $u$ and $v$ are two edges of the block face. Friction forces on shared faces have opposite orientation.

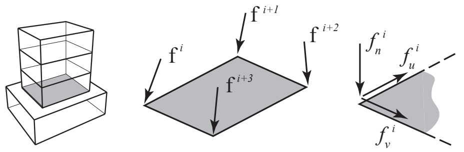


**Figure 2:** *Model of contact forces at interfaces between blocks [Whiting et al. 2009].*

### 3.2 Feasibility Conditions

**Static Equilibrium** Static equilibrium is enforced by setting net force and net torque for each block equal to zero, which can be expressed as a linear system [Livesley 1978]:

$$\mathbf{A}_{eq} \cdot \mathbf{f} + \mathbf{w} = \mathbf{0} \tag{1}$$

where $\mathbf{w}$ is a vector containing the weights of each block, $\mathbf{f}$ is the vector of interface forces, and $\mathbf{A}_{eq}$ is the sparse matrix of coefficients for the equilibrium equations (see supplemental material). External loads can be added using the $\mathbf{w}$ vector.

**Compression Constraint** Masonry material can resist extremely high compression, but mortar between blocks can provide only limited tension. This can be approximated by a non-negativity constraint on the axial forces:

$$f_n^i \geq 0, \quad \forall\, i \in \text{ interface vertices} \tag{2}$$

Over the entire structure this is expressed as a linear system of inequality (lower bound) constraints:

$$\mathbf{I}_{lb} \cdot \mathbf{f} \geq \mathbf{0}$$

**Friction Constraints** We approximate friction constraints by constraining tangential forces $f_u^i, f_v^i$ to be within a conservative friction pyramid of the normal forces $f_n^i$:

$$|f_u^i|, |f_v^i| \leq \alpha f_n^i, \quad \forall\, i \in \text{ interface vertices} \tag{3}$$

where $\alpha$ is the coefficient of static friction with a typical value of 0.7. We express the combined friction constraint across the structure as a sparse linear system of inequalities:

$$\mathbf{A}_{fr} \cdot \mathbf{f} \leq \mathbf{0}$$

### 3.3 Force Solution

As in Whiting et al. [2009], we solve for the forces of infeasible structures by translating the conditions in (3.2) into a penalty formulation that softens the compression constraint. Axial forces are expressed in terms of compression and tension using the difference of two nonnegative variables [Bertsimas and Tsitsiklis 1997]:

$$\begin{aligned} f_n^i &= f_n^{i+} - f_n^{i-} \\ f_n^{i+}&, f_n^{i-} \geq 0 \end{aligned} \tag{4}$$

where $f_n^{i+}$, $f_n^{i-}$ are the positive and negative parts of $f_n^i$. Variable $f_n^{i-}$ represents tension forces, and $f_n^{i+}$ represents compression.

The penalty formulation of (3.2) is a quadratic program:

$$\begin{aligned} \mathbf{f}^* = \quad & \text{argmin}_{\mathbf{f}} \quad \frac{1}{2}\,\mathbf{f}^T\mathbf{H}\mathbf{f} \\ & \text{s.t.} \quad \mathbf{A}_{eq} \cdot \mathbf{f} = -\mathbf{w} \\ & \qquad\;\; \mathbf{A}_{fr} \cdot \mathbf{f} \leq \mathbf{0} \\ & \qquad\;\; \mathbf{I}_{lb} \cdot \mathbf{f} \geq \mathbf{0} \end{aligned} \tag{5}$$

where $\mathbf{H}$ is a diagonal weighting matrix for the forces. In contrast to Whiting et al. [2009], which weighted only tension forces, this formulation places weights on all forces: $\mathbf{H}$ is full-rank with large penalty weight on the tension forces and low weight on the remaining forces (compression and friction). This is an intuitive method to account for the indeterminacy of structures – while many possible solutions of $\mathbf{f}$ may exist that satisfy the constraints, our formulation searches within that space for a solution that simultaneously minimizes tension and keeps overall force values low.

## 4 Measure of Infeasibility

We introduce a new torque-based formulation to measure closeness to a feasible structure. Previous work by Whiting et al. [2009] based feasibility on the magnitude of tension forces under static equilibrium. While their energy leads toward feasibility, it sometimes proposes futile changes. We resolve this limitation by decomposing penalty forces into torque and uniform forces.

Consider the example of a T shape in Figure 3(a) where the penalty solution exhibits tension (in blue) on the right of the interface between the two blocks. The only property of the bottom block that can improve equilibrium is to shift its left side until it reaches the center of mass of the top block. The location of the right side of this block is irrelevant around this configuration. However, when the right side of the bottom block is stretched further right, as in Fig. 3(b), static analysis shows that the tension forces decrease in magnitude. This is due to the increased torque arm – less force is needed to balance the off-center load from the top block. Consequently, in an optimization, gradients will indicate that the vertex translation improves feasibility, which is not the case – rather, the feasibility is unaffected since unbalanced torque remains the same. The result leads to higher material usage. This is a general problem, gradients produced through the energy of Whiting et al. [2009] consistently try to enlarge interfaces in order to increase the torque arm. Further, when the T block is constrained to retain the same dimensions, the competing gradients from the left and right facets mostly cancel each other (Fig. 3(c)), and the energy fails to reveal that the best solution is to translate the whole block to the left.

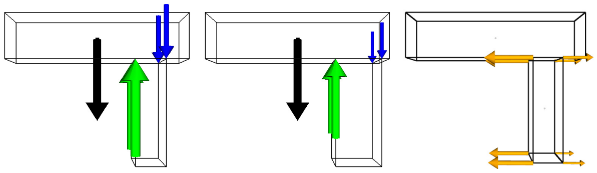

(a) Infeasible T (b) Infeasible T w/ stretched base (c) Gradients neglecting torque

**Figure 3:** *(a) Infeasible T-structure displaying locations of tension forces (blue), compression forces (green) and self-weight of the top block (black). (b) Stretching the base decreases tension forces but does not improve stability of the structure. (c) Gradients produced by the infeasibility metric of Whiting et al. [2009] push the bottom block in opposing directions.*

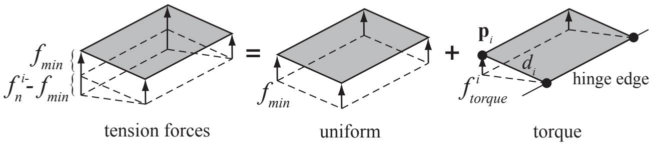


**Figure 4:** *On each block interface, tension forces are decomposed into a uniform component ($f_{min}$) and a torque component ($f^i_{torque}$). Arrow length represents force magnitude. We measure torque infeasibility contributions from each vertex $\mathbf{p}_i$ as $f^i_{torque}d_i$, where $d_i$ is the torque arm w.r.t. the hinge edge. The hinge sits along the vertices of minimum tension.*

We introduce a new infeasibility metric that incorporates the magnitude of torque contribution from tension forces. We decompose the tension forces on each interface into a uniform component and a torque component, as pictured in Figure 4:

$$y = \alpha y_{uniform} + y_{torque} \tag{6}$$

where $\alpha$ accounts for the differing units of uniform tension $[N^2]$ and torque energy $[(Nm)^2]$, and should be set appropriately for the scale of the model.

### 4.1 Uniform-Tension Energy

We define the energy for the uniform tension component as:

$$y_{uniform} = \frac{1}{2}\,\mathbf{f}^T \mathbf{H}_{uniform}\mathbf{f}$$

where $\mathbf{H}_{uniform}$ is a diagonal matrix that extracts the $f_{min}$ component at each block interface $k$:

$$y_{uniform} = \frac{1}{2}\sum_k \#verts_k * (f_{min_k})^2 \qquad (7)$$

Note that uniform tension energy will only be non-zero if all joint vertices are in tension.

### 4.2 Torque-Tension Energy

We define the energy for the torque component of tension forces as:

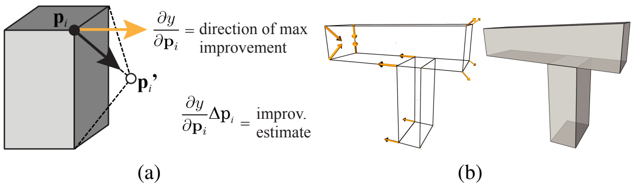
$$y_{torque} = \frac{1}{2}\,\mathbf{f}^T (\mathbf{I} - \mathbf{H}_{min})^T \mathbf{D}_{torque}(\mathbf{I} - \mathbf{H}_{min})\mathbf{f} \qquad (8)$$

$\mathbf{H}_{min}$ is a sparse matrix used to subtract the uniform component ($f_{min}$) from all tension forces. $\mathbf{D}_{torque}$ is a diagonal matrix with each element $d_i$ representing the torque arm of tension force $f_n^{i-}$. The torque arm is the distance from vertex $i$ to the hinge edge, where the hinge edge is defined as the line through the vertices at minimum tension value (Fig. 4). In most cases two vertices are at $f_{min}$ such that the hinge is an edge of the interface polygon. When a single vertex is at $f_{min}$, the hinge line is taken to be perpendicular to the resultant tension force.

In the T example (Fig. 3) the change in torque energy is zero since the tension force decreases but the torque arm increases.

## 5 Analytic Structural Gradient

Given an infeasible model, we demonstrate how to compute *structural gradients* $\nabla y(\Omega)$ that inform how to modify the geometry in a way that improves feasibility (Figure 5). In this section, we focus on the derivation of a closed-form gradient in the neighborhood of a solution of our quadratic program defining our structural infeasibility. We leave the parameterization of geometry general at this point but we will show in the next section that a parameterization that decouples displacements within the plane of an interface from translations and rotations of this plane facilitate the task.



$\mathbf{p}_i$
$\frac{\partial y}{\partial \mathbf{p}_i}$ = direction of max improvement
$\mathbf{p}_i$'
$\frac{\partial y}{\partial \mathbf{p}_i}\Delta \mathbf{p}_i$ = improv. estimate
(a)
(b)

**Figure 5:** *(a) Yellow arrows indicate the gradient of feasibility, parameterized on the model vertices $\mathbf{p}_i$. (b) The set of gradient vectors for an infeasible T-model. In the modified geometry the overall improvement in feasibility $\approx \sum_i \frac{\partial y}{\partial \mathbf{p}_i}\Delta\mathbf{p}_i$*

Our method has the following steps:

i) Starting from the quadratic program in (5) we transform inequality constraints into equalities by considering active constraints at a local solution of $\mathbf{f}^*$. This results in a QP with one big constraint matrix of linear equations.

ii) Given that the resulting QP has only equality constraints, we can derive a closed form expression for the force solution $\mathbf{f}^*$.

iii) Using the closed-form force solution ii) of the quadratic program (5), we derive an analytic gradient for the infeasibility metric $\nabla y(\Omega)$ in terms of geometry.

### 5.1 Closed Form Force Solution

The infeasibility metric of the structure $\Omega$ is formulated as a quadratic program. Let the force vector $\mathbf{f}^*$ be the global minimizer for the tension forces, as given in Expression (5).

The set of *active constraints* is the set of constraints that are satisfied as equalities at $\mathbf{f}^*$. The active friction constraints are denoted by $\widetilde{\mathbf{A}}_{fr} \cdot \mathbf{f} = \mathbf{0}$, and contain the friction inequality constraints that are satisfied exactly at their bound. The active lower bounds are denoted by $\widetilde{\mathbf{I}}_{lb} \cdot \mathbf{f}^* = \mathbf{0}$, which contains the axial forces from $\mathbf{f}^*$ that are exactly at zero.

We identify active constraints using the Lagrange multipliers returned by the QP solver. In the neighborhood of the current solution, we assume that the constraints remain active and that no new constraints intervene. This allows us to turn active constraints into equalities and ignore the inactive ones. We then combine the active constraints into a new set of equalities $\mathbf{C} \cdot \mathbf{f} = \mathbf{b}$, which is a concatenation of the static equilibrium constraints and the active inequality constraints.

$$\mathbf{C} = \begin{bmatrix} \mathbf{A}_{eq} \\ \widetilde{\mathbf{A}}_{fr} \\ \widetilde{\mathbf{I}}_{lb} \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} -\mathbf{w} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

Given the active constraints at $\mathbf{f}^*$, we can reformulate $\mathbf{f}^*$ as:

$$\begin{aligned} \mathbf{f}^* = \quad & \text{argmin}_{\mathbf{f}} \quad \frac{1}{2}\,\mathbf{f}^T \mathbf{H}\mathbf{f} \\ & \text{s.t.} \qquad \mathbf{C} \cdot \mathbf{f} = \mathbf{b} \end{aligned} \qquad (9)$$

which takes the form of an equality-constrained quadratic program. Recall that $\mathbf{H}$ is a full-rank diagonal matrix with large penalty weight on the tension forces and low weight on the remaining forces (compression and friction). A benefit of adding weights to all forces is that $\mathbf{H}$ is positive definite, which is necessary to derive a closed form solution of $\mathbf{f}^*$.

The new equality constrained QP in expression 9 can be re-written as finding the minimum norm solution to a linear system [Bertsekas 1995]. The solution of this problem is obtained through the Moore-Penrose pseudoinverse. The optimal solution in closed form is:

$$\mathbf{f}^* = \mathbf{H}^{-1}\mathbf{C}^T(\mathbf{C}\mathbf{H}^{-1}\mathbf{C}^T)^{-1}\mathbf{b} \qquad (10)$$

It can be seen that $\mathbf{f}^*$ as given above satisfies $\mathbf{C}\mathbf{f} = \mathbf{b}$ as required. Note that $\mathbf{H}$ must be positive definite, and the rows of $\mathbf{C}$ must be linearly independent [Bertsekas 1995]. Since we know the structure of the constraint matrices $\mathbf{A}_{eq}$ and $\mathbf{A}_{fr}$, we remove linearly dependent rows from $\mathbf{C}$.

### 5.2 Energy Derivatives

The expression for the gradient of $y(\Omega)$, combining uniform and torque tension energies, is given by:

$$\begin{aligned} \nabla y &= \alpha \nabla y_{uniform} + \nabla y_{torque} && (11) \\ &= \nabla\Big(\mathbf{f}^{*T}(\alpha \mathbf{H}_{uniform} + \mathbf{H}_{torque})\mathbf{f}^*\Big) && (12) \end{aligned}$$

The gradient $\nabla \mathbf{f}^*$ is possible to evaluate analytically because of our closed form expression for the force solution (Section 5.1). We make the assumption that the set of active constraints stays fixed for small displacements of the model geometry $\Omega$. Matrices $\mathbf{H}_{uniform}$ and $\mathbf{H}_{min}$ are constant since it is assumed the set of minimum-tension vertices remain the same for differential movement. See the supplemental material for a detailed derivation including constraint gradients.

In section 6 we describe the components of our novel shape parameterization.

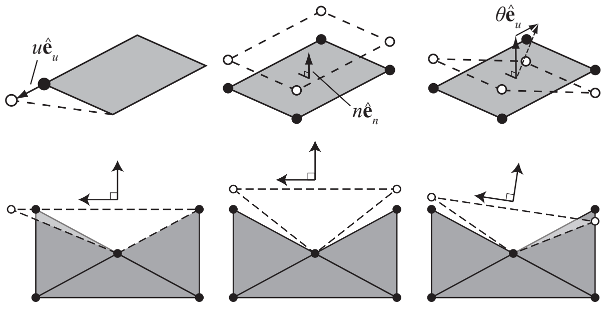
### 5.3 Hessian Approximation

In addition to computing the first-order gradient, we implement the limited-memory Broyden-Fletcher-Goldfarb-Shanno method (L-BFGS) for non-linear optimization [Fletcher 1987]. L-BFGS searches for the minimum by approximating the Hessian matrix, using the first-order gradient. According to our experience, L-BFGS converges to a stable solution with fewer iterations than using only the first-order gradient directly (see Fig. 16), particularly for models with higher complexity.

## 6 Shape Parameterization

To compute the gradient of infeasibility $\nabla y(\Omega)$ we parameterize $\Omega$ using a basis of vertex modifications. The fundamental constraint in choosing the parameterization is to maintain planarity of the block faces. Although contact surfaces between blocks could be represented as a triangular mesh with no planarity constraints, this would break assumptions in structural behavior. Non-planar joint geometry would eliminate the smooth friction surfaces and result in interlocking block faces which are rare in masonry construction.

We use the following parameterization which accounts for five degrees of freedom: in-plane vertex translation, translation of the face plane, and orientation of the face plane (See Figure 6). We assume topology remains consistent under a differential change in the block geometry: block adjacencies and the number of vertices in each contact polygon remain the same. These assumptions are necessary for the quadratic program (9) to be differentiable.



in-plane translation normal translation normal rotation

**Figure 6:** *Gradient parameterization. (Left) In-plane vertex translation. (Center) Face translation along normal. (Right) Face rotation by way of rotation of the normal vector. Only the tetrahedrons incident to the modified face are affected.*

**In-plane vertex translation** The two in-plane orthogonal vectors, $\hat{\mathbf{e}}_u$ and $\hat{\mathbf{e}}_v$, are the basis for vertex translation on the face plane (Figure 6, left). Vertex position, $\mathbf{p}$, is given by

$$\Delta\mathbf{p} = u\,\hat{\mathbf{e}}_u, \quad \Delta\mathbf{p} = v\,\hat{\mathbf{e}}_v \tag{13}$$

Using these parameters, the partial derivatives of the infeasibility metric $\partial y/\partial u_{k,i}$ and $\partial y/\partial v_{k,i}$ are computed for each vertex $i$ on every block face $k$. If a vertex is shared by multiple faces, a derivative will be computed for each face independently. Figure 7(a) shows the in-plane gradient for an asymmetrical T-model.

**Normal translation** The gradient is computed w.r.t. translation of a block face along the face normal $\hat{\mathbf{e}}_n$ (Figure 6, middle). All vertices $\mathbf{p}$ on the face are moved together with the relation:

$$\Delta\mathbf{p} = n\hat{\mathbf{e}}_n \tag{14}$$

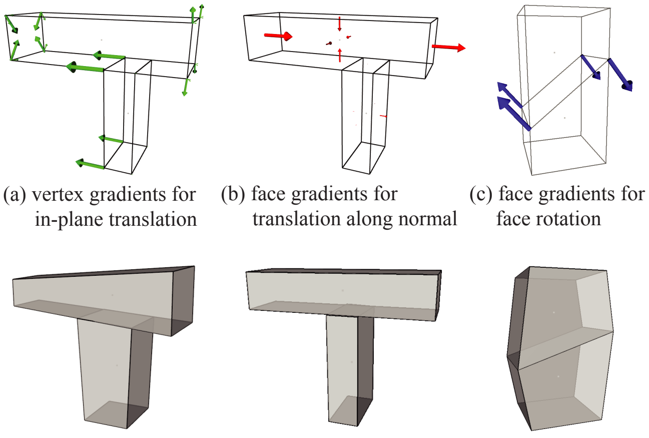
Using this parameter, the partial derivatives of the infeasibility metric $\partial y/\partial n_k$ are computed for each block face $k$. Figure 7(b) shows the result of the normal-translation gradient for an asymmetrical T-model.

**Normal rotation** Two angle parameters, $\theta$ and $\phi$, are the basis for rotation of the face plane. We set the center of rotation at the centroid, $\mathbf{c}$, of the block face. The derivatives of the face vertices are

$$\Delta\mathbf{p} = \theta\hat{\mathbf{e}}_v \times (\mathbf{p} - \mathbf{c}), \quad \Delta\mathbf{p} = \phi\hat{\mathbf{e}}_u \times (\mathbf{p} - \mathbf{c}) \tag{15}$$

Under the rotation parameters, the derivatives of the infeasibility metric $\partial y/\partial \theta_k$ and $\partial y/\partial \phi_k$ are computed for each block face $k$ in the structure. We solve a pseudoinverse to find $\partial\theta_k/\partial\mathbf{p}_i$, which takes into account that vertices are dependent under face rotation.

As discussed previously, Figure 7(c) shows how the orientation of interfaces between blocks affects the stability. In the 2-block example, rotating the interface toward a horizontal orientation reduces the infeasibility due to friction failure.



(a) vertex gradients for in-plane translation (b) face gradients for translation along normal (c) face gradients for face rotation

(d) improved geometry: vertices moved in direction of gradient

**Figure 7:** *Parameterized gradients for infeasible 2-block examples. (a-c) Gradients point in the direction of improved stability. (d) Stable output produced by moving the vertices along the gradient.*

### 6.1 Gradient w.r.t. vertex position

Ultimately we want to know how the infeasibility of a structure changes w.r.t. arbitrary changes in the geometry. Given the gradient components from the above parameterization, we can compute the derivative of the energy $y$ with respect to a vertex position, $\mathbf{p}_i$. This is expressed as the sum of all gradient components over all block faces that share the vertex:

$$\frac{\partial y}{\partial \mathbf{p}_i} = \sum_k \sum_{\omega \in W_{k,i}} \frac{\partial y}{\partial \omega}\frac{\partial \omega}{\partial \mathbf{p}_i}, \quad \nabla y(\Omega) = \left[\frac{\partial y}{\partial \mathbf{p}_1} \cdots \frac{\partial y}{\partial \mathbf{p}_N}\right]^T$$

where $W_{k,i} = \{u_{k,i},\ v_{k,i},\ n_k,\ \theta_k,\ \phi_k\}$. Recall that $u$ and $v$ derivatives are computed for each vertex and $n, \theta$ and $\phi$ are computed per face. The gradient of the infeasibility metric parameterized by vertex positions $\mathbf{p}_i$ is $N \times 3$, where $N$ is the number of vertices in the structure.

Figure 5 shows the gradient of an unstable two-block T structure. Our gradient computation takes into account two properties:

1. **Mass:** Imbalanced torque is reduced by widening the top block on the right, and thinning the overhanging portion on the left.
2. **Joint Geometry:** Increasing the width of the bottom block decreases the overhang of the top block, thus improving instability. Joint vertices are updated along with block geometry and are included in the gradient computations. Recall that contact forces are positioned at joint vertices.

## 7 Tension Elements

Our approach based on quadratic programming can handle other types of constraints such as those corresponding to tension-only elements. In particular, we have implemented an extension to enable cables in architectural designs. Whereas the same principles of static analysis and gradient computation apply, cables are different from rigid blocks in several ways:

1. Cables can resist high tension, but fail under compression. We model cables to be infinitely strong in tension, and penalize compression forces.
2. There are no friction constraints. We model cables as infinitely thin elements that are firmly attached to each other or to another block.
3. Cables can apply forces only along their axis. This means that a structure can be infeasible not only due to the sign of the required forces (i.e. compression on cables or tension on blocks), but because of its limited degrees of actuation. For example, Figure 8 shows a case where there is no configuration of forces along the cable axis that achieves static equilibrium. In order to compute a measure of infeasibility for such structures, we add virtual torques around the centroid of each element and include them in the penalty function.

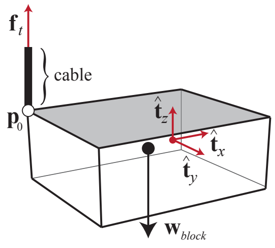


**Figure 8:** *The tension force $\mathbf{f}_t$ applied by the cable balances the block weight, but creates unbalanced torque. Virtual torques are added in directions $\hat{\mathbf{t}}_x$, $\hat{\mathbf{t}}_y$ and $\hat{\mathbf{t}}_z$ around the centroid of the block to solve for static equilibrium.*

The geometry of each cable is defined by its two end points $\{\mathbf{p}_0,\ \mathbf{p}_1\}$ which may be attached to an adjacent cable or to a point on a block surface. The direction of the tension force, $\mathbf{f}_t$, is along its axis (Fig. 8):

$$\mathbf{f}_t = f_t \hat{\mathbf{e}}_t, \quad \hat{\mathbf{e}}_t = \frac{\mathbf{p}_0 - \mathbf{p}_1}{\|\mathbf{p}_0 - \mathbf{p}_1\|} \qquad (16)$$

Analogously to blocks, we split $f_t$ into positive and negative components, $f_t^+$ and $f_t^-$, except in cables we penalize $f_t^+$. It is possible to place an upper bound on $\mathbf{f}_t$, which should reflect the scale of the structure as strength is relative to cross-sectional area.

Cable elements are assigned a weight, $\mathbf{w} = \rho_c L \hat{\mathbf{g}}$, where $L$ is the length of the cable and $\rho_c$ is the mass per unit length.

**Structural Gradient** For gradient computation, we parametrize cables using the $x, y, z$ coordinates of their end points. We consider geometric changes but not topological changes such as detaching a cable. The derivation for the gradient of cable geometry is given in the supplemental material.

## 8 Gradient Modifications

Given the gradients of infeasibility w.r.t. the chosen parameterizations, we now show how to improve the stability of a structure while preserving properties of its design. We frame the task as an optimization problem that steps towards a more structurally feasible design while allowing a user to specify constraints on the desired shapes and range of acceptable changes.

### 8.1 Snapping To Gradient

The first type of constraint is to modify vertex positions in the direction of the computed gradient. At each vertex in the structure, we compute a displacement vector $\Delta\mathbf{p}$ that combines all contributions from in-plane movement, face translation and face rotation. We define an optimization that solves for vertex positions best matching the desired gradient vector while maintaining planarity constraints of the block faces and coincidence constraints of contiguous blocks.

$$\begin{aligned} \mathbf{p}^* = \underset{\mathbf{p}}{\text{argmin}}\ & \|\mathbf{p} - \mathbf{p}_0 + \Delta\mathbf{p}\|^2 \\ \text{s.t.}\ & \text{faces remain planar} \\ & \text{block interfaces remain coincident} \end{aligned}$$

Under the assumption of quad-faced blocks, we adopt a planarity constraint that enforces the angles of each face to sum to $2\pi$ [Liu et al. 2006]:

$$f_{planar}(\mathbf{p}) = \sum_F \|\phi_F^1 + \ldots + \phi_F^4 - 2\pi\|^2$$

Our coincidence constraint ensures that each pair of coincident faces $(F, G) \in C$ between two adjoining blocks remains connected:

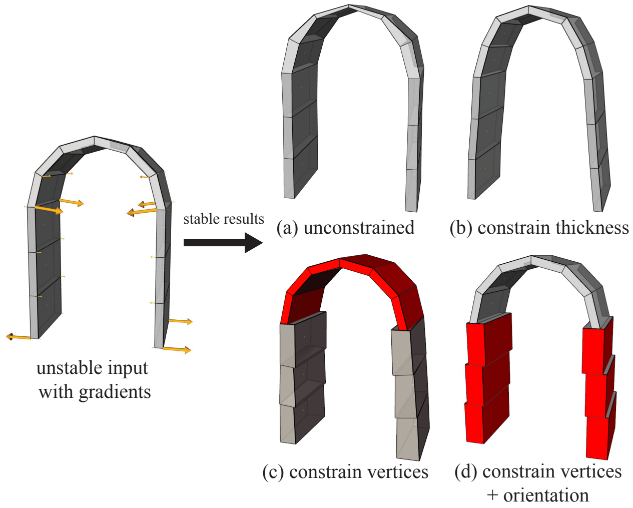
$$f_{coin}(\mathbf{p}) = \sum_{(F,G)\in C} \sum_{j \in 1\ldots4} \|\hat{\mathbf{e}}_{n_F}(\mathbf{p}_G^j - \mathbf{p}_F^1)\|^2$$

Due to the nonlinearity of these constraints, we enforce them using a penalty function technique.

$$\mathbf{p}^* = \underset{\mathbf{p}}{\text{argmin}}\ \|\mathbf{p} - \mathbf{p}_0 + \Delta\mathbf{p}\|^2 + \lambda_1 f_{planar}(\mathbf{p}) + \lambda_2 f_{coin}(\mathbf{p})$$

We use a Gauss-Newton iteration approach to solve for the best vertex positions satisfying our constraints. Characteristic to penalty techniques, the constraints are satisfied within a set tolerance. In practice we set this tolerance to $5 \times 10^{-3}$, which can be modified with trade-offs between accuracy and performance.

Though the penalty function is not convex, in practice we begin in a state that is close to satisfying planarity and coincidence constraints. The original vertex positions $\mathbf{p}_0$ satisfy the constraints by construction. We can control the difficulty of the optimization by reducing the size of the displacement $\Delta\mathbf{p}$ to ensure a valid solution.



(a) unconstrained (b) constrain thickness

(c) constrain vertices (d) constrain vertices + orientation

**Figure 9:** *The user may explore possible designs with improved stability by modifying constraints on the gradient. The initial arch is unstable, yellow arrows indicate gradients for improving feasibility. (a) Feasible unconstrained result. (b) Feasible output with constrained thickness. (c) The arch blocks are fixed resulting in thicker and slightly tapered column blocks. (d) In addition to constrained arch vertices, the column blocks have fixed orientation.*

## 8.2 Constraints

While the full gradient provides useful information on how a structure can be optimally modified to improve feasibility, a user may be interested in incorporating constraints that express design intent. Our system features a set of high-level constraints, such as block thickness or floor orientation which allow design variations. We describe how these controls can be implemented by modifying the gradient with low-level constraints on point positions and face normals.

**Fix block thickness** This is achieved by constraining the distance between two opposing faces $(F, G)$ of a block to be constant.

$$f_{thickness}(\mathbf{p}) = \sum_{(F,G)} \sum_{j=1\ldots4} \left\| th(F,G,j,\mathbf{p}) - th(F,G,j,\mathbf{p}_0) \right\|^2$$

where $th(F,G,j,\mathbf{p}) = \hat{\mathbf{e}}_{n_F} \cdot (\mathbf{p}_F^j - \mathbf{p}_G^j)$. In Figure 9(b) the arch has constrained thickness, resulting in a catenary-like output.

**Fix vertices** This is implemented by setting gradients of unaffected vertices to zero and pinning them to a specific location in the optimization. Alternatively, we can remove these vertices from the optimization altogether. In Figure 9(c) the arches are kept constant, only column vertices are modified.

$$f_{point}(\mathbf{p}) = \sum_{(i,\mathbf{p}')} \left\| \mathbf{p}^i - \mathbf{p}' \right\|^2$$

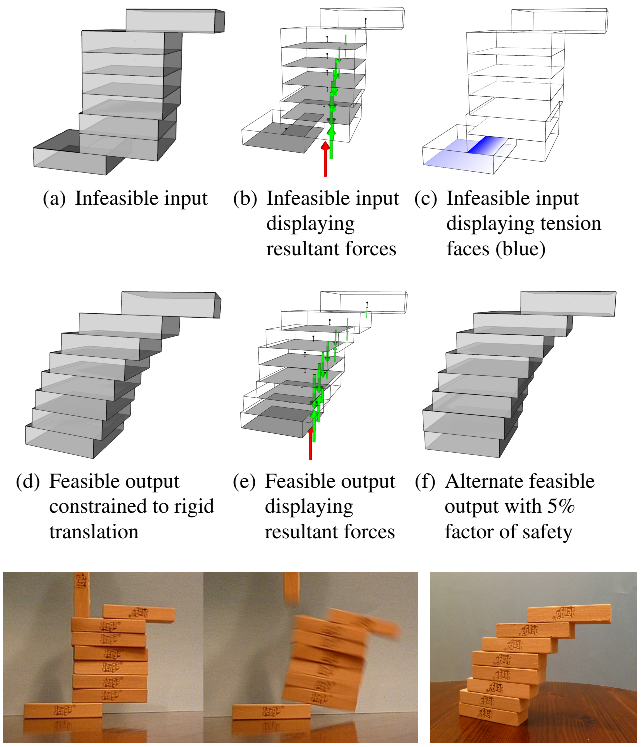
**Fix face orientation** We fix horizontal floors and vertical walls by zeroing rotational gradients and constraining a face to have a specific normal vector. For example, in Figure 9(d) all column block faces are orientation-constrained to maintain vertical faces.

$$f_{normal}(\mathbf{p}) = \sum_{(F,\hat{\mathbf{e}}_{n_F})} \sum_{j=2\ldots4} \left| \hat{\mathbf{e}}_{n_F} \cdot \frac{\mathbf{p}_F^j - \mathbf{p}_F^1}{\|\mathbf{p}_F^j - \mathbf{p}_F^1\|} \right|^2$$

**Fix block volume** To constrain volume, we divide each block $i$ into five tetrahedra $T_{i,j}$ and constrain the sum of the volumes of the five tets to remain constant. $v_T(\mathbf{p})$ is the volume of the tetrahedron $T$ given vertex positions $\mathbf{p}$.

$$f_{volume}(\mathbf{p}) = \sum_i \left\| \sum_{j=1\ldots5} v_{T_{i,j}}(\mathbf{p}) - v_{T_{i,j}}(\mathbf{p}_0) \right\|^2$$

Additionally, we can achieve other design constraints through combination of the penalty functions. For example, restricting a block to rigid translation (Figure 13(b)) involves applying a thickness and orientation constraint to each pair of opposing faces.



(a) Infeasible input (b) Infeasible input displaying resultant forces (c) Infeasible input displaying tension faces (blue)

(d) Feasible output constrained to rigid translation (e) Feasible output displaying resultant forces (f) Alternate feasible output with 5% factor of safety

(g) Unstable input falls after release (h) Stable output

**Figure 10:** *(a-c) Input: unstable stack of 8 identical blocks. Top and bottom blocks are fixed (red vertices), the remaining blocks are constrained to rigid translation. (d,e) Stable structure after multiple iterations. Resultant forces lie inside block interfaces. (f) Alternate solution with small safety factor by shrinking interfaces 5%. (g,h) Physical models. Hinge point at failure corresponds to interface with greatest tension (intensity of blue shading in (c)).*

## 8.3 Objective Functions

In addition to constraints on the gradient direction, the user may incorporate other design criteria by setting custom objective functions. We demonstrate this capability with a volume minimization example. The new multi-objective function becomes a weighted combination of the infeasibility metric $g$ and the total volume $v$.

$$\nabla y(\Omega) = \nabla g(\Omega) + \gamma \nabla v(\Omega)$$

where $\gamma$ is a weighting on the volume minimization. The derivative of $v$ w.r.t. each parameterization is identical to that used for weight in the $\mathbf{w}$ vector without the constant term for density (see supplemental material). Figure 11 shows an example of this objective applied to a structure of stacked blocks.

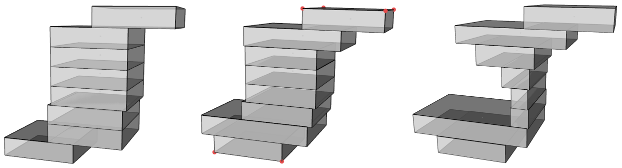

(a) Input (b) without volume minimization (c) with volume minimization

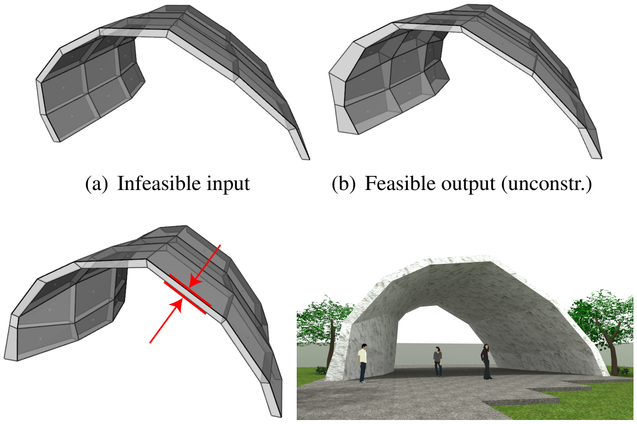
**Figure 11:** *Variations based on volume minimization. In (b,c) the constraints are face orientation and vertical thickness (horizontal stretching is permitted): (a) Infeasible input, weight = 160 units. (b) Feasible output, weight = 174 units. (c) Feasible output with volume minimization, weight = 127 units, initial $\gamma_0$ = 0.5.*

## 9 Results

**Modeling Stable Structures** We show example structures with visualizations of the computed infeasibility gradients, and various ways in which the user might modify the structure.

Figure 9 shows a variety of outputs for an infeasible model of an arch balanced on two columns. We compare results from unconstrained optimization, constant thickness and fixing arch vertices.

Figures 10 and 11 show various results for an unstable stack of blocks. In Figure 10 the constraints include rigid translation using a combination of thickness and orientation constraints. We show visualizations of the resultant forces at each interface, illustrating that resultants lie within the joints for the feasible result. We also apply a small factor of safety by shrinking the effective interfaces between blocks. Figure 11 applies only orientation constraints and vertical thickness, so that the blocks can stretch horizontally. An alternative design using volume minimization is provided.


(a) Infeasible input (b) Feasible output (unconstr.)

(c) Alternative feasible output (with thickness constraints) (d) Second feasible output in context

**Figure 12:** *(a) Starting from an unstable shell structure, modifying the geometry according to the gradients paired with user-controlled objectives can lead to two possible structurally-sound solutions: (b) A thickened profile to support the bulging shape. (c) A shifted arch shape with user-constraints on block thickness.*

In Figure 12(a) the shell is infeasible due to a bulging shape over the support region. Result 12(b) shows an output from unconstrained optimization where the profile is significantly thickened. In 12(c), the thickness of the shell is constrained, resulting in a modified shape closer to a traditional arch.

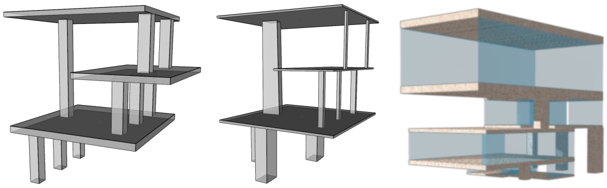

(a) Unstable input (b) Feasible output (constrained) (c) Example building

**Figure 13:** *(a) An infeasible building with columns supporting a system of slabs ($y_0 = 7.65 \times 10^5$). (b) Stable output with fixed floors and rigidity constraints on the base columns ($y^* = 8.6 \times 10^{-7}$). (c) Another stable multi-level design with a strong overhang.*

In Figure 13, the unstable input model consists of three slabs and a series of columns. The result in Figure 13(b) has constraints on slabs orientation, and rigid translation of the base columns. There is noticeable thinning and shortening of the slabs to reduce weight. Columns on the over-hanging portion are thinned while the left-hand column is thickened for counter-balance. The slabs and columns were fixed part-way through the design when a minimum preferred thickness was reached. The base columns are translated to the center of the floorplan so that the resultant load of the upper levels falls within their support region. Figure 14 shows a partial torus with doors, which is made stable through geometry changes following the gradient.

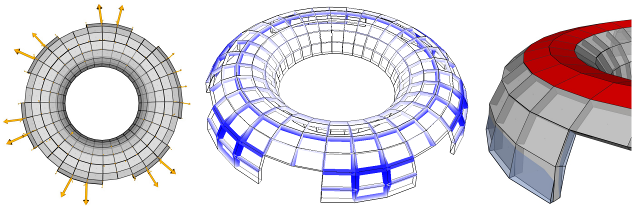
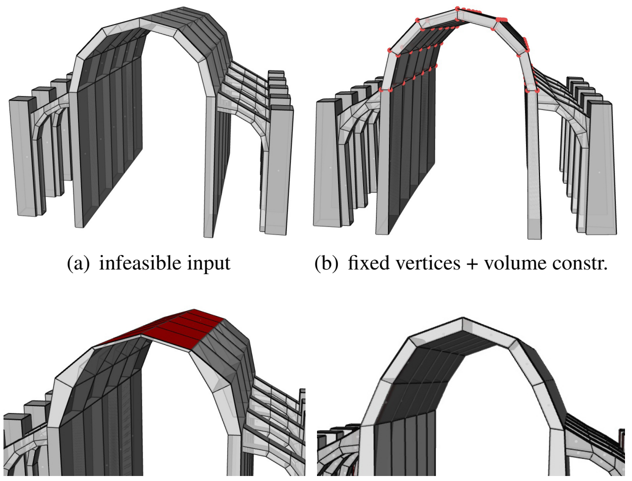

(a) (b) (c)

**Figure 14:** *(a) An infeasible half-torus model with cut out entranceways. Yellow arrows represent the gradient. (b) Faces in tension colored in blue. (c) The improved structure changes the curvature at the base blocks. Red blocks were constrained in thickness.*

The starting shape in Figure 15 is a barrel vault with flying buttresses. With a constraint to maintain volume, the result changes the buttress tapering, as well as modifying the curvature of the arches.

**Implementation** We use the IBM CPLEX quadratic program solver. Intel MKL is used for nonlinear optimization of block planarity under user constraints. In section 5.2 it is known that solving the normal equations $\mathbf{x} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b}$ explicitly is prone to numerical errors [Trefethen and Bau 1997]. For improved numerical stability we use QR decomposition to compute the inverse $\mathbf{E}^{-1} = (\mathbf{C}\mathbf{H}^{-1}\mathbf{C}^T)^{-1}$.

**Performance** Our formulation for analytic gradients provides a significant performance improvement over finite differencing techniques, which is necessary for interactive applications. See Table 1 for results and comparison. Performance times for finite differencing are equivalent to the time for solving the quadratic program (§4)



(a) infeasible input (b) fixed vertices + volume constr.

(c) unconstrained (d) thickness constrained

**Figure 15:** *(a) An infeasible barrel vault with flying buttresses. (b) Improved structure with walls and buttress blocks volume-constrained and arch vertices fixed (red). (c,d) Dominant changes are in the shape of the arched ceiling when unconstrained or thickness constrained.*

multiplied by the number of components in the gradient vector $\nabla y$. Times for analytic gradients include the time for computing partial derivatives (§5) added to the time for computing planar vertex offsets (§8.1). While computing the analytic derivatives takes some time, it is orders of magnitude faster than running the quadratic program solver multiple times as in the finite differencing approach.

Table 2 provides results for the optional user constraints (8.2). On top of planarity constraints, additional user constraints do not affect performance of each iteration significantly.

| model | blocks | length($\nabla y$) | gradient | finite diff. |
|---|---|---|---|---|
| arch (Fig. 9) | 12 | 792 | 0.41 s | 127 s |
| stack (Fig. 10) | 8 | 528 | 0.36 s | 84.5 s |
| shell (Fig. 12) | 24 | 1584 | 1.19 s | 269 s |
| torus (Fig. 14) | 232 | 15312 | 80.6 s | $8.37{\times}10^3$ s |
| torus (Fig. 14) | 400 | 26400 | 138.6 s | $2.28{\times}10^4$ s |
| torus (Fig. 14) | 576 | 38016 | 720.6 s | $4.95{\times}10^4$ s |

**Table 1:** *Performance results for computing the analytic structural gradient with comparisons to finite differencing.*

| model | w/o constr. | w/ constr. | # constraints |
|---|---|---|---|
| arch (Fig. 9) | 0.08 s | 0.09 s | 14 (*t*, *o*) |
| stack (Fig. 10) | 0.09 s | 0.12 s | 70 (*t*, *v*, *o*) |
| shell (Fig. 12) | 0.60 s | 0.73 s | 6 (*t*) |
| shell (Fig. 12) | 0.60 s | 0.75 s | 12 (*t*) |

**Table 2:** *Performance of gradient modifications: timings for planarity constraints with and without optional user constraints. Additions include block thickness (t), fixed vertices (v), and face orientation (o).*

**Convergence** Our method typically takes only a few iterations to converge to a feasible structure. Using a stopping criteria of 1% of the initial infeasibility measure, the arch (Fig. 9(b-d)) and shell (Fig. 12(b,c)) took $\leq 4$ iterations. The cable bridge (Fig. 1(c,d)) required 10 iterations. The stacked block model (Fig. 10(d)) required 40 iterations but is a difficult case where only a small range of block configurations is feasible.

The plot in Figure 16 shows the progress of the infeasibility metric for the unconstrained arch (Fig. 9(a)). At each iteration the value of infeasibility decreases, and converges toward $y(\Omega) = 0$ (feasibility condition) after a small number of gradient steps.

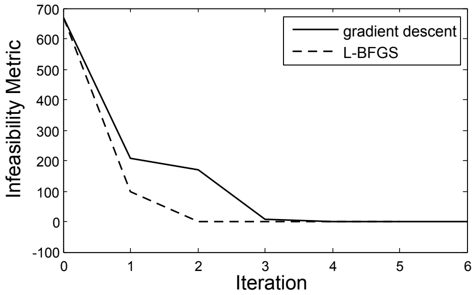


**Figure 16:** *Infeasibility vs. Iteration for the unconstrained arch (Fig. 9(a)). The model converges to a feasible solution ($y \approx 0$) in a small number of iterations, which is improved further by L-BFGS.*

**Validation** The computation of analytic gradients involves a number of challenges such as appropriate parameterization and turning inequality constraints into equalities. We validated correctness using finite differences. In the 2-block T model of Figure 7, we compared the change in constraint matrices $\Delta\mathbf{A}_{eq}$ and $\Delta\mathbf{w}$ with the analytic result $(\Delta\mathbf{A}_{eq})_{analytic} = \Sigma_i(\partial\mathbf{A}_{eq}/\partial\mathbf{p}_i)\Delta\mathbf{p}_i$ over selected vertices $i$. Varying groups of vertices were tested with shifts along $x, y, z$ axes. In all cases the error was below 1%. In the 2-block model of Figure 7(c), we measured the energy change from a small shift in the angle of the interface. The result of a forward finite difference was accurate to 2.3% of the analytic value for $\nabla y$.

The infeasibility metric introduced by Whiting et al. [2009] systematically leads to artificial thickening of blocks to increase the torque arm. Our new energy tackles this and leads to stable designs that are thinner. Comparisons of material use are given in Table 3. Using each feasibility metric, we applied gradient descent until convergence to a feasible structure, then measured weight increase of the result. Measurements were taken for a specified set of blocks affected by the optimization.

| model | weight increase | | part modified |
|---|---|---|---|
| | new energy | [Whiting‘09] | |
| T (Fig. 7) | 105 % | 129 % | bottom block |
| arch (Fig. 9) | 9.9 % | 31.1 % | 6 column blocks |
| stack (Fig. 10) | 148 % | 189 % | bottom block |
| shell (Fig. 12) | 256 % | 367 % | bottom left 3 rows |
| slabs (Fig. 13) | 180 % | 500 % | 3 base columns |
| buttress (Fig. 15) | 34.5 % | 66.7 % | buttress blocks |

**Table 3:** *Comparison in material use for our torque-based energy vs. [Whiting et al. 2009]. Weight increase is measured for the specified blocks after convergence to a feasible structure. Vertices other than the part modified were fixed. An exception was the arch structure which was unconstrained in order to reach feasibility.*

**Limitations** Planarity and user constraints may compete with the displacement vector $\Delta\mathbf{p}$ and in some cases may have the effect of

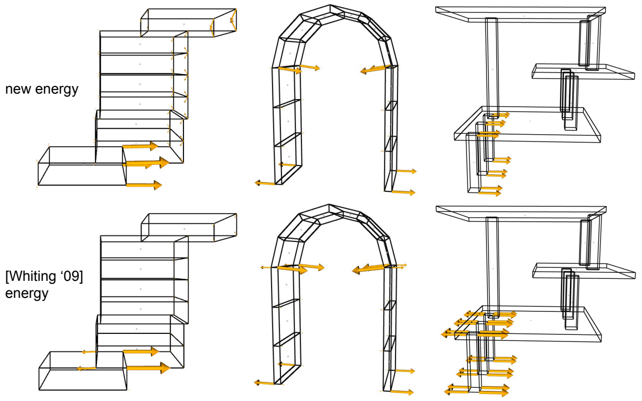


**Figure 17:** *Comparison in gradients between our torque-based energy (top), and the infeasibility metric of Whiting et al. [2009] (bottom). Gradients are shown for the input infeasible structure.*

pushing vertices away from the gradient direction. In this case $\Delta\mathbf{p}$ may decrease the feasibility of the structure.

The gradient $\nabla y(\Omega)$ is dependent on the choice of active constraints by the QP solver. Recall the assumption that active constraints remain active in the neighborhood of the current solution for $\mathbf{f}^*_\Omega$. As a consequence, if the normal force at a vertex $i$ is zero (i.e. lower bounds on $f_n^{i+}, f_n^{i-}$ are both active), then the partial derivatives $\partial f_n^{i-}/\partial\omega$ and $\partial f_n^{i+}/\partial\omega$ will also be zero. Structures are typically hyperstatic, meaning the solution satisfying static equilibrium is non-unique. Our quadratic energy function favors solutions distributing small tension forces over the structure, rather than clumping large tension forces in few locations.

Forces in typical structures have large ranging magnitude, e.g. blocks at the base carry the force of the entire building. Very large compression forces could overpower small tension forces, preventing the gradient from eliminating tension in the structure. We experimented with re-weighting forces at each interface according to expected compression magnitudes, but found that alternative strategies such as better step-size selection and L-BFGS were more effective for convergence.

Our method improves the feasibility of a structure within possible continuous changes in the geometry. There is no guarantee that a feasible solution exists.

## 10 Conclusions

We have introduced a technique that can guide the design of structurally stable buildings using new structural gradients. We have derived equations for the gradient of structural feasibility as a function of geometry and demonstrated accuracy compared to finite difference. The gradient can then be modified according to user constraints and objective, and applied to the current design, yielding a more sound configuration.

**Acknowledgements** Thanks to Sylvain Paris and reviewers of the MIT pre-deadline for helpful discussions, and Adriana Schulz for renderings. This work was partially supported by Shell. E. Whiting acknowledges the NSERC Canada PGS program. H. Shin acknowledges the Samsung Scholarship Foundation.

## References

ALLAIRE, G., JOUVE, F., AND TOADER, A.-M. 2004. Structural optimization using sensitivity analysis and a level set method. *Journal of Computational Physics 194*, 1, 363–393.

ALLEN, E., AND ZALEWSKI, W. 2009. *Form and Forces: Designing Efficient, Expressive Structures*. Wiley.

ATTAR, R., AISH, R., STAM, J., BRINSMEAD, D., TESSIER, A., GLUECK, M., AND KHAN, A. 2009. Physics-based generative design. In *Proc. CAAD Futures Conf.*, 231–244.

ATTAR, R., AISH, R., STAM, J., BRINSMEAD, D., TESSIER, A., GLUECK, M., AND KHAN, A. 2010. Embedded rationality: A unified simulation framework for interactive form-finding. *Intl. Journal of Architectural Computing 8*, 4, 399–418.

BERTSEKAS, D. P. 1995. *Nonlinear Programming*, second ed. Athena Scientific, ch. 2.1.1.

BERTSIMAS, D., AND TSITSIKLIS, J. N. 1997. *Introduction to Linear Optimization*. Athena Scientific.

BLOCK, P., AND OCHSENDORF, J. 2007. Thrust network analysis: a new methodology for three-dimensional equilibrium. *Journal Intl. Assoc. for Shell and Spatial Structures 48*, 3, 167–173.

DELFOUR, M. C., AND ZOLÉSIO, J.-P. 2001. *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*. SIAM.

EIGENSATZ, M., KILIAN, M., SCHIFTNER, A., MITRA, N. J., POTTMANN, H., AND PAULY, M. 2010. Paneling architectural freeform surfaces. *ACM Trans. Graph. 29*, 4, 45:1–45:10.

FLETCHER, R. 1987. *Practical methods of optimization*, second ed. John Wiley & Sons, New York.

GAL, R., SORKINE, O., MITRA, N. J., AND COHEN-OR, D. 2009. iWIRES: An analyze-and-edit approach to shape manipulation. *ACM Trans. Graph. 28*, 3, 33:1–33:10.

HARADA, M., WITKIN, A., AND BARAFF, D. 1995. Interactive physically-based manipulation of discrete/continuous models. In *Proc. SIGGRAPH*, ACM, 199–208.

HART, J. C., BAKER, B., AND MICHAELRAJ, J. 2003. Structural simulation of tree growth and response. *The Visual Computer 19*, 2-3, 151–163.

KILIAN, M., FLÖRY, S., CHEN, Z., MITRA, N. J., SHEFFER, A., AND POTTMANN, H. 2008. Curved folding. *ACM Trans. Graph. 27*, 3, 75:1–75:9.

LAPORTE, E., AND LE TALLEC, P. 2003. *Numerical Methods in Sensitivity Analysis and Shape Optimization*. Birkhäuser.

LI, Y., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2004. Lazy snapping. *ACM Trans. Graph. 23*, 3, 303–308.

LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph. 25*, 3, 681–689.

LIVESLEY, R. K. 1978. Limit analysis of structures formed from rigid blocks. *International Journal for Numerical Methods in Engineering 12*, 1853–1871.

LIVESLEY, R. K. 1992. A computational model for the limit analysis of three-dimensional masonry structures. *Meccanica 27*, 3, 161–172.

MARTIN, S., KAUFMANN, P., BOTSCH, M., GRINSPUN, E., AND GROSS, M. 2010. Unified simulation of elastic rods, shells, and solids. *ACM Trans. Graph. 29*, 4, 39:1–39:10.

POTTMANN, H., LIU, Y., WALLNER, J., BOBENKO, A., AND WANG, W. 2007. Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graph. 26*, 3, 65.

POTTMANN, H., SCHIFTNER, A., BO, P., SCHMIEDHOFER, H., WANG, W., BALDASSINI, N., AND WALLNER, J. 2008. Freeform surfaces from single curved panels. *ACM Trans. Graph. 27*, 3, 76:1–76:10.

SCHIFTNER, A., AND BALZER, J. 2010. Statics-sensitive layout of planar quadrilateral meshes. In *Advances in Architectural Geometry*, Springer, 221–236.

SHEA, K. 2000. EifForm: a generative structural design system. In *Proc. ACSA Technology Conference: The Intersection of Design and Technology*, 87–92.

SHI, X., ZHOU, K., TONG, Y., DESBRUN, M., BAO, H., AND GUO, B. 2007. Mesh puppetry: cascading optimization of mesh deformation with inverse kinematics. *ACM Trans. Graph. 26*, 3, 81.

SMITH, J., HODGINS, J. K., OPPENHEIM, I., AND WITKIN, A. 2002. Creating models of truss structures with optimization. *ACM Trans. Graph. 21*, 3, 295–301.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Proc. Eurographics Symposium on Geometry Processing*, 109–116.

TERZOPOULOS, D., AND FLEISCHER, K. 1988. Modeling inelastic deformation: viscolelasticity, plasticity, fracture. *SIGGRAPH Comput. Graph. 22*, 4, 269–278.

TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. *SIGGRAPH Comput. Graph. 21*, 4, 205–214.

TREFETHEN, L. N., AND BAU, D. 1997. *Numerical Linear Algebra*. SIAM, ch. 19.

VOUGA, E., HÖBINGER, M., WALLNER, J., AND POTTMANN, H. 2012. Design of self-supporting surfaces. *ACM Trans. Graph. 31*, 4, 87:1–87:11.

WELCH, W., AND WITKIN, A. 1992. Variational surface modeling. *SIGGRAPH Computer Graphics 26*, 2, 157–166.

WHITING, E., OCHSENDORF, J., AND DURAND, F. 2009. Procedural modeling of structurally-sound masonry buildings. *ACM Trans. Graph. 28*, 5, 112:1–112:9.

XU, W., WANG, J., YIN, K., ZHOU, K., VAN DE PANNE, M., CHEN, F., AND GUO, B. 2009. Joint-aware manipulation of deformable models. *ACM Trans. Graph. 28*, 3, 35:1–35:9.