



Three-dimensional funicular analysis of masonry vaults

P. Block*, L. Lachauer

Institute of Technology in Architecture, ETH Zurich, 8093 Zurich, Switzerland



ARTICLE INFO

Article history:

Received 31 August 2013

Received in revised form

28 November 2013

Accepted 30 November 2013

Available online 9 December 2013

Keywords:

Unreinforced masonry vaults

Limit analysis

Equilibrium analysis

Best-fit thrust network

Thrust Network Analysis

ABSTRACT

This paper introduces a new computational equilibrium analysis method for unreinforced masonry vaults, which extends Thrust Network Analysis using structural matrix analysis and efficient optimization strategies. By identifying independent states of equilibrium of funicular networks with any topology and fixed plan geometry, a robust and efficient solving algorithm is presented that allows practical limit analysis of (historic) masonry vaults with complex geometry under parallel loading conditions by fitting a provided geometrical target, e.g. the vault's mid surface. The presented framework can be seen as a strong foundation for practical equilibrium analysis of vaulted masonry.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Understanding the stability, or equilibrium, of unreinforced masonry is of primary concern (Heyman, 1995; Ochsendorf, 2002). The importance of equilibrium methods for the analysis of masonry structures, framed in an extensive historical overview, is provided and argued very clearly by Huerta (Huerta, 2001, 2004, 2008). This paper presents an important further development of equilibrium methods for three-dimensional problems based on Heyman's classic limit analysis that uses a no-tension model for masonry (Heyman, 1966). It continues work on discrete compressive force networks, called *thrust networks* by Block (Block and Ochsendorf, 2007; Block, 2009) and Fraternali (Fraternali and Rocchetta, 2002; Fraternali, 2010), solving the challenges in the former on control mechanisms and solving algorithm, and in the latter with respect to singularities in the boundary conditions or the discretized equilibrium surfaces. Fraternali (2010) gives a clear state of the art of the relevant work of the limit analysis framework that served as a base for both researches. Bigoni and Noselli (2010a,b) recently confirmed by means of photoelastic experiments that the statics of dry masonry structures is governed by the formation of localized force networks. Many other analysis models have been proposed, and for an exhaustive and general overview of the state of the art of approaches to structural analysis of historic masonry constructions, we also refer to Roca et al. (2010).

Based on Block (2009) and Fraternali (2010), Vouga et al. (2012) translated the thrust network approach (TNA) to computational geometry, proposing faster and more robust algorithms than presented by Block and Lachauer (2011) to compute equilibrium solutions that map a given target surface. The solving algorithms for this problem have been further improved by Liu et al. (2013) and de Goes et al. (2013) for triangulated meshes. Panozzo et al. (2013) provided an extension to arbitrary topologies.

This paper focuses on providing practical and structural insight to funicular networks with arbitrary topology by identifying and visualizing their force dependencies, or degrees of freedom (Block and Lachauer, 2011). This structural insight, the key difference to previous research, is furthermore used to develop an elegant solving algorithm to find a solution mapping a given height field best. If this height field is the mid surface of a non-cracked vault, than this objective produces the 3D equivalent of Wrinkler's "elastic" thrust line (Kurrer, 2008) for the chosen discretization. This equilibrium solution represents an admissible stress state for the vault and furthermore provides a good lower bound on the *geometric safety factor* (GSF) of the vault for uniformly distributed loading cases. The geometric safety factor is defined as the ratio between the thinnest possible vault geometry enveloping a thrust network in equilibrium with these given loads, and the actual vault geometry (Heyman, 1982; O'Dwyer, 1999). A particular objective of this paper is to serve as foundation for an implementation of a robust, flexible and intuitive 3D equilibrium analysis framework for assessing vaulted masonry. Block and Lachauer (2014) give applications for the assessment of historic structures of the algorithms presented in this paper.

* Corresponding author. Tel.: +41 44 633 68 44.

E-mail addresses: pblock@ethz.ch (P. Block), ilorenz@ethz.ch (L. Lachauer).

This paper is structured as follows: in [Section 2](#), the Force Density Method (FDM) is adapted for the use of limit analysis for masonry, using the core concepts of TNA and insights from matrix analysis of structural frames; in [Section 3](#), the computational setup is illustrated, explaining the nonlinear search algorithms in detail; in [Section 4](#), the method is applied to the analysis of historic and contemporary vaults, demonstrating the power and potential of the presented approach; and in [Section 5](#), the results of this paper are discussed and future work identified.

2. Equilibrium of funicular networks

To extend two-dimensional limit analysis using thrust lines ([Heyman, 1995](#); [Block et al., 2006](#)) to three-dimensional problems, additional constraints of compression are added to a general three-dimensional force network.

2.1. General force networks

Looking at a general network of m bars, called *branches*, pinned in $n = n_i + n_b$ nodes, with n_i inner, or *free*, nodes, and n_b boundary, or *fixed*, nodes, its $(3n_i)$ equations of equilibrium can be written in matrix form:

$$\mathbf{C}_i^T \mathbf{U} \mathbf{q} = \mathbf{p}_x \quad (1a)$$

$$\mathbf{C}_i^T \mathbf{V} \mathbf{q} = \mathbf{p}_y \quad (1b)$$

$$\mathbf{C}_i^T \mathbf{W} \mathbf{q} = \mathbf{p}_z \quad (1c)$$

in which \mathbf{C}_i^T is the transpose of \mathbf{C}_i , the sub-matrix consisting of the n_i columns corresponding to the free nodes of the $(m \times n)$ branch-node matrix of the network, \mathbf{C} ([Schek, 1974](#)). \mathbf{p}_x , \mathbf{p}_y , \mathbf{p}_z are the component vectors of length n_i of the loads \mathbf{p} applied to the free nodes. \mathbf{U} , \mathbf{V} , \mathbf{W} are the diagonalized matrices of the (branch) coordinate difference vectors of size m , \mathbf{u} , \mathbf{v} , \mathbf{w} , which are defined as:

$$\mathbf{u} = \mathbf{Cx} = \mathbf{C}_i \mathbf{x}_i + \mathbf{C}_b \mathbf{x}_b \quad (2a)$$

$$\mathbf{v} = \mathbf{Cy} = \mathbf{C}_i \mathbf{y}_i + \mathbf{C}_b \mathbf{y}_b \quad (2b)$$

$$\mathbf{w} = \mathbf{Cz} = \mathbf{C}_i \mathbf{z}_i + \mathbf{C}_b \mathbf{z}_b \quad (2c)$$

with \mathbf{x} , \mathbf{y} , \mathbf{z} the nodal coordinate vectors of length n . \mathbf{q} in [\(1a–1c\)](#) is the vector of size m listing the *force densities* ([Schek, 1974](#)), also called *tension coefficients* ([Pellegrino and Calladine, 1986](#)) of the branches in the network. These are defined as the ratio of the axial branch forces, \mathbf{s} , and the 3D lengths of the branches, \mathbf{l} , or equivalently, as the ratio of the horizontal components of the branch forces, i.e. the thrusts, \mathbf{s}_H , and the horizontal lengths of the branches, \mathbf{l}_H :

$$\mathbf{q} = \mathbf{L}^{-1} \mathbf{s} = \mathbf{L}_H^{-1} \mathbf{s}_H \quad (3)$$

in which \mathbf{L}_H and \mathbf{L} are the diagonalized matrices of \mathbf{l}_H and \mathbf{l} , respectively.

The equilibrium equations [\(1a–1c\)](#) are bilinear in \mathbf{u} , \mathbf{v} , \mathbf{w} and \mathbf{q} . To linearize these equations, the Force Density Method (FDM) prescribes values to \mathbf{q} ([Linkwitz and Schek, 1971](#); [Schek, 1974](#)):

$$\mathbf{C}_i^T \mathbf{Q} \mathbf{C} \mathbf{x} = \mathbf{p}_x \quad (4a)$$

$$\mathbf{C}_i^T \mathbf{Q} \mathbf{C} \mathbf{y} = \mathbf{p}_y \quad (4b)$$

$$\mathbf{C}_i^T \mathbf{Q} \mathbf{C} \mathbf{z} = \mathbf{p}_z \quad (4c)$$

which are thus only linear in \mathbf{x} , \mathbf{y} , \mathbf{z} . Note that for Eqs. [\(4a–4c\)](#) to be in equilibrium, the \mathbf{x} , \mathbf{y} will not necessarily stay fixed, i.e. will not be the ones provided as input.

2.2. Thrust networks

Because most (historic) masonry vaults have a heavy, dominant self-weight, it is often sufficient for the static analysis to consider vertical loads. It is furthermore often meaningful to keep the horizontal projection of the laid-out network fixed during the analysis, for example, to follow ribs in the structures.

Enforcing that the force network stays fixed in plan (\mathbf{x} , \mathbf{y} , and thus also \mathbf{u} , \mathbf{v} , are known), for vertical loading ($\mathbf{p}_x = \mathbf{p}_y = 0$), the horizontal equilibrium Eqs. [\(1a\)](#) and [\(1b\)](#) can be written as

$$\mathbf{C}_i^T \mathbf{U} \mathbf{q} = \mathbf{0} \quad (5a)$$

$$\mathbf{C}_i^T \mathbf{V} \mathbf{q} = \mathbf{0} \quad (5b)$$

which is a linear system of equations in \mathbf{q} . Solving [\(5\)](#) gives force densities \mathbf{q} that result in a horizontal equilibrium of a force network which has the provided horizontal projection. These then allow linearizing the vertical equilibrium Eq. [\(4c\)](#), and the nodal heights \mathbf{z}_i of the free nodes can directly be computed, because as \mathbf{x} , \mathbf{y} is kept fixed, \mathbf{p}_z can be computed for all nodes and is thus known:

$$\mathbf{z}_i = \mathbf{D}_i^{-1} (\mathbf{p}_z - \mathbf{D}_b \mathbf{z}_b) \quad (6)$$

with the $(n_i \times n_i)$ matrix $\mathbf{D}_i = \mathbf{C}_i^T \mathbf{Q} \mathbf{C}_i$ and the $(n_i \times n_b)$ matrix $\mathbf{D}_b = \mathbf{C}_i^T \mathbf{Q} \mathbf{C}_b$. If additionally all \mathbf{q} are positive, then the resulting solution from [\(6\)](#) is in compression-only.

For most network topologies, Eq. [\(5\)](#) does not have a unique solution, and not all combinations of \mathbf{q} will be valid solutions, or in other words, not all force densities are free variables. To search for thrust networks G^t , with a defined horizontal projection, called the *primal grid*, Γ , that have certain properties, or optimize particular objectives, two particular challenges need to be solved:

- Identify a set of branches, whose force densities, \mathbf{q}_{id} , can be chosen independently such that the resulting set of $\mathbf{q}(\mathbf{q}_{id})$ satisfies Eq. [\(5\)](#) for any choice of them. This will be addressed in the [Section 2.3](#).
- Correlate these \mathbf{q}_{id} , the variables of the equilibrium problem, to the unknown nodal heights \mathbf{z}_i of the free nodes, and relate them to the chosen objective to be optimized. This will be addressed in [Section 3](#).

2.3. Static indeterminacy of thrust networks

The problem of dealing with the degrees of indeterminacy of thrust networks with a given projection, i.e. understanding the solution space of [\(5\)](#), will be addressed in this section.

In the case of vertical loading, these loads do not appear in the horizontal equilibrium of G . The horizontal equilibrium of the thrust network G can thus also be seen as a specific in-plane equilibrium of G_H , which is defined as the planar bar-node structure coinciding with the horizontal projection of G . Inversely, each in-plane equilibrium of G_H can be interpreted as a possible horizontal equilibrium of a yet-unknown G , again for all loads vertical. For a given set of applied loads \mathbf{p} , each such in-plane equilibrium of G_H corresponds to a unique G , if its boundary conditions are given, i.e. the heights of the fixed boundary nodes, \mathbf{z}_b .

Note that the TNA framework requires parallel loading cases, of which vertical loads are a special case ([Block, 2009](#)). In the case of vertical loading, one can simply refer to the vertical and horizontal equilibrium, instead of the equilibrium in the direction of the applied (parallel) loading and the equilibrium in a plane perpendicular to them in the case of parallel loading. Even though the approach is thus more general, for simplicity of description and notation, the case of vertical loading is considered in this paper.

Noticing that G_H is unloaded (5), the different in-plane equilibria of the 2D bar-node structure are its different *states of self-stress*. These are indeed defined as distributions of equilibrating branch forces that keep the geometry of the unloaded structure unchanged. The number of states of self-stress of G_H thus represents the degree of structural indeterminacy of the 3D equilibrium network G with provided primal grid Γ . Pellegrino and Calladine (1986) provided a matrix analysis algorithm to identify the states of self-stress (and inextensional mechanisms) in frameworks. They explained that the branches associated to the *nullspace* of the equilibrium matrix \mathbf{A} represent a possible set of branches whose force densities \mathbf{q}_{id} can be chosen independently, which will be referred to as *independent branches*.

From (5), the $(2n_i \times m)$ equilibrium matrix \mathbf{A} is obtained by writing the nodal equilibrium equations in the following form

$$\mathbf{A}\mathbf{q} = \mathbf{p}, \mathbf{A} = \begin{bmatrix} \mathbf{C}_i \mathbf{U} \\ \mathbf{C}_i \mathbf{V} \end{bmatrix} \quad (7)$$

The dimension k of the nullspace of \mathbf{A} is $m - r_A$, with r_A the rank of \mathbf{A} . The identification of a base of the nullspace can be done using Gauss–Jordan Elimination (GJE) (Pellegrino and Calladine, 1986). This base provides a possible set of independent branches. With each choice of force densities \mathbf{q}_{id} for these branches, a unique state of self-stress corresponds, i.e. an equilibrium which does not change the geometry of the unloaded 2D framework G_H , or equivalently keeps the plan layout, Γ , of G fixed.

Having identified the independent branches and after reordering the equilibrium matrix \mathbf{A} and the force density vector \mathbf{q} such that the last k elements correspond to these branches, the full set of force densities \mathbf{q} corresponding to that state of self-stress is then found as

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_d \\ \mathbf{q}_{id} \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_d^{-1} \mathbf{A}_{id} \\ \mathbf{I}_k \end{bmatrix} \mathbf{q}_{id} = \mathbf{K} \mathbf{q}_{id} \quad (8)$$

in which the $(2n_i \times m - k)$ matrix \mathbf{A}_d and the $(2n_i \times k)$ matrix \mathbf{A}_{id} are the submatrices of \mathbf{A} corresponding to the dependent and independent branches, and their respective force densities, \mathbf{q}_d and \mathbf{q}_{id} , and \mathbf{I}_k the identity matrix of size k . Except for axisymmetrical networks, the $(2n_i \times m - k)$ matrix \mathbf{A}_d is square and non-singular, and thus invertible. A more general and robust formulation for the dependent force densities \mathbf{q}_d is

$$\mathbf{q}_d = -\mathbf{A}_d^\dagger \mathbf{A}_{id} \mathbf{q}_{id} \quad (9)$$

with \mathbf{A}_d^\dagger the *generalized inverse* or *Moore–Penrose pseudo-inverse* of \mathbf{A}_d .

In this paper, positive axial branch forces, \mathbf{s} , are considered compressive. From (3), it is thus clear that all \mathbf{q} need to be constrained to be positive in order to guarantee that they correspond to a compression-only state of self-stress for G_H . There are some requirements on the topology and geometry of G_H and Γ , but the reader is referred to Ash et al. (1988) and Block (2009) for these.

2.4. Relation to reciprocal diagrams

In Thrust Network Analysis (TNA) compression solutions were guaranteed by constructing allowed compression reciprocal force diagrams (Block, 2009). The geometrical information of these *reciprocal grids*, Γ^* , which described possible horizontal equilibria of G , allowed linearizing Eqs. (1a–1c) and (4) by measuring the lengths of the corresponding reciprocal branches, \mathbf{l}_H^* :

$$\mathbf{q} = \mathbf{L}_H^{-1} \mathbf{s}_H = \mathbf{L}_H^{-1} \mathbf{l}_H^* \quad (10)$$

in which \mathbf{L}_H is the diagonalized matrix of \mathbf{l}_H , listing the lengths of the branches of the primal grid Γ . As shown in Block (2009) and Vouga et al. (2012), these reciprocal branch lengths are furthermore

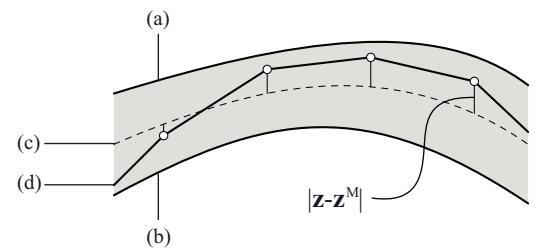


Fig. 1. The height deviation $|z_i - z_i^M|$ of the thrust network from the middle surface of the vault: (a) the vault's intrados, (b) extrados, (c) its mid surface, and (d) the thrust network.

proportional to the dihedral angles along the corresponding edges of the polyhedral stress functions (Fraterno et al., 2002; Fraterno, 2010).

3. Computational algorithm

For three-dimensional networks, obtaining equilibrium solutions for a given masonry vault with complex geometry, i.e. finding a compressive solution in equilibrium with the given loads that stays within the vault's section (Heyman, 1995), is not a straightforward task, because of the high degree of static indeterminacy of such models. Put simply, rather than having one horizontal thrust to consider, as in planar structures such as an arch, these networks have highly dependent combinations of thrusts in their elements. It is important to control all these parameters to perform an effective search.

The optimization algorithms presented in this section extend the Force Density Method (Schek, 1974) using new insights provided by Thrust Network Analysis (TNA) (Block, 2009) and matrix analysis of structures (Pellegrino and Calladine, 1986). Based on the new approach for identifying the force dependencies between branches in equilibrium networks (Section 2.3), an efficient optimization routine will be presented that allows finding best-fit thrust networks with a prescribed horizontal projection and given loading to a target surface.

The problem is formulated as finding the least-squares “best-fit” thrust network to the mid surface of the vault (Fig. 1):

$$f(\mathbf{z}_i) = \|\mathbf{z}_i - \mathbf{z}_i^M\|^2 \quad (11)$$

with $\mathbf{z}_i^M = (\mathbf{z}_i^I + \mathbf{z}_i^E)/2$, \mathbf{z}_i^I and \mathbf{z}_i^E being the vault's intrados and extrados heights fields for the free nodes of the primal Γ . Additionally, one needs to enforce that all \mathbf{q} are positive to guarantee a compression-only solution.

The key steps in the nonlinear solving algorithm, also representing the parts of the software implementation, can be divided into three stages: pre-processing steps (see Section 3.1), the best-fit solving (see Section 3.2); and the computation and visualization of results.

3.1. Pre-processing steps

The input for the algorithm is a set of connected lines, drawn in plan, representing the force pattern, and two surfaces S^I and S^E , representing the intrados and extrados of the vault. In some cases, e.g. for design applications, it is more useful to directly provide S^M , the mid, or *target*, surface.

As Γ stays unchanged in the nonlinear optimization, the computations related to the topology and geometry of the primal grid can be performed separately first.

3.1.1. Construct data structure

From the line input, an efficient procedural algorithm has been implemented to construct the primal grid Γ as a directed graph, and its branch-node matrix \mathbf{C} , from which then also directly the dual branch-node matrix \mathbf{C}^* can be constructed (Block, 2009). Using Γ , \mathcal{S}^I and \mathcal{S}^E , the bounds \mathbf{z}_i^I and \mathbf{z}_i^E , and thus also \mathbf{z}_i^M , are computed.

3.1.2. Generate centroidal dual

The (centroidal) dual of the spatial network, which is obtained by projecting the nodes of Γ onto the target surface, defined as the mid surface of the vault, \mathbf{z}_i^M , is computed. This dual will be used to approximate the self-weight of the vault (see Section 3.1.3), and as a starting point for the nonlinear optimization (see Section 3.2.1). The $(n^* \times n)$ face-node matrix \mathbf{F} is introduced, which is obtained as:

$$F_{ij} = \begin{cases} 0 & \text{if } (|\mathbf{C}^*|^T |\mathbf{C}|)_{ij} = 0 \\ 1 & \text{otherwise} \end{cases} \quad (12)$$

Notice that the i th row of \mathbf{F} indicates the nodes around face i of the primal grid Γ and thrust network G and the j th column the faces around node j . The centroidal dual coordinates are then

$$\mathbf{x}_D = \mathbf{SF}^{-1}\mathbf{Fx}, \quad \mathbf{y}_D = \mathbf{SF}^{-1}\mathbf{Fy}, \quad \mathbf{z}_D = \mathbf{SF}^{-1}\mathbf{Fz}^M. \quad (13)$$

with \mathbf{SF} the diagonalized matrix of \mathbf{sf} , the vector of length n with as entries the number of faces around the primal nodes, which is obtained by adding up the elements of each column of \mathbf{SF} , i.e. $\mathbf{sf}_j = \sum_i F_{ij}$.

3.1.3. Generate loading

Since the optimization searches for the best-fit solution to a given target surface, a good approximation of the 3D tributary area for generating the lumped nodal loads is to take the dual 3D area associated to that node by connecting the centroids of the faces around that node. If the solution is far off from the target surface, then this calculation could be included in the parameter search instead, which would of course slow down the search algorithm significantly.

For this step, one can readily use the centroids computed in (13) to calculate the tributary areas, and with the thicknesses \mathbf{d} of the vault at each node, the approximate load \mathbf{p} due to the self-weight of the vault is found. Additional applied loading can also be added, as long as these are also vertical.

3.1.4. Identify degrees of freedom

With the matrix analysis given in Section 2.3, a possible set of independent branches is identified, whose force densities, \mathbf{q}_{id} , are the k independent parameters of the problem. Having identified these branches, it is practical to reorder the equilibrium matrix \mathbf{A} and the force density vector \mathbf{q} such that the last k elements correspond to these branches. The key matrices related to the input geometry and topology, such as \mathbf{A} , \mathbf{K} , \mathbf{D}_i^{-1} ...are computed in advance.

The GJE algorithm is computationally an expensive pre-processing step, and limits the size of the networks that can be handled. This upfront cost is acceptable though as it heavily reduces the number of variables of the problem (see examples in Section 4), allowing for an efficient and robust solving strategy for the constrained nonlinear problem described next.

3.2. Find best-fit solution

Finding the best-fit solution for a thrust network with fixed projection can, as mentioned above, be reduced to finding the set of \mathbf{q}_{id}

that minimizes the global objective (11). As these variables do not explicitly appear in the objective function, this problem is solved using a *black-box function* which allows describing the objective function implicitly as a function of the variables (Section 3.2.2). Even though this constrained least-squares problem is nonlinear and non-convex, it can be solved efficiently with a black-box solver that uses a quasi-Newton search method, and by providing the gradient vector ∇f of the objective function in closed form (Section 3.2.2). Because of the non-convexity of the problem, it is important to use a good starting point for the search (Section 3.2.1).

3.2.1. Generate starting point

Finding a starting point for the nonlinear optimization happens in two steps:

- generate a set of force densities for the independent branches, $\mathbf{q}_{id,0}$, that results in positive (proportional) force densities, \mathbf{q}_0 , and hence a compression solution; and
- scale these $\mathbf{q}_{id,0}$, using the vertical equilibrium equations, such that the best-fit thrust network is obtained for the distribution of the generated set, considering the loads \mathbf{p} and the target heights, \mathbf{z}^M .

A set of \mathbf{q} that satisfies $\mathbf{Aq}=0$, together with the additional inequality constraints that $\mathbf{q} \geq 0$, corresponds to an in-plane compression equilibrium of G_H , and thus also to a possible horizontal equilibrium of a compressive thrust network G . Using (8), such a set can be found with the following quadratic programming (QP) problem:

$$\begin{array}{ll} \min_{\mathbf{q}_{id}} & \|\mathbf{AKq}_{id}\|^2 + \|\mathbf{L}_H \mathbf{Kq}_{id}\|^2 \\ \text{subject to} & \mathbf{Kq}_{id} \geq \mathbf{0} \\ & b_1 = 1 \end{array} \quad (14)$$

The second term in the objective of (14) is a regularizer that minimizes the horizontal force components \mathbf{s}_H ($= \mathbf{l}_H^* = \mathbf{L}_H \mathbf{q}$), which we observed to generate symmetrical force density distributions for symmetrical primal grids, and thus also in reciprocal grids. The additional equality constraint in (14), which gives a value to one of the independent force densities, e.g. a value of 1 to the first one, has to be included to avoid the solution to collapse to the trivial solution, $\mathbf{q}_{id} = \mathbf{0}$. It effectively also removes the degree of freedom of the global scale of the force densities, respectively the resulting reciprocal.

It is typically useful to provide the “force densities” generated from the centroidal dual as a starting point for the QP problem in (14) as we observed that the optimization then converges faster. So from (13), this starting point is

$$\mathbf{q}_D = \mathbf{L}_H^{-1} \mathbf{l}_{D,H} \quad (15)$$

$$\text{with } \mathbf{l}_{D,H} = \sqrt{(\mathbf{Cx}_D)^2 + (\mathbf{Cy}_D)^2}.$$

Using (8) and the resulting $\mathbf{q}_{id,0}$ from (14), all \mathbf{q}_0 can directly be found. These are only defined up to a scale factor, i.e. one can have the same distribution of force densities, but then with a different magnitude that also satisfy the constraints. To have a good starting point for our nonlinear best-fit problem, the obtained \mathbf{q}_0 need to be scaled to fit the target surface as well as possible. This can be done using the following least-squares problem with linear constraints:

$$\begin{array}{ll} \min_{\mathbf{z}, r} & \|\mathbf{z} - \mathbf{z}^M\|^2 \\ \text{subject to} & \mathbf{D}_r \mathbf{z} - r \mathbf{p} = \mathbf{0} \\ & r > 0 \end{array} \quad (16)$$

with $\mathbf{D}_r = \mathbf{C}_i^T \mathbf{Q}_0 \mathbf{C}$. The optimization problem (16) thus provides the scale r of the compression force densities obtained in (14). The starting point for the nonlinear optimization, $\mathbf{q}_{id,1}$, is thus

$$\mathbf{q}_{id,1} = \left(\frac{1}{r} \right) \mathbf{q}_{id,0} \quad (17)$$

3.2.2. Solve nonlinear problem

The problem to be solved is

$$\begin{aligned} \min_{\mathbf{q}_{id}} \quad & \| \mathbf{z}_i(\mathbf{q}_{id}) - \mathbf{z}_i^M \|^2 \\ \text{subject to} \quad & \mathbf{D}(\mathbf{q}_{id})\mathbf{z}(\mathbf{q}_{id}) = \mathbf{p} \\ & \mathbf{K}\mathbf{q}_{id} \geq \mathbf{0} \end{aligned} \quad (18)$$

which cannot be solved directly, as the unknowns \mathbf{z}_i are not variables, but rather a function of \mathbf{q}_{id} , which do not appear in the objective function. This optimization is thus solved using a black-box function that computes the objective function $f(\mathbf{z}_i(\mathbf{q}(\mathbf{q}_{id})))$ in several steps:

$$\mathbf{q}_{id} \rightarrow \mathbf{q} \rightarrow \mathbf{z}_i \rightarrow f \quad (19)$$

where the first two steps are linear transformations, and the last step involves taking the norm. These steps, computed at each iteration, can be solved quickly. The black-box objective function is

$$f(\mathbf{q}_{id}) = \mathbf{z}_i^T \mathbf{z}_i - 2(\mathbf{z}_i^M)^T \mathbf{z}_i + (\mathbf{z}_i^M)^T \mathbf{z}_i^M \quad (20)$$

and the gradient vector of size k , $\nabla f(\mathbf{q}_{id})$, provided analytically using (Block, 2009; Van Mele and Block, 2011), is

$$\nabla f(\mathbf{q}_{id}) = -2(\mathbf{D}_i^{-1}(\mathbf{D}_i \mathbf{Z}_i + \mathbf{D}_b \mathbf{Z}_b) \mathbf{K})^T (\mathbf{z}_i - \mathbf{z}_i^M) \quad (21)$$

in which \mathbf{Z}_i and \mathbf{Z}_b are the diagonalized matrices of \mathbf{z}_i and \mathbf{z}_b , and using $\mathbf{z}_i = \mathbf{D}_i^{-1}(\mathbf{p}_z - \mathbf{D}_b \mathbf{z}_b)$ in both (20) and (21).

Having obtained the full set of balanced force densities \mathbf{q} of the resulting thrust network, and using that $\mathbf{u}^* = \mathbf{C}^* \mathbf{x}^*$ and $\mathbf{v}^* = \mathbf{C}^* \mathbf{y}^*$, the dual coordinate vectors \mathbf{x}^* and \mathbf{y}^* are obtained by solving the following linear problem:

$$\begin{aligned} (\mathbf{C}^{*T} \mathbf{C}^*) \mathbf{x}^* &= \mathbf{C}^{*T} \mathbf{U} \mathbf{q}, \\ (\mathbf{C}^{*T} \mathbf{C}^*) \mathbf{y}^* &= \mathbf{C}^{*T} \mathbf{V} \mathbf{q}. \end{aligned} \quad (22)$$

Discrete thrust networks allow the user to clearly visualize the possible force paths in the vault, and by using reciprocal diagrams, to determine a satisfactory distribution of internal forces. Discrete reciprocal force diagrams are graphical and intuitive, and their visual nature allows a graphical verification of the process (Maxwell, 1870), e.g. a compression-only reciprocal grid is composed of only convex polygons. The solutions can be checked more easily than numerical or arithmetic methods, and the method is very transparent.

3.3. Prototype implementation

The computational approach, described in the previous sections, has been implemented in Matlab (The Mathworks, 2012a), using the Optimization Toolbox (The Mathworks, 2012b):

- quadprog and lsqlin are used for the quadratic programming (QP) and least-squares problems in Section 3.2.1; and
- fmincon's *interior-point method* is used for the nonlinear optimization problems in Section 3.2.2.

Custom scripts have been written for Rhinoceros (Rutten, 2007) to read in the line and surface input, and to visualize the results, both thrust network and reciprocal diagram.

4. Applications

This section shows and discusses results of the presented best-fit algorithm. The approach has relevant applications for equilibrium assessment of masonry vaults, but also design of funicular shells in general. For both categories of applications, vertical loading is appropriate: for the analysis of historic masonry vaults, their self-weight is typically the dominant loading, and in structural design of shells, only dead load is typically considered during form finding. In Section 4.1, the geometry of the sophisticated nave vaults of the Jerónimos monastery are used as case study of an equilibrium analysis of a complex vault; and, in Section 4.2, a thrust network is found that approximates a freeform input geometry, and the importance of force pattern topology is demonstrated for the example of a funicular shell with an applied point load, demonstrating a general strategy to assess the stability of funicular vaults under live load.

4.1. Complex Gothic vault geometry

This case study is based on the nave vaults of Jerónimos, finished around 1499 (Fig. 2a). From plans and sections, complemented with photographs and a photogrammetric survey (Monteiro Genin, 2001), a simplified 3D model of the intrados of the vaults was made. The rib pattern and the vault's stereotomy, then served as guide to draw a force pattern (Fig. 3a), defined such that it connects most stones perpendicular to their cut edges. It has 2676 branches and 1467 nodes. The problem is reduced to 208 variables, which are the independent branches, showing that this is a highly indeterminate problem. With the current prototypical MatLab implementation, the best-fitting thrust network (Fig. 3c) was found in less than half an hour, demonstrating that intricate network topologies and complex vault target geometries can be solved within reasonable time.

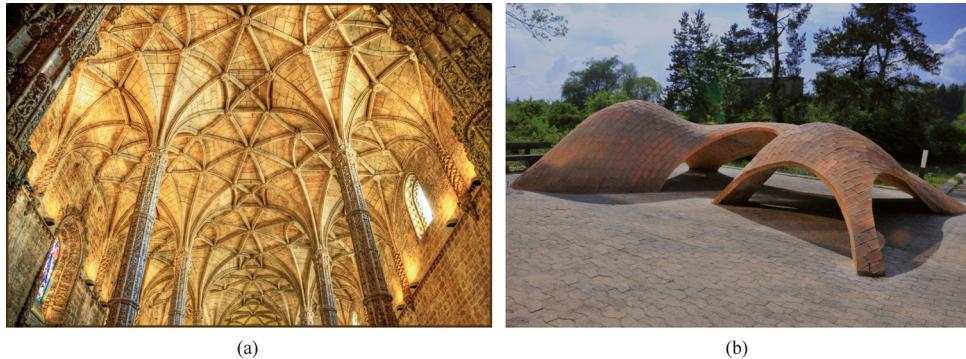
Assuming a constant thickness of 15 cm for the vaults, a lower bound on the geometric safety factor of only 0.34 is found, hence suggesting that the vaults would not be stable. The real vaults though have varied thicknesses, different levels of vault infill, and a grid of diaphragm walls on top of the vaults, which support the roof structure (Lourenço et al., 2007). Our approximate model furthermore did not include the significant deformations and possible cracks.

This example nonetheless serves as a demonstration of the potential and relevance of the presented algorithm as a base for advanced funicular analysis of complex masonry vaults. Examples of applications for the assessment of historic structures are presented in Block and Lachauer (2014).

4.2. Analysis of non-funicular loading

The prototype of a "free-form" unreinforced tile vault, built in 2011 at ETH Zurich, Switzerland (Davis et al., 2012) was used as starting geometry (2b). The best-fit algorithm was first tested to reproduce the same thrust network, respectively reciprocal (force) grid, for the primal grid used in the form finding process (red solution in Fig. 4c), taking into account the self-weight of a vault of uniform thickness. Next, a point load was added with a magnitude of a tenth of the total self-weight of the shell with an average thickness of 12 cm.

O'Dwyer (1999) showed very clearly, using a simple barrel vault with a point load, that the network topology has to be dependent on the loading case to be analyzed. Indeed, the pattern used to explain the stability, and in this case also to obtain the shape of the vaulted structure under self-weight, is not necessarily a good pattern to explain the vault's stability for a loading other than the design load. Using a similar approach, the original force pattern (Fig. 4a, top) is combined with a radial pattern for the point load (Fig. 4b, bottom).



(a)

(b)

Fig. 2. a) Nave vaults of the Monastery of Jerónimos, Lisbon, c. 1499 (Photo by Bert Kaufmann); and b) prototype of a “free-form” unreinforced tile vault, ETH Zurich, Switzerland, 2011 (Photo by Klemen Breitfuss).

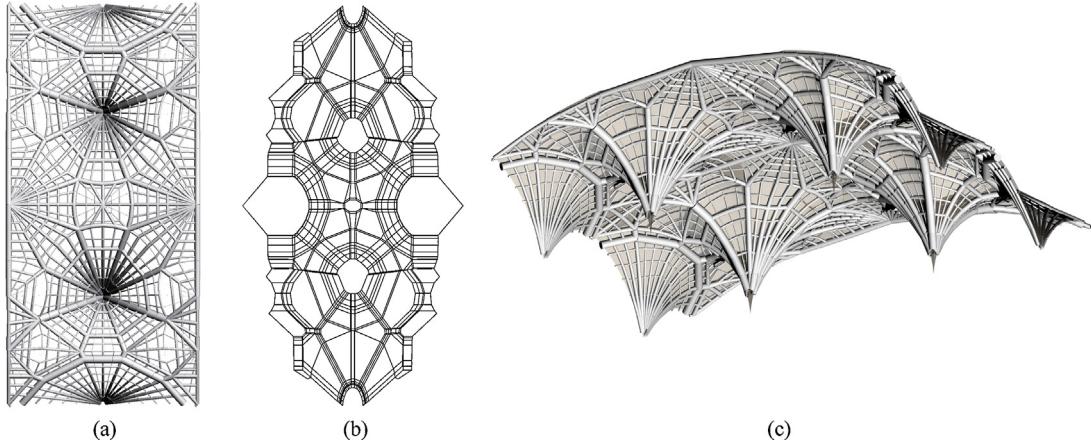


Fig. 3. (a) Primal grid, directly using the rib layout and stereotomy of the vaults; (b) the resulting best-fit reciprocal (force) grid; and (c) axonometry of the target surface, constructed from documentation, and the best-fit thrust network with pipes proportional to the axial forces in the branches.

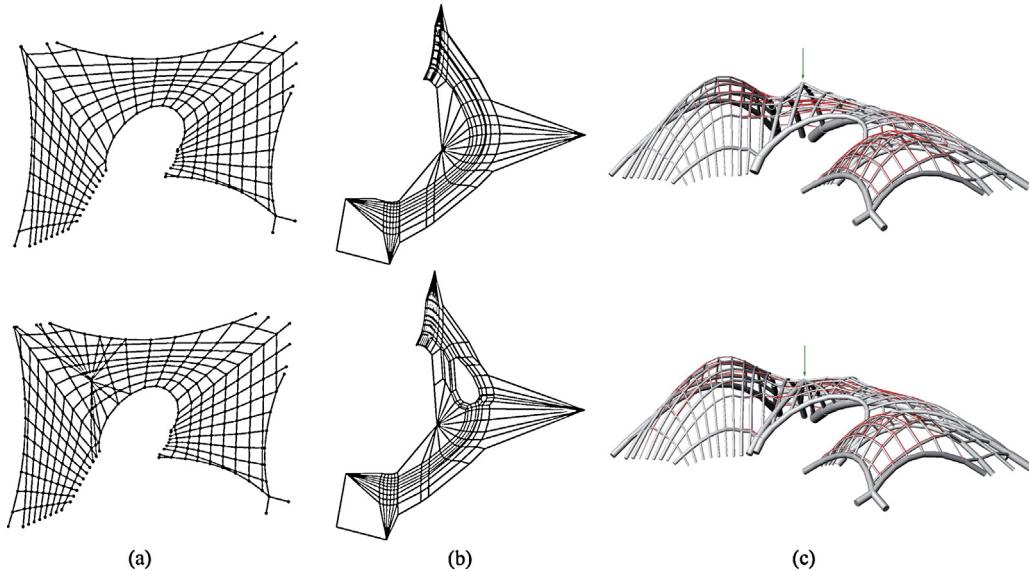


Fig. 4. For the original pattern used in the form finding (top), and with the overlaid radial pattern (bottom), respectively: (a) primal grid, and (b) reciprocal grid of (c) the resulting best-fit solution, considering both self-weight and point load.

Table 1
Results of the comparison between the best-fit of the combined uniform and point loading using (a) the original primal grid and (b) the primal grid with the overlaid radial pattern.

pattern	m	n_i	k	iter	$f/n_i [\text{mm}^2]$	$ \mathbf{z} - \mathbf{z}^M _{\max} [\text{mm}]$	$ \mathbf{z} - \mathbf{z}^M _{\text{aver}} [\text{mm}]$	$t [\text{s}]$	GSF_{lower}
Original	525	261	6	41	133	456	34	23.339	0.18
Overlaid	632	308	18	140	93	106	18	32.758	0.56

Note that only radials are chosen that locally do not go up from the application point. When applying the primal grid used for the form finding of the funicular vault, the resulting best-fit thrust network for the combined loading (piped solution in Fig. 4c, top) is far off of the target (red line solution in Fig. 4c, top). The best-fit solution for the superposed pattern is visibly much better (Fig. 4c, bottom), showing the importance of informed primals, as can also be seen by the improved fitting results in Table 1.

5. Discussion and future work

As shown through the example applications in the previous section, the presented method forms a good basis for limit analysis of masonry vaults with complex geometry. Especially the speed of finding equilibrium solutions for sophisticated vaults makes the approach appropriate and useful for practical applications, where time and budget constraints often do not permit the time-intensive modeling and analyzing of nonlinear finite element or discrete element models. The input is furthermore very straightforward, and allows to easily/manually incorporate singularities in both loading (point loads) and boundary conditions (non-properly supported edges). The visual and intuitive control and feedback of the presented framework furthermore make its results easy to interpret. Both latter points are additional key aspects for practical applications.

The method can furthermore be applied for design tasks by solving the inverse problem of fitting a compression-only solution to a target surface, which allows the optimization of the shape of an intended vault geometry (Block and Lachauer, 2011).

The presented algorithm is based on structural knowledge, more specifically, the identification of independent branches (these are the only branches whose force densities can be chosen freely) for a provided, fixed horizontal layout of a thrust network. This not only gives the user insights on the dependencies of a chosen pattern with respect to the corresponding, possible equilibria, particularly when visualized and thus made explicit (see e.g. Block and Lachauer, 2011). We argue that this aspect is important to contribute to the practicality of a discrete force network approach as it clarifies why certain force topologies can never work well to assess certain geometries. More importantly, it also allows a more direct and effective search. This is in contrast to other, brute-force methods, such as in Vouga et al. (2012), Liu et al. (2013) and Panozzo et al. (2013) that use two-step iterative algorithms to deal with these constraints and dependencies. As a consequence, these not always improve at each step, which is guaranteed in our method.

The example in Section 4.2 illustrated that the outcome of the method strongly depends on the layout (topology and geometry) of the primal grid. Therefore, a general strategy for generating appropriate primal grids to assist the analyst would be helpful. A first step in solving this problem is provided in Panozzo et al. (2013), but a general strategy should not only take into account the vault's geometry, but also the loading case, as well as properties of the measured, real geometry, including the vault's stereotomy or observed structural pathologies, such as cracks, hinge lines or holes. The low lower bounds on the GSF furthermore showed that the current objective does not provide good estimates for non-uniform live load cases, and that the objective function will need to be modified to measure deviations normal to the target surface, and furthermore minimize the maximum (absolute) deviation.

With the current set-up, open cracks could be taken into account by not having branches that cross this zone, in which case it is important to be able to keep the horizontal projection of the force pattern fixed, an inherent property of the presented approach. Cracks that form hinge lines could be taken into account by forcing the crossing branches to go through the contact zones by setting

bounds on the nodal heights of the thrust network along the cracks. Currently, these bounds cannot be enforced as hard constraints, but only as soft constraints by adding weighting factors to these nodes in the objective function (11). The integration of hard constraints, but also e.g. the extension to non-parallel loads will be addressed in future research.

6. Conclusion

This paper presented an algorithm for obtaining the best-fit thrust network for a provided, fixed horizontal projection to a target surface. Through examples, the power and potential was shown of the robust and efficient solving algorithm, which uses the dependencies of the force densities of such constrained networks. This new solving method allowed finding best-fit thrust networks for masonry vaults with complex geometry, under given vertical loading cases, providing admissible stress states for them. The presented framework, which extended Thrust Network Analysis using matrix analysis and efficient optimization strategies, can be seen as a strong foundation for fully three-dimensional and practical equilibrium analysis of vaulted masonry.

Acknowledgements

The authors wish to thank Dr. Tom Van Mele for the feedback and constructive discussions, and Dr. Daniele Panozzo for the help with optimizing the formulations of the solving algorithms.

References

- Ash, P., Bolker, E., Crapo, H., Whiteley, W., 1988. Covex polyhedra, dirichelet tessellations, spider webs. In: Senechal, M., Fleck, G. (Eds.), *Shaping Space: A Polyhedral Approach*. Birkhäuser, Boston, pp. 231–250.
- Bigoni, D., Noselli, G., 2010a. Localized stress percolation through dry masonry walls. Part I – experiments. *European Journal of Mechanics A/Solids* 29, 291–298.
- Bigoni, D., Noselli, G., 2010b. Localized stress percolation through dry masonry walls. Part II – modelling. *European Journal of Mechanics A/Solids* 29, 299–307.
- Block, P., 2009. *Thrust Network Analysis: Exploring Three-dimensional Equilibrium* (PhD Dissertation). Massachusetts Institute of Technology, Cambridge, USA.
- Block, P., Ciblac, T., Ochsendorf, J.A., 2006. Real-time limit analysis of vaulted masonry buildings. *Computers and Structures* 84, 1841–1852.
- Block, P., Lachauer, L., 2011. Closest-fit, compression-only solutions for free-form shells. In: *Proceedings of the IABSE-IASS Symposium 2011*, London, UK.
- Block, P., Lachauer, L., 2014. Three-dimensional equilibrium analysis of gothic masonry vaults. *International Journal of Architectural Heritage* 8, 1–24.
- Block, P., Ochsendorf, J., 2007. Thrust Network Analysis: a new methodology for three-dimensional equilibrium. *Journal of the International Association of Shell and Spatial Structures* 48, 167–173.
- Davis, L., Rippmann, M., Pawlofsky, T., Block, P., 2012. Innovative funicular tile vaulting: a prototype in Switzerland. *The Structural Engineer* 90, 46–56.
- Fraternali, F., 2010. A thrust network approach to the equilibrium problem of unreinforced masonry vaults via polyhedral stress functions. *Mechanics Research Communications* 37, 198–204.
- Fraternali, F., Angelillo, M., Fortunato, M., 2002. A lumped stress method for plane elastic problems and the discrete-continuum approximation. *International Journal of Solids and Structures* 39, 6211–6240.
- Fraternali, F., Rocchetta, G., 2002. Shape optimization of masonry vaults. In: *Proceedings of the Second International Conference on Advances in Structural Engineering and Mechanics, ASE'02*, Busan, Korea, pp. 1–8.
- de Goes, F., Alliez, P., Owhadi, H., Desbrun, M., 2013. On the equilibrium of simplicial masonry structures. *ACM Transactions on Graphics* 32, 93:1–93:10.
- Heyman, J., 1966. The stone skeleton. *International Journal of Solids and Structures* 2, 249–279.
- Heyman, J., 1982. *The Masonry Arch*. Ellis Horwood, Chichester.
- Heyman, J., 1995. *The Stone Skeleton: Structural Engineering of Masonry Architecture*. Cambridge University Press, Cambridge.
- Huerta, S., 2001. Mechanics of masonry vaults: the equilibrium approach. In: Lourenço, P.B., Roca, P. (Eds.), *Proceedings of Historical Constructions*. Guimarães, Portugal, pp. 47–69.
- Huerta, S., 2004. Arcos bóvedas y cúpulas. Geometría y equilibrio en el cálculo tradicional de estructuras de fábrica. Instituto Juan de Herrera, Madrid.
- Huerta, S., 2008. The analysis of masonry architecture: a historical approach. *Architectural Science Review* 51, 297–328.
- Kurrer, K.E., 2008. *The History and Theory of Structures. From Arch Analysis to Computational Mechanics*. Ernst & Sohn, Berlin.

- Linkwitz, K., Schek, H.J., 1971. Einige Bemerkungen von vorspannten Seilnetzkonstruktionen. *Ingenieur-Archiv* 40, 145–158.
- Liu, Y., Pan, H., Snyder, J., Wang, W., Guo, B., 2013. Computing self-supporting surfaces by regular triangulation. *ACM Transactions on Graphics* 32, 92:1–92:10.
- Lourenço, P., Krakowiak, K., Fernandes, F., Ramos, L., 2007. Failure analysis of Monastery of Jerónimos, Lisbon: How to learn from sophisticated numerical models. *Engineering Failure Analysis* 14, 280–300.
- Maxwell, J.C., 1870. On reciprocal figures, frames, and diagrams of forces. *Transactions of the Royal Society of Edinburgh* 26, 1–40.
- Monteiro Genin, S., 2001. The nave vault of the Hieronymites Monastery Church in Lisbon. In: Lourenco, P.B., Roca, P. (Eds.), *Proceedings of Historical Constructions. Guimarães, Portugal*, pp. 293–302.
- Ochsendorf, J.A., 2002. Collapse of Masonry Structures (PhD Dissertation). University of Cambridge, Cambridge, UK.
- O'Dwyer, D.W., 1999. Funicular analysis of masonry vaults. *Computers and Structures* 73, 187–197.
- Panozzo, D., Block, P., Sorkine-Hornung, O., 2013. Designing unreinforced masonry models. *ACM Transactions on Graphics* 32, 91:1–91:12.
- Pellegrino, S., Calladine, C.R., 1986. Matrix analysis of statically and kinematically indeterminate frameworks. *International Journal of Solids and Structures* 22, 409–428.
- Roca, P., Cervera, M., Gariup, G., Pela, L., 2010. Structural analysis of masonry historical constructions. Classical and advanced approaches. *Archives of Computational Methods in Engineering* 17, 299–325.
- Rutten, D., 2007. *RhinoScript™ 101 for Rhinoceros 4.0*. Robert McNeel & Associates.
- Schek, H.J., 1974. The force density method for form finding and computation of general networks. *Computer Methods in Applied Mechanics and Engineering* 3, 115–134.
- The Mathworks™, 1994–2012a. MATLAB®: The Language of Technical Computing [Computer Software]. <http://www.mathworks.com/products/matlab/>
- The Mathworks™, 1994–2012b. Optimization Toolbox™ for MATLAB®. <http://www.mathworks.com/products/optimization/>
- Van Mele, T., Block, P., 2011. Novel form finding method for fabric formwork for concrete shells. *Journal of the International Association of Shell and Spatial Structures* 52, 217–224.
- Vouga, E., Höbinger, M., Wallner, J., Pottmann, H., 2012. Design of self-supporting surfaces. *ACM Transactions on Graphics* 31, 87:1–87:11.