


Load-path optimisation of funicular networks

A. Liew  · D. Pagonakis · T. Van Mele · P. Block

Received: 14 February 2017 / Accepted: 17 June 2017
© Springer Science+Business Media B.V. 2017

Abstract This paper describes the use of load-path optimisation for discrete, doubly curved, compression-only structures, represented by thrust networks. The load-path of a thrust network is defined as the sum of the internal forces in the edges multiplied by their lengths. The presented approach allows for the finding of the funicular solution for a network layout defined in plan, that has the lowest volume for the given boundary conditions. The compression-only thrust networks are constructed with Thrust Network Analysis by assigning force densities to the network's independent edges. By defining a load-path function and deriving its associated gradient and Hessian functions, optimisation routines were used to find the optimum independent force densities that minimised the load-path function subject to compression-only constraints. A selection of example cases showed a dependence of the optimum load-path and force distribution on the network topology. Appropriate

selection of the network pattern encouraged the flow of compression forces by avoiding long network edges with high force densities. A general, non-orthogonal network example showed that structures of high network indeterminacy can be investigated both directly for weight minimisation, and for the understanding of efficient thrust network patterns within the structure.

Keywords Load-path optimisation · Minimum volume problem · Thrust Network Analysis · Structural optimisation · Thrust network topology

1 Introduction

This paper describes the use of load-path optimisation for discrete, doubly curved, compression-only structures, represented by funicular thrust networks. The load-path is the product of the force densities of the network edges and their lengths, whose minimisation is equivalent to minimising the volume of material for a given stress. It is implemented through Thrust Network Analysis (TNA), under a variety of assumptions: (1) fixed vertical loading, (2) coplanar fixed boundary vertices, and (3) fixed horizontal distribution of vertices. The goal is to compare the influence of the topology of the network on the performance of the structure measured in terms of minimal load-path, which is related to the volume of material for a given

A. Liew (✉) · T. Van Mele · P. Block
Institute of Technology in Architecture, ETH Zurich,
8093 Zurich, Switzerland
e-mail: liew@arch.ethz.ch

T. Van Mele
e-mail: van.mele@arch.ethz.ch

P. Block
e-mail: block@arch.ethz.ch

D. Pagonakis
Department of Mathematics, MIT, Cambridge, MA, USA
e-mail: dimpag@mit.edu

stress. Self-weight is not considered, with loading represented only by the fixed vertical loads.

The presented approach allows for the finding of the funicular solution for a network layout defined in plan, that is the minimal volume solution for the given boundary conditions and given stress. The minimal volume solution is found by optimising the statical indeterminacy of the structure through a load-path objective function. The solvers utilised in the optimisation process are based on function and function and gradient methods, with the Hessian also derived using the analytical load-path function and its gradient. The optimisation is constrained, because all force densities must be positive for a compression-only solution, and are further constrained to a given range or bounds.

To keep the horizontal projection of the network fixed (i.e. the x and y coordinates of the given vertices) only the independent edges are used as variables in the optimisation routines, which greatly reduces the number of degrees-of-freedom that are necessary for the optimisation process, and is therefore preferable to optimising for the force densities of all edges. Furthermore, by using only the independent edges, the predetermined layout of forces in the form diagram stays fixed, without needing further constraints on the optimisation process.

The structure of this paper is as follows. In Sect. 2, the concept of load-path is introduced, along with its key properties and the relationship to volume minimisation. In Sect. 3, the load-path function is expressed as a function of force densities, and a TNA load-path scale optimisation is performed. In Sect. 3.1, both continuous and discrete load-path optimisations are applied in the funicular parabolic arch as a validation of the discrete approach. In Sect. 3.2, a refined version of Maxwell's load-path formula is derived for TNA, and the desired fixed vertical projection load-path optimisation is formulated. In Sect. 3.3, the computational and mathematical setup is illustrated, explaining the constructed algorithms. In Sect. 4, several structural test cases on load-path optimisation are showcased. Finally, a discussion of the results in Sect. 5, demonstrating the potential and breadth of the presented approach in least volume structures.

2 Introduction to load-paths

In this section, the theoretical framework for load-paths is described. In using the term load-path, we are

describing the path that static external loads applied to a structure find their way to the supports through a truss or network-type structure. For a statically indeterminate structure with given external loads and support conditions, different internal load distributions can be found that lead to the structure being in static equilibrium. These internal force distributions are characterised with a load-path function, which is defined as the sum of the internal forces multiplied by the network's edge lengths, providing a scalar quantity that allows for the comparison of different force distributions to optimise their performance with respect to minimum weight.

Various research has been conducted in optimising structures in the literature. Structural optimisation with non-dimensional morphological indicators including strength, stiffness and weight, can be found in [1] with specific application to trusses in [2] and [3]. Also for trusses, [4, 5] and [6] showed that optimising for the minimal load-path results in the minimal volume solution for a given stress σ in all members, such that

$$\min \sum_i V_i = \min \sum_i A_i l_i = \min \frac{1}{\sigma} \sum_i |F_i| l_i, \quad (1)$$

in which V_i , A_i , l_i and F_i are the volume, cross-sectional area, length, tensile (negative) and compressive (positive) axial forces of member i , respectively.

Inspired by the theorem of [5, 7] continued to show that

$$\sum_i |F_i| l_i = \sum_j \vec{P}_j \cdot \vec{r}_j, \quad (2)$$

where $\vec{P}_j \cdot \vec{r}_j$ is the dot product of external forces and the vectors defined by the loads' application points. The external contribution delivered by a force corresponds to the dot product of the distance that the force travels with the magnitude of the force along the distance vector. This equation can further be split into the following components:

$$\left(\sum F l \right)_{\text{compression}} + \left(\sum F l \right)_{\text{tension}} = \left(\sum \vec{P} \cdot \vec{r} \right)_{\text{loads}} + \left(\sum \vec{P} \cdot \vec{r} \right)_{\text{reactions}}, \quad (3)$$

in which the internal contributions from the compression and tension members and the external contributions from the applied loads and the reactions are the

left-hand-side and right-hand-side, respectively. Further description and examples of the equations can be found in [8].

3 Load-paths applied to discrete thrust networks

Maxwell's formulae on load-paths can be generalised to arbitrary (vertical) load cases, and to three-dimensional networks of forces using the Thrust Network Analysis (TNA) framework [9–13]. TNA is an equilibrium method, which extends the Force Density Method for the specific case of vertical loading and networks with fixed horizontal projection. The method uses reciprocal force diagrams, derived from concepts based on graphic statics, to further control the distribution of horizontal thrusts in the compressive funicular solution. Thrust networks can be described as discrete compressive force networks that are in equilibrium with the applied loading.

Visualised in Fig. 1, TNA can be summarised as follows:

- In plan, a form Γ and force Γ^* diagram are defined, which represent the projected layout of the edges and the proportional equilibrated distribution of horizontal forces along the corresponding edges of the compressive thrust network G .

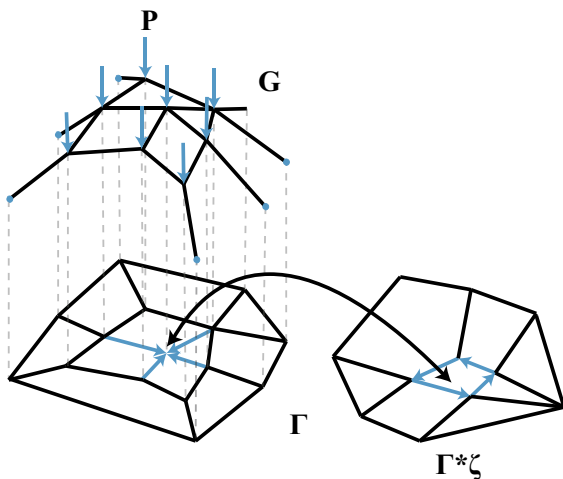


Fig. 1 Thrust Network Analysis: form diagram Γ , force diagram Γ^* , with given scale $1/r$, the reciprocal relation between one vertex in the form diagram and corresponding face in the force diagram, and the thrust network G for given support vertices z_b and loading p

- Form and force diagrams have a reciprocal relation [7], which results in nodal equilibrium in the former being represented by a closed polygon of force vectors in the latter.
- When constrained to vertical loading cases only, a free parameter is the overall scale $1/r$ of the force diagram Γ^* .

3.1 Load-path scale optimisation

3.1.1 Internal load-path formulation

Introducing force densities \mathbf{q} [14, 15], which are defined as

$$\mathbf{q} = \mathbf{L}^{-1} \mathbf{f}, \quad (4)$$

where \mathbf{f} and \mathbf{l} are thrust forces and three-dimensional lengths of the edges of thrust network G , and \mathbf{L} is the matrix obtained by diagonalising length vector \mathbf{l} .

In matrix notation, the internal load-path objective function in Equat. (1) can be rewritten as

$$f_i(r) = \mathbf{q}^T \mathbf{L}^T \mathbf{l}, \quad (5)$$

as $\mathbf{q}^T \mathbf{L}^T$ represents the force vector and \mathbf{l} the edge lengths vector. For simplicity, the notation $\mathbf{a}^2 = \mathbf{A}^T \mathbf{a}$ will be used in the rest of the paper, with \mathbf{A} as the diagonalised matrix of vector \mathbf{a} . Therefore

$$f_i(r) = \mathbf{q}^T \mathbf{l}^2. \quad (6)$$

As $\mathbf{L}^{-1} \mathbf{f} = \mathbf{L}_H^{-1} \mathbf{f}_H$, the force densities \mathbf{q} can be related to the lengths of the corresponding edges in the form and force diagrams, \mathbf{l}_H and \mathbf{l}_H^* respectively, as

$$\mathbf{q} = \frac{1}{r} (\mathbf{L}_H^{-1} \mathbf{l}_H^*). \quad (7)$$

As the interpretation of the reciprocal force diagram is that

$$\mathbf{f}_H = \frac{1}{r} \mathbf{l}_H^*, \quad (8)$$

with \mathbf{f}_H the horizontal force components of \mathbf{f} , and r the scaling factor of the form and force diagrams.

Introducing the connectivity matrix \mathbf{C} , and its sub-matrices \mathbf{C}_i and \mathbf{C}_b corresponding to the n_i internal vertices and n_b fixed boundary vertices, respectively, for vertical loads \mathbf{p}_z , the vertical equilibrium of vertices gives

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_i \\ \mathbf{z}_b \end{bmatrix} = \begin{bmatrix} \mathbf{D}_i^{-1}(\mathbf{p}_z - \mathbf{D}_b \mathbf{z}_b) \\ \mathbf{z}_b \end{bmatrix}, \quad (9)$$

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 all force densities \mathbf{q} are positive, then the resulting solution from Eq. (9) is compressive, which is guaranteed when using Eq. (7). From this solution, one can obtain the vertical coordinate differences $\mathbf{w} = \mathbf{C} \mathbf{z}$ for each edge.

One can find different funicular solutions based on different r values, so for instance with $r = 1$ one acquires the base solution \mathbf{z}^0 (and \mathbf{w}^0). The heights of the different funicular solutions are, due to the vertical-only loading constraint, inversely proportional to the scale $1/r$ of the horizontal forces [16], which is equivalent to the relationship between the horizontal thrust and the rise of a funicular arch for vertical loading cases (Fig. 2).

More specifically, from Eq. (7), $\mathbf{z} = r \mathbf{z}^0$, or

$$\mathbf{w} = r \mathbf{w}^0. \quad (10)$$

Using $\mathbf{I}^2 = \mathbf{I}_H^2 + \mathbf{w}^2$ and Eqs. (7) and (10), Eq. (6) can be rewritten as

$$f_i(r) = \mathbf{q}^T \mathbf{I}^2 = (\mathbf{L}_H^{-1} \mathbf{I}_H^*)^T \left(\frac{1}{r} \mathbf{I}_H^2 + r \mathbf{w}^{02} \right) \quad (11)$$

in which the scalar r is the only unknown.

The minimum of Eq. (11) is then directly obtained by solving

$$\begin{aligned} \frac{df_i(r)}{dr} &= (\mathbf{L}_H^{-1} \mathbf{I}_H^*)^T \left(-\frac{1}{r^2} \mathbf{I}_H^2 + \mathbf{w}^{02} \right) \\ &= -\frac{1}{r} \mathbf{q}^T \mathbf{I}_H^2 + r \mathbf{q}^T \mathbf{w}^{02} = 0, \end{aligned} \quad (12)$$

from which the following is obtained

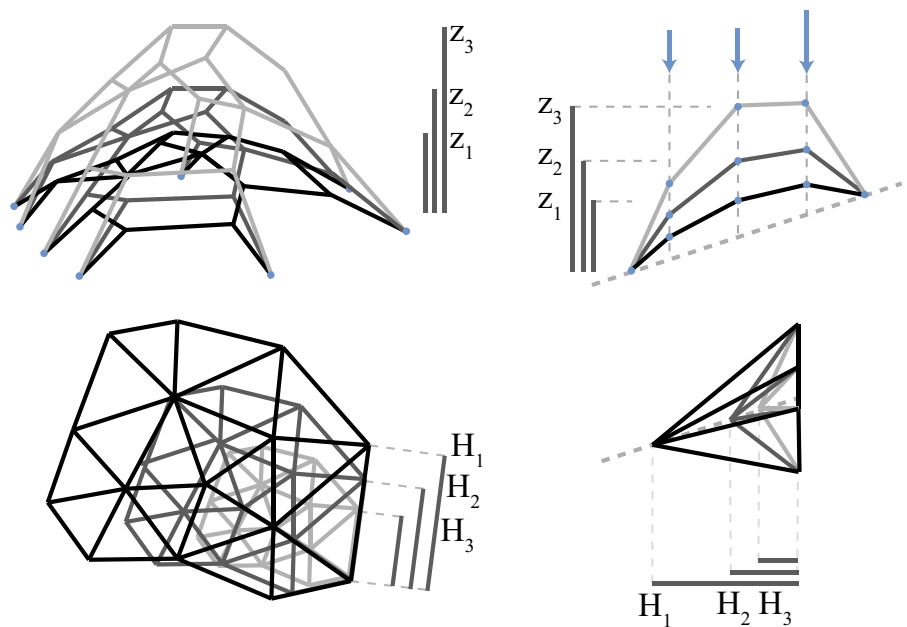
$$r_{\min} = \pm \sqrt{\frac{\mathbf{q}^T \mathbf{I}_H^2}{\mathbf{q}^T \mathbf{w}^{02}}} = \pm \sqrt{\frac{(\mathbf{I}_H^*)^T \mathbf{I}_H}{(\mathbf{L}_H^{-1} \mathbf{I}_H^*)^T \mathbf{w}^{02}}}. \quad (13)$$

The minimal-volume solution for the given loading and chosen form diagram (network layout defined in plan) and force diagram (proportional distribution of horizontal thrust) is then found using Equation (9) with $r = r_{\min}$. The minimum scale r_{\min} is useful for generating starting points in the general thrust network load-path optimisation. As a validation of the discrete scale optimisation, a load-path scale optimisation is demonstrated next in Sect. 3.1.2.

3.1.2 Validation: continuous and discrete parabolic arch

Using Maxwell's theorem (Eq. 2), it is possible to find the optimal-volume height of a parabolic compression arch. A parabola is the funicular shape for an applied

Fig. 2 The relation between (horizontal) thrust and funicular shape is similar for two and three-dimensional equilibrium structures. For vertical loading, the scale of the horizontal force diagram is inversely proportional to height [16]



load that is proportional to the horizontal projection, that is, for the uniformly distributed load w in Fig. 3. To validate the optimality of the scaling factor, through a discretisation of the parabolic arch, we perform load-path optimisation as a function of the scale factor to find the optimal height to span ratio. This ratio is then estimated using continuous load-path optimisation.

The symmetrically discretised parabola in Fig. 3 has been constructed with graphic statics [16]. From here, one can measure the following properties: $\mathbf{z}^0 = [1, 2.2, 2.6, 2.2, 1]^T$, $\mathbf{w}^0 = [1, 1.2, 0.4, -0.4, -1.2, -1]^T$; $\mathbf{l}_H = [1, 2, 2, 2, 1]^T$; and, because $R_H = 5$, $\mathbf{l}_H^* = [5, 5, 5, 5, 5]^T$. Using Equation (13), $r_{\min} = \frac{5}{3}$, and thus the ratio of the height h over the span L is $\frac{h}{L} = \frac{13}{30} \approx \frac{\sqrt{3}}{4}$.

Now to compare this result with continuous load-paths, the same shape of the parabolic arch in a continuous form is considered. The shape of the parabola can be written as a function of its rise h and span L :

$$y = 4h \left(\frac{x}{L} - \frac{x^2}{L^2} \right). \quad (14)$$

For a uniformly distributed load w , the vertical and horizontal reactions are

$$R_V = \frac{wL}{2}, \quad \text{and} \quad R_H = \frac{wL^2}{8h} \quad (15)$$

As the compression arch does not have tension members, Equation (3) can be reduced to

$$\left(\sum Fl \right)_{\text{compression}} = \left(\sum \vec{P} \cdot \vec{r} \right)_{\text{loads}} + \left(\sum \vec{P} \cdot \vec{r} \right)_{\text{reactions}}. \quad (16)$$

The right-hand side of Equation (16) is equivalent to the continuous summations

$$\begin{aligned} \left(\sum \vec{P} \cdot \vec{r} \right)_{\text{loads}} &= \int_0^L -w4h \left(\frac{x}{L} - \frac{x^2}{L^2} \right) dx \\ &= -4wh \left(\frac{x^2}{2L} - \frac{x^3}{3L^2} \right) \Big|_0^L = -\frac{2}{3}hwL, \end{aligned} \quad (17)$$

and

$$\left(\sum \vec{P} \cdot \vec{r} \right)_{\text{reactions}} = -R_H L = -\frac{wL^3}{8h}. \quad (18)$$

The total load-path $\sum Fl$ for the continuous parabolic arch is thus:

$$V = \frac{\sum Fl}{\sigma} = \frac{wL}{24\sigma h} (3L^2 + 16h^2). \quad (19)$$

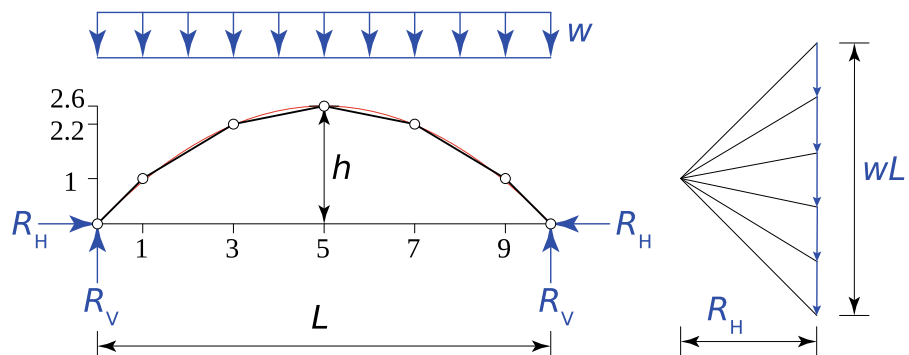
in which we note that for a parabolic arch, the external contribution of the applied loads is the applied load wL times the average height of the arch $\frac{2}{3}h$. This can in fact be generalised for any height difference between the fixed points of the parabolic arch, since the horizontal reaction force remains the same, the equation of the partial parabolic arch is again a skewed parabola and thus the external work $(\sum \vec{P} \cdot \vec{r})_{\text{loads}}$ is still the applied load times the average height of the arch.

The optimal height of the arch, i.e. the height at which the arch's volume in Eq. (19) is minimised, can be found by taking the derivative of the volume with respect to the height h :

$$\frac{d \sum Fl}{dh} = -\frac{wL^3}{8h^2} + \frac{2wL}{3} = 0, \quad (20)$$

and we get the optimal height-to-span-ratio

Fig. 3 Parameters of a parabolic arch under uniformly distributed load w : horizontal and vertical reaction forces, R_V and R_H , span L and rise h . The piece-wise linear approximation of the parabola was constructed using graphic statics



$$\frac{h}{L} = \pm \frac{\sqrt{3}}{4}, \quad (21)$$

matching the discretised parabola result. The positive and negative signs corresponds to the compressive and the hanging parabola.

3.2 Discrete thrust-network force distribution optimised

On a broader perspective, one can consider all force densities \mathbf{q} as parameters in the load-path optimisation, not just the overall scale of a chosen fixed force diagram. This means that the horizontal force distribution will no longer be kept fixed, but rather only the horizontal projection of the thrust network, i.e. the form diagram.

However for a fixed projection, these \mathbf{q} values cannot be chosen independently ([17, 18]):

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_d \\ \mathbf{q}_{id} \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_d^{-1} \mathbf{A}_{id} \\ \mathbf{I}_k \end{bmatrix} \mathbf{q}_{id} = \mathbf{K} \mathbf{q}_{id}, \quad (22)$$

in which the $(2n_i \times m - k)$ non-singular square matrix \mathbf{A}_d and the $(2n_i \times k)$ matrix \mathbf{A}_{id} are the sub-matrices of \mathbf{A} corresponding to the dependent and independent edges, with their respective force densities \mathbf{q}_d and \mathbf{q}_{id} , and the identity matrix \mathbf{I}_k of size k . Thus, for fixed horizontal projection, we want to find the independent edges' force densities \mathbf{q}_{id} for which the total load-path (and hence the total volume) is minimised, while equilibrium still holds.

The concept of independent edges can be best understood by inspecting the equilibrium of an unloaded, self-stressed, perfectly orthogonal grid. As the force in each of the sets of continuous edges is entirely independent from the force in any other continuous set, the state of equilibrium of the grid can be described determinately by independently choosing one force density per continuous set of edges. Clearly, there are infinitely many solutions, however, if other combinations of edge force densities are chosen, the geometry of the grid has to change to find a new state of equilibrium. This principle also applies to less regular grids, but then the relationship between the dependent and independent edges is less obvious.

Numerically, the independent edges of a form diagram can be identified as the non-pivoting columns of the reduced row echelon form (RREF) of the matrix describing its unloaded, self-stressed, horizontal

equilibrium. The RREF of a matrix can be obtained using Gauss Jordan Elimination. Note that if identification is not required, the number of independent edges can be determined using rank analysis of the equilibrium matrix using Singular Value Decomposition. Note that every choice of independent force densities corresponds to a self-stressed state of the form diagram, which is a state of equilibrium that keeps the geometry of the diagram fixed. However, not every choice of positive independent force densities corresponds to a full set of positive force densities, which is the requirement for a compression-only solution.

Equilibrium is ensured by

$$\mathbf{A} \mathbf{q} = \mathbf{p}, \quad \text{with} \quad \mathbf{A} = \begin{bmatrix} \mathbf{C}_i^T \mathbf{U} \\ \mathbf{C}_i^T \mathbf{V} \end{bmatrix}, \quad (23)$$

where \mathbf{U} and \mathbf{V} are the diagonal matrices of vectors \mathbf{u} and \mathbf{v} , which are the coordinate differences in the x and y directions for the vertices of each edge.

3.2.1 External load-path formulation

From Eq. (6), the internal load-path function was given as

$$f_i(\mathbf{q}_{id}) = \mathbf{q}^T \mathbf{I}^2, \quad (24)$$

with variables the independent force densities \mathbf{q}_{id} . The objective is to find the optimal \mathbf{q}_{id} set that minimises the load-path function, subject to funicular (compression-only) constraints for all edges of the thrust network.

By Maxwell's theorem Eq. (3), the internal load-path function is equivalent to the external load-path function. The external load-path formulation avoids the third order non-linearity of Eq. (24) by separating the load-path function into the load-path of the reaction forces and the load-path of the applied loads. The detailed derivation of the external load-path now follows.

$$\left(\sum F l \right)_{\text{compression}} = \left(\sum \vec{P} \cdot \vec{r} \right)_{\text{loads}} + \left(\sum \vec{P} \cdot \vec{r} \right)_{\text{reactions}} \quad (16)$$

The $\left(\sum \vec{P} \cdot \vec{r} \right)_{\text{loads}}$ term is the inner product of all applied vertical loads \mathbf{p}_z with the distance to any coplanar point of reference. A critical assumption is that all fixed vertices lie on a common plane. In this case, the external load-path takes the form

$$\left(\sum \vec{P} \cdot \vec{r}\right)_{\text{loads}} = \mathbf{p}_z^T \cdot \mathbf{z}_i, \quad (25)$$

as the only vector that survives the dot product is the vertical distance \mathbf{z}_i . Additionally, note that $(\sum \vec{P} \cdot \vec{r})_{\text{reactions}}$ is the sum of the reaction forces on the base dotted with the distance from each fixed vertex, n_b . From Eq. (23), the horizontal equilibrium takes the form

$$\mathbf{p}_x = \mathbf{C}_i^T \mathbf{Q} \mathbf{C}_x \mathbf{x}, \quad (26a)$$

$$\mathbf{p}_y = \mathbf{C}_i^T \mathbf{Q} \mathbf{C}_y \mathbf{y}. \quad (26b)$$

Thus, the total horizontal reaction forces are accordingly

$$\mathbf{C}_b^T \mathbf{Q} \mathbf{C}_x \mathbf{x} + \mathbf{C}_b^T \mathbf{Q} \mathbf{C}_y \mathbf{y}. \quad (27)$$

Note, that the vertical reaction forces dotted with the distances is zero, since we assume the base points are all at the same elevation. The distance vectors are \mathbf{x}_b and \mathbf{y}_b and so the total distance vector is $\vec{r}_{\text{reactions}} = \mathbf{x}_b + \mathbf{y}_b$, and the reactions' load-path is

$$\begin{aligned} \left(\sum \vec{P} \cdot \vec{r}\right)_{\text{reactions}} &= (\mathbf{p}_x + \mathbf{p}_y)^T (\mathbf{x}_b + \mathbf{y}_b) \\ &= (\mathbf{C}_b^T \mathbf{Q} \mathbf{C}_x \mathbf{x} + \mathbf{C}_b^T \mathbf{Q} \mathbf{C}_y \mathbf{y})^T (\mathbf{x}_b + \mathbf{y}_b) \\ &= (\mathbf{C}_b^T \mathbf{Q} \mathbf{C}_x \mathbf{x})^T \mathbf{x}_b + (\mathbf{C}_b^T \mathbf{Q} \mathbf{C}_y \mathbf{y})^T \mathbf{y}_b. \end{aligned} \quad (28)$$

From Eqs. (25) and (28), the total load-path function is

$$f_i = f_e(\mathbf{q}_{\text{id}}) = \mathbf{p}_z^T \mathbf{z}_i + (\mathbf{C}_b^T \mathbf{Q} \mathbf{C}_x \mathbf{x})^T \mathbf{x}_b + (\mathbf{C}_b^T \mathbf{Q} \mathbf{C}_y \mathbf{y})^T \mathbf{y}_b \quad (29)$$

From [17], the height of the internal vertices is given by

$$\mathbf{z}_i = \mathbf{D}_i^{-1} (\mathbf{p}_z - \mathbf{D}_b \mathbf{z}_b). \quad (30)$$

Inserting Eq. (30) into Eq. (29), gives the external load-path function

$$\begin{aligned} f_e(\mathbf{q}_{\text{id}}) &= \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{p}_z + \mathbf{x}^T \mathbf{C}^T \mathbf{Q} \mathbf{C}_b \mathbf{x}_b \\ &\quad + \mathbf{y}^T \mathbf{C}^T \mathbf{Q} \mathbf{C}_b \mathbf{y}_b - \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{D}_b \mathbf{z}_b. \end{aligned} \quad (31)$$

3.3 Computational algorithm

The three dimensional load-path optimisation conceptually works as follows. The inputs are the applied

vertical loads \mathbf{p}_z , the planar coordinates of all vertices \mathbf{x}, \mathbf{y} (fixed horizontal projection), and consequently matrices \mathbf{C}_i and \mathbf{C}_b . The objective function is the load-path function which is minimised subject to the constraint $q_i \geq 0, \forall q_i$. From the optimal distribution of force densities, one can compute \mathbf{z}_i , the vertical heights of all free vertices that compose the least-volume three-dimensional configuration of the structure.

3.3.1 Mathematical formulation

In mathematical form, the optimisation can be written as

$$\min_{\mathbf{q}_{\text{id}}} f(\mathbf{q}_{\text{id}}) \text{ s.t. } q_i \geq 0 \quad \forall i \in \{1, 2, \dots, m\} \quad (32)$$

Due to the nature of the load-path function, this is a positive semi-definite programming problem. However, by smoothness (smoothness here implies the existence of the second derivatives) of $f(\mathbf{q}_{\text{id}})$ we know that $\mathbf{C}_i^T \mathbf{Q} \mathbf{C}_i \succ 0$ (positive definite) at the solution, but this does not ensure that all elements of \mathbf{q} are positive. As described previously, for an orthogonal grid there exists always one independent edge along any grid row or column. As a consequence, for any given positive independent force density, all edges in that continuous line must have positive force densities for nodal equilibrium to be satisfied. This leads to the ideal case that the dependent force densities will always be positive for given positive independent force densities, ensuring a compression-only solution. For non-orthogonal networks this is not the case, as \mathbf{q}_{id} values may lead to negative \mathbf{q} values in dependent edges.

3.3.2 Computational approach

The numerical implementation of the load-path optimisation has been tested on both function based and combined function and gradient based optimisation solvers using the numerical and scientific NumPy [19] and SciPy [20] packages for the Python programming language [21]. The optimisation solvers that were used, were (1) an implementation of the Differential Evolution solver [22], which is a function only based evolutionary algorithm, and (2) the SLSQP solver in SciPy Optimise, which is a function and gradient

(numerical or analytical gradient) Sequential Least Squares Programming method. The algorithm proceeds as follows:

- For the SLSQP method an initial starting point must be given. A randomly selected point within the bounds of the \mathbf{q}_{id} domain may not necessarily allow the solver to proceed efficiently or successfully, as it may for example start with many negative force densities. An alternative selection of starting point, is to use the Differential Evolution method to find a selection of good starting candidates. The default SciPy SLSQP relative convergence tolerance of 10^{-6} is used for the requested accuracy parameter in the `fmin_slsqp` function.
- When using the Differential Evolution algorithm, agents in the population that have negative force densities \mathbf{q}_{neg} are penalised with a penalty function of $f = A + \sum |\mathbf{q}_{neg}|$ to distinguish between poor agents with many large negative \mathbf{q} values and better quality agents with one or only a few small negative \mathbf{q} values. The value of A should be selected high enough to be sufficiently greater than the expected load-path magnitude, so that agents that contain compression-only solutions are unaffected.
- For the networks studied in this research, it was found that the Differential Evolution solver was sometimes needed for between 100–500 generations to find a suitable starting point for the SLSQP solver. It was not necessary for the starting point to be exclusively a compression-only state, as a compression only inequality constraint could still be used to guide to the final state. But it was observed that randomly selected SLSQP starting points without some degree of pre-selection would cause divergence during the optimisation.
- The Differential Evolution method was also used on its own to reach the same optimum solution as the DE + SLSQP combination, although with considerably greater computation time. So with this observation the combination of the two methods was preferred. The SLSQP solver was chosen in preference to other function and gradient based optimisation methods as it can easily include bounds on \mathbf{q}_{id} , non-negativity inequalities for $\mathbf{q} > 0$, and permits the use of the analytical

gradient or a local numerical approximation of the gradient.

- The gradient of the load-path function has been derived (see Sect. 6.1 for the full derivation and definition of the following terms and symbols) as

$$\begin{aligned} \nabla f(\mathbf{q}_{id}) = & \sum_{i=1}^m (\mathbf{x}_b^T \mathbf{C}_b^T \mathbf{e}_i \mathbf{x}^T \mathbf{C}^T \mathbf{E}_i \mathbf{K} + \mathbf{y}_b^T \mathbf{C}_b^T \mathbf{e}_i \mathbf{y}^T \mathbf{C}^T \mathbf{E}_i \mathbf{K} \\ & - \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_i \mathbf{K} \\ & + \mathbf{z}_b^T (\mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T - \mathbf{C}_b^T) \mathbf{e}_i \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_i \mathbf{K}). \end{aligned} \quad (33)$$

- Other optimisation algorithms, such as quasi-Newton (SR1, BHHH, BFGS family) and trust-region methods, can make use of the analytical Hessian, which is a $k \times k$ matrix, given by

$$\begin{aligned} \nabla^2 f(\mathbf{q}_{id}) = & \sum_{i=1}^m [\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} [\mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) \mathbf{e}_i \mathbf{C}_i^T \\ & + \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i \mathbf{C}_i^T + \mathbf{p}_z \mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T] \otimes (\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) \\ & + \mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{p}_z \mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T \\ & \otimes \mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T)] \Omega \mathbf{K}, \end{aligned} \quad (34)$$

where $\Omega = \sum_{i=1}^m \mathbf{I}_{m^2 \times m} (m(i-1) + i, i)$ is a constant sparse matrix. The Hessian function, along with methods that require either its analytical or numerical approximation, have not been studied in this paper. The function is included here for completeness and for further application in the optimisation methods that can benefit from it.

Solving the above semiconic optimisation problem outputs the optimal distribution of (horizontal) forces that results in the overall minimum-volume solution for given form diagram, applied loads and boundary conditions.

3.4 Computational simplifications on the load-path gradient and Hessian to improve efficiency

It is worth noting that computationally, the gradient and Hessian functions are simplified from the presented rigorous mathematical equations. In particular, this is to avoid explicitly using the summations over i , as this involves the use of programming loops that will slow down computations significantly. The following computational steps simplify the gradient and Hessian calculations.

- If the summations are utilised and are programmed as loop statements, it is recommended to use sparse two-dimensional arrays where possible for the sparsely populated matrices. This is especially true for terms involving the multiplication of multiple sparse matrices such as $\mathbf{D}_b = \mathbf{C}_i^T \mathbf{Q} \mathbf{C}_b$. For networks with higher numbers of elements, loop statements may be time consuming to evaluate.
- Summation loops may be avoided by vectorising the calculations. This may be performed by recognising that the matrices used in the calculations are 2D of general size $(a \times b)$, and can be stacked/tiled as slices into 3D arrays of size $(a \times b \times m)$ where m is the number of edges. For the \mathbf{e}_i ($m \times 1$) and \mathbf{E}_i ($m \times m$) arrays described later in Sect. 6.1, these will be converted into $(m \times 1 \times m)$ and $(m \times m \times m)$ sized arrays, respectively. The matrix calculations can then proceed as normal with appropriate functions supporting vectored input, with the summation finally applied in a vectored manner to the final arrays along the stacked dimension (third dimension in this case). Speed gains can be found with larger arrays sizes with this vectorisation, but at the cost of using arrays that consume more memory. The Ω matrix for the Hessian can be pre-constructed and used in a similar manner.
- All of the terms that are independent of \mathbf{q} , and so do not vary with changes in \mathbf{q}_{id} , may be calculated before the start of the optimisation process to avoid spending time on recalculating arrays. This includes individual vectors and matrices such as

\mathbf{x} , \mathbf{C} and \mathbf{p}_z , but also more complicated terms like $\mathbf{x}_b^T \mathbf{C}_b^T \mathbf{e}_i \mathbf{x}^T \mathbf{C}^T \mathbf{E}_i \mathbf{K}$.

4 Results

4.1 Grid comparisons

An orthogonal grid measuring ten by ten units and with ten bays in each direction was studied. Unit point loads were applied upwards at each vertex, with the perimeter vertices fixed from translating in all directions and so forming the boundary vertices. There are 18 degrees-of-freedom in this network, one along each row and column line of continuous edges. The independent force densities were bounded in the range $(0, 10]$. The calculated optimum solution with a load-path of 449.4 is plotted in Fig. 4 and formed the base solution for which to compare other solutions to. For each of the example solutions presented in this section, the number of function and gradient iterations for the SLSQP solver is given in each figure caption, which was used after 500 iterations of the differential evolution solver to find a starting point.

The thrust network for the result in Fig. 4 is plotted in Fig. 5, alongside the globally scaled (r_{\min}) constant force density solution. For the globally scaled uniform force density case, the maximum rise of the thrust network is 5.32, compared to 4.15 for the optimised case. The comparison shows that the load-path, and hence volume of the structure, was improved by 10%

Fig. 4 Form and force diagrams for the resulting optimised compression-only force distribution for the 10×10 orthogonal grid, with free vertices (white) and boundary vertices (blue). The thickness of the red lines represents the area, or magnitude of the internal forces for given stress. (Function evaluations: 1076, Gradient evaluations: 52). (Color figure online)

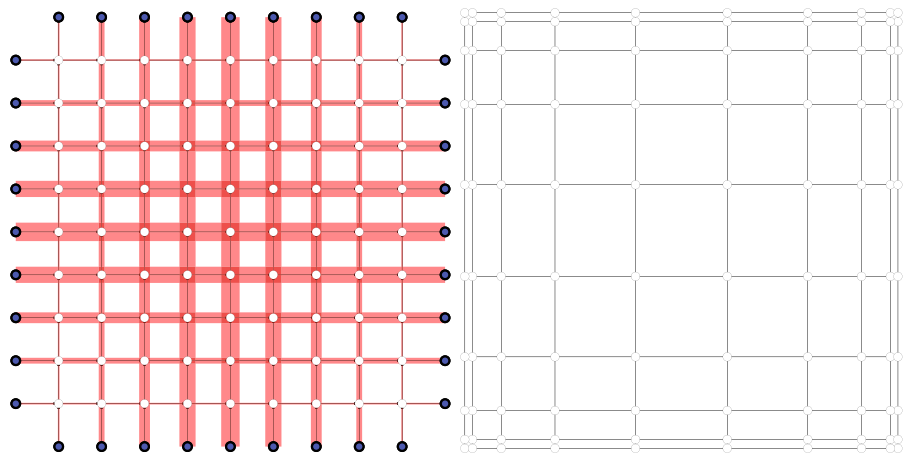


Fig. 5 Constant (scale optimised) and global optimised results for the orthogonal 10×10 grid

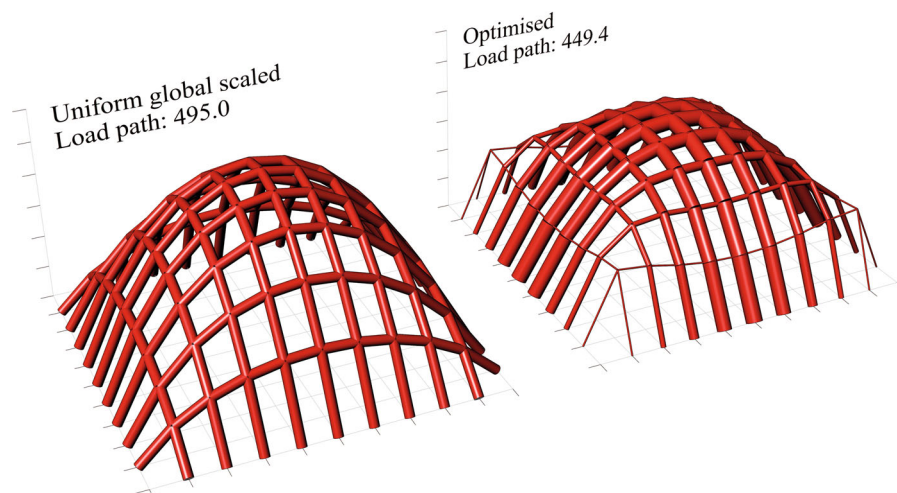
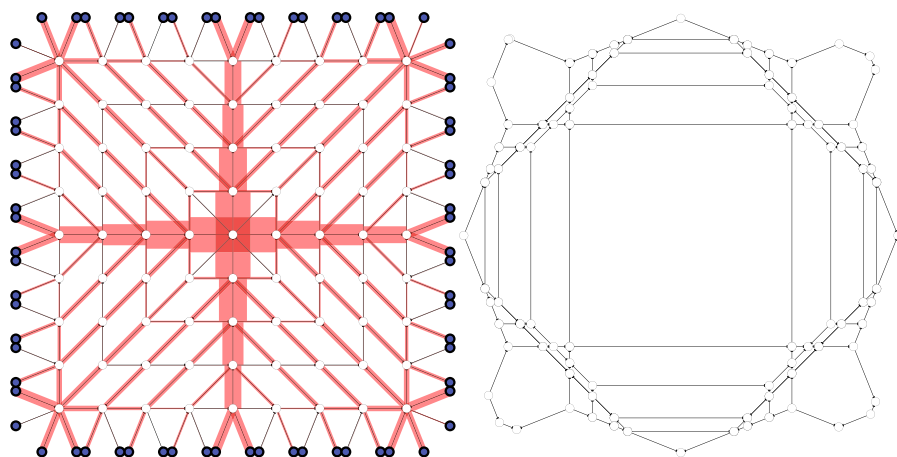


Fig. 6 V-shaped grid with general reactions and some base grid removed: Load-path = 525.8, $k = 70$. Two main arches split the network into quadrants, with the diagonals directing the forces towards the corner supports. This load-path was found to be the least efficient of all examined patterns. (Function evaluations: 4286, Gradient evaluations: 59)



by flattening the peak of the structure with increases in force densities. Such behaviour can be expected by recalling that the load-path is calculated by the force densities multiplied by the square of the edge lengths, emphasising greater importance on length reductions than force density increases. It is also observed that it was more efficient to redirect forces to the main arches in the middle, where loads are taken more effectively through the steeper forces. As a consequence of this, in this example loads are redirected away from the corners.

As the loads on the structure can only be carried towards the supports along the predefined directions of the edge patterns of the form diagrams, the choice of pattern determines which force paths can be formed and will therefore influence the obtained value for the load-path. Figs. 6, 7, 8, 9, 10 and 11 show, for the same

applied loads, different patterns of edges, either by overlaying new edges on top of the base orthogonal grid, or by removing some edges entirely. For all of these figures, the thickness of the red lines represents the area, or magnitude of the internal forces for given stress. These figures are presented in descending order of load-path value, from the highest load-path value of 525.8 to the lowest value of 438.1, the latter being an improvement over the original value of 449.4. Adding the new edges in these examples increases the degrees-of-freedom k from 18 to between 70 and 182, allowing more potential load traversal routes. In the limit that the number of elements keeps increasing, the continuum representation is approached, albeit with rapidly increasing analysis times. Figs. 7 and 8 show that for the same internal network layout, introducing general reactions with horizontal and vertical components in

Fig. 7 V-shaped grid with orthogonal reactions: Load-path = 446.0, $k = 82$. Diagonal edges were overlaid onto the orthogonal grid without any edges removed, and showed that an improvement in the load-path could be found by shedding some of the load into a cross pattern at the centre of the network. (Function evaluations: 7020, Gradient evaluations: 83)

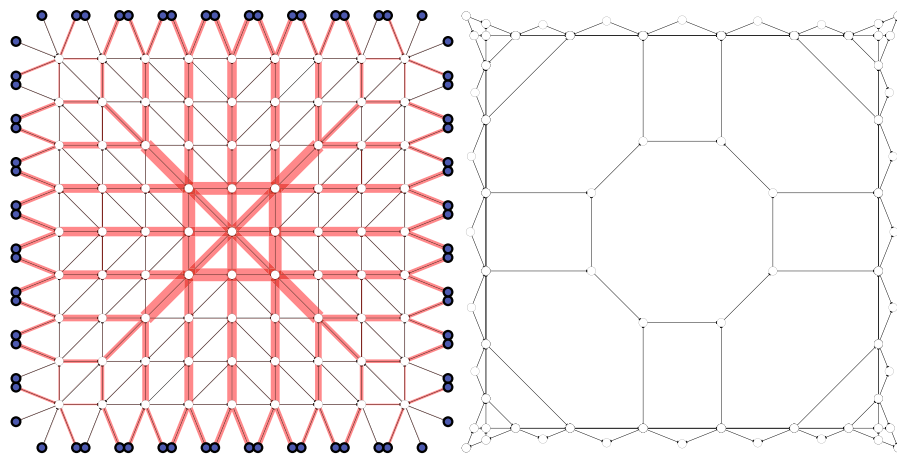
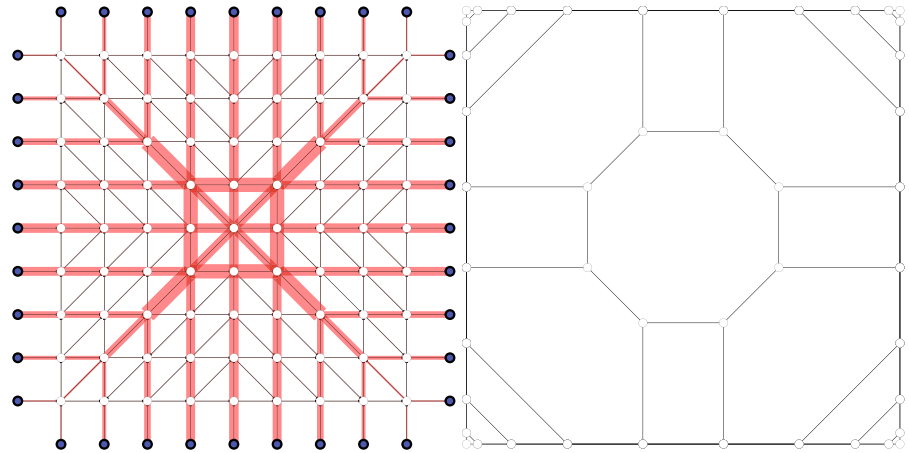


Fig. 8 V-shaped grid with general reactions: Load-path = 445.4, $k = 118$. By changing the supports in Fig. 7 from taking only orthogonal reactions to general reactions with components in both planar directions, a very slight

improvement was found. This can be attributed to allowing thrusts coming at angles to the supports to be taken by the fixed nodes and not from the orthogonal set of edges. (Function evaluations: 7713, Gradient evaluations: 64)

plan rather than just orthogonal reactions, can subtly change the form and force diagrams for an improved load-path.

4.2 General non-orthogonal network

The network in Fig. 12 possesses a general, non-orthogonal form diagram and is relatively dense with $k = 217$. It consists of a triangulated network supported along three boundary edges, and with concentrated loading applied at the vertices based on scaled area-weighted loads. The topology for this non-orthogonal network was generated with a Delaunay triangulation of a distribution of vertices along the

boundary and interior of the network. For analysis, a combination of the Differential Evolution algorithm to find a starting point was then followed by the function and gradient SLSQP method. The allowable force density range for the \mathbf{q}_{id} input was $(0, 5]$, where at the end of the analysis, the optimum load-path was found to be 249.8 with a maximum independent force density of 3.61. The main arching edge loops can be seen spanning between the support lines, as well as secondary main thrust lines forming within the interior of the network. The associated thrust network is plotted in Fig. 13, which took 13540 function evaluations and 215 gradient evaluations to analyse.

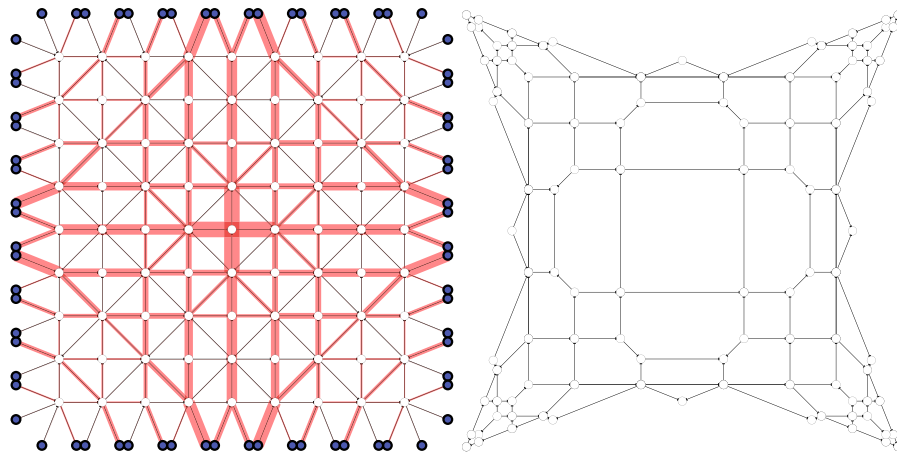
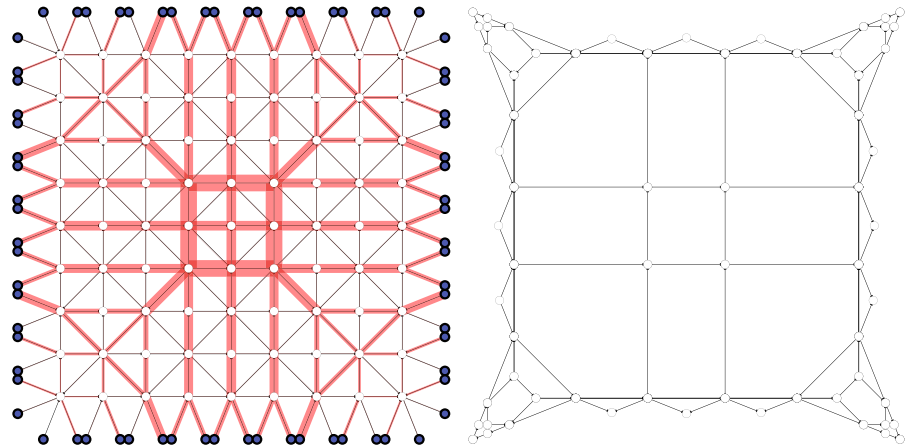


Fig. 9 Diamond grid with general reactions: Load-path = 441.0, $k = 118$. By changing the diagonal biased networks of Figs. 7 and 8 into a diamond shaped network with diagonals in both directions, a completely different pattern of

internal forces developed. Forces were found to move away from the supports and form a larger diagonal shaped region in the centre of the network. (Function evaluations: 9256, Gradient evaluations: 75)

Fig. 10 Shifted diamond grid with general reactions: Load-path = 440.5, $k = 118$. By simply translating the diamond pattern of Fig. 9, a new force distribution was found. This can be seen as similar to the pattern of Fig. 8, but with additional diagonal elements in the other direction. (Function evaluations: 9021, Gradient evaluations: 74)



5 Conclusions

This paper presented the derivation and implementation of a load-path optimisation method for discrete thrust networks based on Thrust Network Analysis. The internal and external load-path functions, which are related to the volume and hence mass of the thrust network elements, were derived for compression-only structures. The geometry of these compression-only thrust networks can be manipulated by assigning force densities to the independent edges for a given set of point loads and fixed boundary points on a common plane.

The load-path concept was illustrated first with a continuous and discrete parabolic arch example,

whereby the optimal height of the compression arch was found by discrete scale optimisation. This method determined the value of the scale factor of the form and force diagrams that gave the minimum volume solution for the discrete parabolic arch, mirroring the analytical solution of the continuous case.

The external load-path function was laid out, along with the associated gradient and Hessian functions for use in optimisation solvers. A numerical implementation featuring the use of function and both function and gradient-based solvers was used to determine the general optimum load-path, so that spatial discretised thrust networks could be tackled. A combination of SLSQP and Differential Evolution solvers was used to find the set of independent edge force densities that

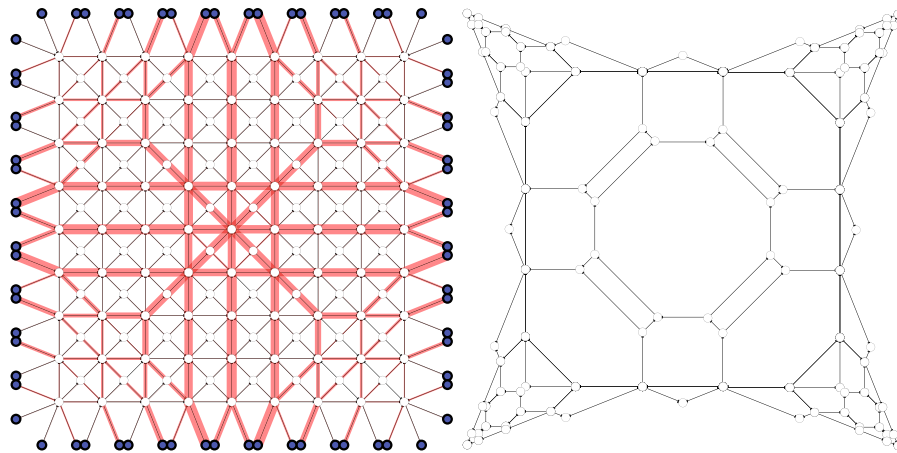


Fig. 11 X-shaped grid with general reactions: Load-path = 438.1, $k = 182$. The lowest value for these load-path examples was found by inserting additional vertices within each bay, and then joining them to the original vertices with four edges forming a cross. This was the most general solution with the

highest number of degrees-of-freedom. The independent branches that did not need to be activated, were found to have optimum force density close to zero. (Function evaluations: 9371, Gradient evaluations: 92)

Fig. 12 Form and force diagrams with resulting compression-only force distribution for the dense triangulated network

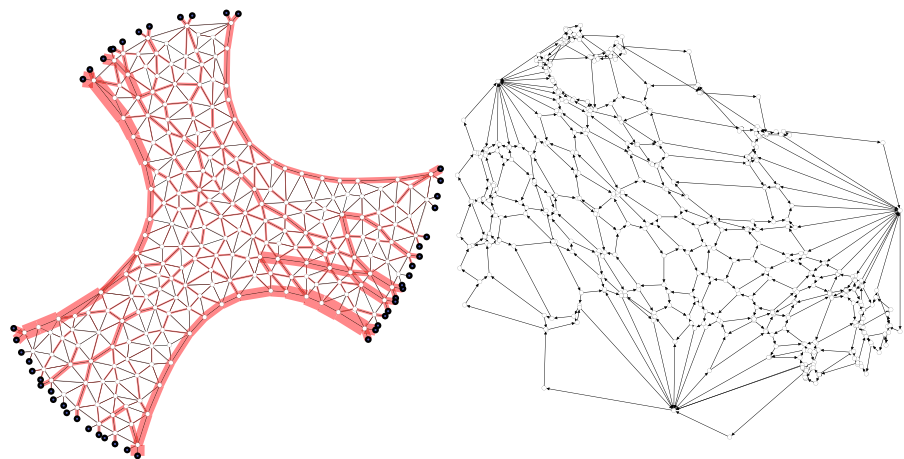
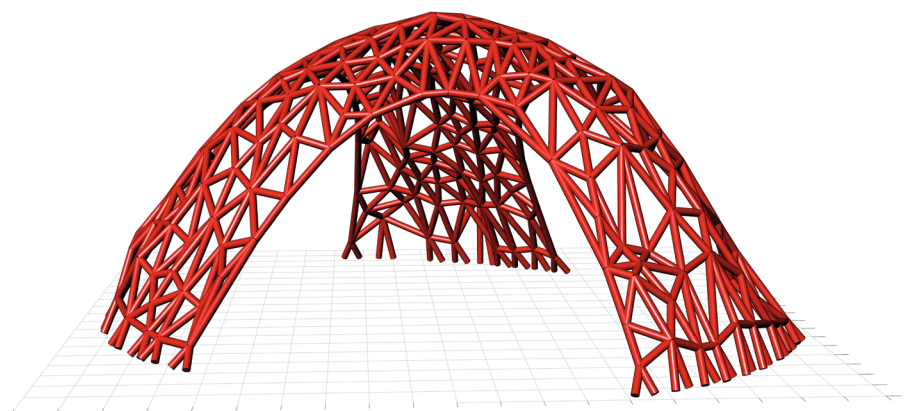


Fig. 13 Thrust network for the triangulated structure



minimised the load-path function subject to compression-only constraints.

A set of load-path optimised thrust networks were compared that used the same loading state, with differences in the pattern of additional overlaid edges and whether the fixed vertex supports were orthogonal or general. The comparisons showed that different network topologies lead to different values of optimised load-path function, with the efficiency dependent on how well the network encourages the flow of compression forces. The optimal solutions, when compared with a discrete scaled uniform force density solution, tended to have less overall rise and with preference to shorter members rather than longer members with higher force densities.

A further example demonstrating a general non-orthogonal network showed that structures of high network indeterminacy can be handled with the current load-path implementation. The optimised load-path method allows for the investigation of efficient load carrying patterns for discrete networks, so that they may be analysed for weight minimisation.

Acknowledgements The authors would like to acknowledge William Baker for presenting the load-path concept to the authors and for solving the optimal h/l ratio for the continuous parabola.

Appendix

Gradient and Hessian derivation of the load-path function

To calculate the gradient and Hessian of the external load-path function in Eq. (31), the following common matrix calculus properties are used, which follow [23] Differential Calculus notation,

$$\frac{d\mathbf{AXB}}{d\mathbf{X}} = \mathbf{B}^T \otimes \mathbf{A} \quad (35a)$$

$$\frac{d\mathbf{A}^{-1}}{d\mathbf{A}} = -(\mathbf{A}^{-T} \otimes \mathbf{A}^{-1}), \quad (35b)$$

Note that this derivative is essentially a fourth-order tensor, whose generic component represents the derivative of the inverse of a matrix with respect to its matrix. Let matrix functions $f: \mathbb{R}^{n \times k} \rightarrow \mathbb{R}^{m \times p}$ and $g: \mathbb{R}^{n \times k} \rightarrow \mathbb{R}^{p \times q}$ then

$$\frac{df(\mathbf{X}) \cdot g(\mathbf{X})}{d\mathbf{X}} = (g(\mathbf{X})^T \otimes \mathbf{I}_m) f'(\mathbf{X}) + (\mathbf{I}_q \otimes f(\mathbf{X})) g'(\mathbf{X}), \quad (35c)$$

where \mathbf{I}_m and \mathbf{I}_q are the $m \times m, q \times q$ identity matrices, and \otimes is the Kronecker matrix operator, defined as the complete multiplication between two matrices i.e. if $\mathbf{A} (m \times n)$ and $\mathbf{B} (p \times q)$ matrices, then $\mathbf{A} \otimes \mathbf{B}$ is an $(mp \times nq)$ matrix. Using the chain rule on the load-path function, Equation (31) gives

$$\frac{df(\mathbf{q}_{id})}{d\mathbf{q}_{id}} = \frac{df(\mathbf{q}_{id})}{d\mathbf{Q}} \cdot \frac{d\mathbf{Q}}{d\mathbf{q}_{id}} \quad (36)$$

To find $\frac{d\mathbf{Q}}{d\mathbf{q}_{id}}$, notice that the diagonal matrix \mathbf{Q} can mathematically be written as a function of vector \mathbf{q} with

$$\mathbf{Q} = \sum_{i=1}^m \mathbf{E}_i \mathbf{q} \mathbf{e}_i^T, \quad (37)$$

where \mathbf{E}_i is an $m \times m$ matrix with all its entries zero except for identity in (i, i) , and \mathbf{e}_i is an $m \times 1$ vector with identity on the i th element and zero everywhere else. This derivative then becomes

$$\begin{aligned} \frac{d\mathbf{Q}}{d\mathbf{q}_{id}} &= \frac{d \sum_{i=1}^m \mathbf{E}_i \mathbf{q} \mathbf{e}_i^T}{d\mathbf{q}_{id}} = \frac{d \sum_{i=1}^m \mathbf{E}_i \mathbf{K} \mathbf{q}_{id} \mathbf{e}_i^T}{d\mathbf{q}_{id}} \\ &= \sum_{i=1}^m \mathbf{e}_i \otimes \mathbf{E}_i \mathbf{K} \end{aligned} \quad (38)$$

$$= \sum_{i=1}^m \mathbf{I}_{m^2 \times m} (m(i-1) + i, i) \mathbf{K} = \mathbf{\Omega} \mathbf{K}, \quad (39)$$

where the matrix $\mathbf{\Omega}$ is constructed by adding unity in the aforementioned slots $\forall i, 1 \leq i \leq m$ of the $m^2 \times m$ matrix \mathbf{I} .

Now the derivative $\frac{df(\mathbf{q}_{id})}{d\mathbf{Q}}$ of Eq. (36) is

$$\begin{aligned} \frac{df(\mathbf{q}_{id})}{d\mathbf{Q}} &= -(\mathbf{p}_z^T \otimes \mathbf{p}_z^T)(\mathbf{D}_i^{-1} \otimes \mathbf{D}_i^{-1})(\mathbf{C}_i^T \otimes \mathbf{C}_i^T) \\ &\quad + \mathbf{x}_b^T \mathbf{C}_b^T \otimes \mathbf{x}^T \mathbf{C}^T + \mathbf{y}_b^T \mathbf{C}_b^T \otimes \mathbf{y}^T \mathbf{C}^T \\ &\quad - \frac{d\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{D}_b \mathbf{z}_b}{d\mathbf{Q}} \\ &= -\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T + \mathbf{x}_b^T \mathbf{C}_b^T \otimes \mathbf{x}^T \mathbf{C}^T \\ &\quad + \mathbf{y}_b^T \mathbf{C}_b^T \otimes \mathbf{y}^T \mathbf{C}^T - \frac{d\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{D}_b \mathbf{z}_b}{d\mathbf{Q}} \end{aligned} \quad (40)$$

Note that the term $\mathbf{D}_i^{-1} \mathbf{D}_b$ is a function of \mathbf{q}_{id} if $\mathbf{C}_i \mathbf{C}_i^T$ and is singular, i.e its spectral decomposition has a diagonal matrix with at least one zero diagonal element. The invariance of this term, in the case that $\mathbf{C}_i \mathbf{C}_i^T$ is invertible, holds by

$$\begin{aligned} & \mathbf{p}_z^T (\mathbf{C}_i^T \mathbf{Q} \mathbf{C}_i)^{-1} \mathbf{C}_i^T \mathbf{Q} \mathbf{C}_b \mathbf{z}_b \\ &= \mathbf{p}_z^T (\mathbf{C}_i^T \mathbf{Q} \mathbf{C}_i)^{-1} \mathbf{C}_i^T \mathbf{Q} \mathbf{C}_i \mathbf{C}_i^T (\mathbf{C}_i \mathbf{C}_i^T)^{-1} \mathbf{C}_b \mathbf{z}_b \\ &= \mathbf{p}_z^T \mathbf{C}_i^T (\mathbf{C}_i \mathbf{C}_i^T)^{-1} \mathbf{C}_b \mathbf{z}_b. \end{aligned} \quad (41)$$

Assuming that $\mathbf{C}_i \mathbf{C}_i^T$ is not invertible, using Eq. (35c), the derivative of $\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{D}_b \mathbf{z}_b$ is then

$$\begin{aligned} & \frac{d[\mathbf{p}_z^T (\mathbf{C}_i^T \mathbf{Q} \mathbf{C}_i)^{-1}] \cdot (\mathbf{C}_i^T \mathbf{Q} \mathbf{C}_b \mathbf{z}_b)}{d\mathbf{Q}} \\ &= -(\mathbf{z}_b^T \mathbf{D}_b^T \otimes \mathbf{I}_1)(\mathbf{D}_i^{-1} \mathbf{C}_i^T \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) \\ & \quad + (\mathbf{I}_1 \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1})(\mathbf{z}_b^T \mathbf{C}_b^T \otimes \mathbf{C}_i^T) \\ &= -(\mathbf{z}_b^T \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) \otimes (\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) \\ & \quad + (\mathbf{z}_b^T \mathbf{C}_b^T) \otimes (\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) \\ &= [\mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T)] \otimes (\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T). \end{aligned} \quad (42)$$

Thus,

$$\begin{aligned} \frac{df(\mathbf{Q})}{d\mathbf{Q}} &= -\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T + \mathbf{x}_b^T \mathbf{C}_b^T \otimes \mathbf{x}^T \mathbf{C}^T \\ & \quad + \mathbf{y}_b^T \mathbf{C}_b^T \otimes \mathbf{y}^T \mathbf{C}^T \\ & \quad - [\mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T)] \otimes (\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T). \end{aligned} \quad (43)$$

Inserting Eqs. (43) and (39) into (36) gives the $1 \times k$ gradient of the load-path

$$\begin{aligned} \nabla f(\mathbf{q}_{id}) &= \frac{df(\mathbf{Q})}{d\mathbf{Q}} \cdot \frac{d\mathbf{Q}}{d\mathbf{q}_{id}} \\ &= [-(\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) + \mathbf{x}_b^T \mathbf{C}_b^T \otimes \mathbf{x}^T \mathbf{C}^T \\ & \quad + \mathbf{y}_b^T \mathbf{C}_b^T \otimes \mathbf{y}^T \mathbf{C}^T \\ & \quad - [\mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T)] \otimes (\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T)] \cdot \Omega \mathbf{K} \\ &= \sum_{i=1}^m [\mathbf{x}_b^T \mathbf{C}_b^T \mathbf{e}_i \mathbf{x}^T \mathbf{C}^T \mathbf{E}_i \mathbf{K} + \mathbf{y}_b^T \mathbf{C}_b^T \mathbf{e}_i \mathbf{y}^T \mathbf{C}^T \mathbf{E}_i \mathbf{K} \\ & \quad - \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_i \mathbf{K} \\ & \quad + \mathbf{z}_b^T (\mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T - \mathbf{C}_b^T) \mathbf{e}_i \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_i \mathbf{K}]. \end{aligned} \quad (44)$$

In the same fashion, to determine the Hessian of the function, the following chain rule is used

$$\nabla^2 f(\mathbf{q}_{id}) = \frac{d\nabla f}{d\mathbf{Q}} \cdot \frac{d\mathbf{Q}}{d\mathbf{q}_{id}}. \quad (45)$$

Note that one can temporarily ignore the sums of the gradient, since $d \sum = \sum d$, and consider them in the final chain-rule calculation. The first two terms of Equation (44) vanish in the Hessian, leaving only the derivative

$$\begin{aligned} \frac{d\nabla f}{d\mathbf{Q}} &= -\frac{d\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_i \mathbf{K}}{d\mathbf{Q}} \\ & \quad + \frac{d\mathbf{z}_b^T (\mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T - \mathbf{C}_b^T) \mathbf{e}_i \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_i \mathbf{K}}{d\mathbf{Q}}. \end{aligned} \quad (46)$$

The first term of Eq. (46) by the multiplication rule is equivalent to

$$\frac{d(\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i)(\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_i \mathbf{K})}{d\mathbf{Q}} \quad (47)$$

$$= -(\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{p}_z \otimes \mathbf{I}_1)[\mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T] \quad (48)$$

$$- (\mathbf{I}_{k \times k} \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i)[\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T] \quad (49)$$

$$= -[\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} (\mathbf{p}_z \mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T + \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i \mathbf{C}_i^T) \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T]. \quad (50)$$

The second derivative of Eq. (46) is slightly more complicated and is thus split in two parts

$$\frac{d\mathbf{z}_b^T \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_i \mathbf{K}}{d\mathbf{Q}} \quad (51)$$

$$\frac{d\mathbf{z}_b^T \mathbf{C}_b^T \mathbf{e}_i \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_i \mathbf{K}}{d\mathbf{Q}}. \quad (52)$$

Using Eqs. (35a) and (35b), the derivative (52) becomes

$$-\mathbf{z}_b^T \mathbf{C}_b^T \mathbf{e}_i (\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T). \quad (53)$$

Using the multiplication rule on Eq. (51) gives

$$\frac{d(\mathbf{z}_b^T \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i)(\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_i \mathbf{K})}{d\mathbf{Q}} \quad (54)$$

$$= (\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{p}_z \otimes \mathbf{I}_1)[(\mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T) \otimes \mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T)] \quad (55)$$

$$- (\mathbf{I}_{k \times k} \otimes \mathbf{z}_b^T \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i) [\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T] \quad (56)$$

$$= \mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{p}_z \mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T \otimes \mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) \quad (57)$$

$$- \mathbf{z}_b^T \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i (\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T \otimes \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) \quad (58)$$

Putting (50), (58) and (53) together gives

$$\begin{aligned} \frac{d\nabla f}{d\mathbf{Q}} = \sum_{i=1}^m & [\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} [\mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) \mathbf{e}_i \mathbf{C}_i^T \\ & + \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i \mathbf{C}_i^T + \mathbf{p}_z \mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T] \\ & \otimes (\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) + \mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{p}_z \mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T \\ & \otimes \mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T)] \end{aligned} \quad (59)$$

Finally, using Eq. (45) one gets the $k \times k$ Hessian matrix of the load-path

$$\begin{aligned} \nabla^2 f(\mathbf{q}_{id}) = \sum_{i=1}^m & [\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} [\mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) \mathbf{e}_i \mathbf{C}_i^T \\ & + \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i \mathbf{C}_i^T + \mathbf{p}_z \mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T] \\ & \otimes (\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) + \mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{p}_z \mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T \\ & \otimes \mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T)] \cdot \Omega \mathbf{K} \\ = \sum_{j=1}^m & \sum_{i=1}^m [\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} [\mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T) \mathbf{e}_i \mathbf{C}_i^T \\ & + \mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_i \mathbf{C}_i^T + \mathbf{p}_z \mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T] \mathbf{e}_j \\ & \otimes (\mathbf{p}_z^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_j \mathbf{K}) + (\mathbf{K}^T \mathbf{E}_i \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{p}_z \mathbf{e}_i^T \mathbf{C}_i \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{e}_j) \\ & \otimes \mathbf{z}_b^T (\mathbf{C}_b^T - \mathbf{D}_b^T \mathbf{D}_i^{-1} \mathbf{C}_i^T \mathbf{E}_j \mathbf{K})]. \end{aligned} \quad (60)$$

All authors confirm/declare that they have no conflict of interests with respect to the submitted research project.

References

- De Wilde WP (2006) Conceptual design of lightweight structures: the role of morphological indicators and the structural index. High Perform Struct Mater III WIT Trans Built Environ 85:3–12
- Vandenbergh T, De Wilde WP, Lateur P, Verbeeck B, Ponsaert W, Van Steirteghem J (2006) Influence of stiffness constraints on optimal design of trusses using morphological indicators. High Performance Structures and Materials III. WIT Trans Built Environ 85:31–40
- Pyl L, Sitters CWM, De Wilde WP (2013) Design and optimization of roof trusses using morphological indicators. Adv Eng Softw 62–63:9–19
- Mazurek A, Baker WF, Tort C (2011) Geometrical aspects of optimum truss like structures. Struct Multidisciplinary Optim 43(2):231–242
- Baker WF, Beghini LL, Mazurek A, Carrion J, Beghini A (2013) Maxwells reciprocal diagrams and discrete Michell frames. Struct Multidisciplinary Optim 48(2):267–277
- Beghini LL, Carrion J, Beghini A, Mazurek A, Baker WF (2014) Structural optimization using graphic statics. Struct Multidisciplinary Optim 49(3):351–366
- Maxwell JC (1864) On reciprocal figures and diagrams of forces. Philos Mag J Ser 4(27):250–261
- Baker WF, Beghini A, Mazurek A (2012) Applications of structural optimization in architectural design. In: 20th analysis and computation speciality conference, pp 257–266
- Block P, Ochsendorf J (2007) Thrust network analysis: a new methodology for three-dimensional equilibrium. J Int Assoc Shell Spat Struct 48(3):167–173
- Block P (2009) Thrust network analysis: exploring three-dimensional equilibrium. PhD dissertation, Massachusetts Institute of Technology, Cambridge, USA
- O'Dwyer DW (1999) Funicular analysis of masonry vaults. Comput Struct 73:187–197
- Fraternali F (2010) A thrust network approach to the equilibrium problem of unreinforced masonry vaults via polyhedral stress functions. Mech Res Commun 37:198–204
- Marmo F, Rosati L (2017) Reformulation and extension of the thrust network analysis. Comput Struct 182:104–118
- Linkwitz K, Schek HJ (1971) Einige Bemerkungen zur Berechnung von vorgespannten Seilnetzkonstruktionen. Ingenieur Archiv 40:145–158
- Schek HJ (1974) The force density method for form finding and computation of general networks. Comput Methods Appl Mech Eng 3:115–134
- Van Mele T, Rippmann M, Lachauer L, Block P (2012) Geometry-based understanding of structures. J Int Assoc Shell Spat Struct 53(2):285–295
- Block P, Lachauer L (2014) Three-dimensional funicular analysis of masonry vaults. Mech Res Commun 56:53–60
- Van Mele T, Block P (2014) Algebraic graph statics. Comput Aided Des 53:104–116
- Van der Walt S, Colbert C, Varoquaux G (2011) The NumPy array: a structure for efficient numerical computation. Comput Sci Eng 13:22–30
- Jones E, Oliphant T, Peterson P, et al. (2001) SciPy: open source scientific tools for Python, (Online; accessed 2017-02-13)
- Python Software Foundation. Python Language Reference, 2016. Version 3.5
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
- Magnus JR, Neudecker H (1985) Matrix differential calculus with applications to simple, hadamard, and kronecker products. J Math Psychol 29(4):474–492