Research Article

Francesco Marmo*

ArchLab: a MATLAB tool for the Thrust Line Analysis of masonry arches**

https://doi.org/10.1515/cls-2021-0003
Received Sep 23, 2020; accepted Nov 18, 2020

Abstract: According to Heyman's safe theorem of the limit analysis of masonry structures, the safety of masonry arches can be verified by finding at least one line of thrust entirely laying within the masonry and in equilibrium with external loads. If such a solution does exist, two extreme configurations of the thrust line can be determined, respectively referred to as solutions of minimum and maximum thrust.
In this paper it is presented a numerical procedure for determining both these solutions with reference to masonry arches of general shape, subjected to both vertical and horizontal loads. The algorithm takes advantage of a simplification of the equations underlying the Thrust Network Analysis. Actually, for the case of planar lines of thrust, the horizontal components of the reference thrusts can be computed in closed form at each iteration and for any arbitrary loading condition. The heights of the points of the thrust line are then computed by solving a constrained linear optimization problem by means of the Dual-Simplex algorithm. The MATLAB implementation of presented algorithm is described in detail and made freely available to interested users (https://bit.ly/3krlVxH). Two numerical examples regarding a pointed and a lowered circular arch are presented in order to show the performance of the method.

Keywords: thrust line, limit analysis, masonry arch

1 Introduction

The analysis of masonry arches is a classical problem of structural mechanics. It was from the intuitions by Hooke, hidden in a famous anagram, and by Gregory, explicitly stated, that started a series of researches employing the *catenary analogy* to model the equilibrium of arches by means of compressive internal actions that are funicular of the applied loads [11]. Alternative approaches, initiated by La Hire and developed by Coulomb, analyze the equilibrium of arches by studying collapse mechanisms of a series of voussoirs capable to transfer a limited set of actions [28]. A formal explanation of these approaches was given in two renowned papers by Heyman [25] and Koohar-ian [29], where the *safe* and *unsafe* theorems of the limit analysis of masonry structures are presented.

Nowadays several computational methods are available for the analysis of masonry structures. Some of them are based on the Finite Element Method (FEM) and are capable to take into account sophisticated material models [3, 32, 33, 47]. However, an appropriate application of FEM based analyses requires an accurate knowledge of the value and spatial distribution of mechanical properties of materials and support settlements [26, 27], which requires detailed survey techniques, unmotivated for ordinary structures.

A interesting alternative is represented by the methods that employ the No-Tension (NT) material model [7, 17], which is characterized by a non-smooth behavior and requires suitable techniques to be successfully employed for the analysis of real structures [8, 18]. Additionally, the NT model is embodied by the hypotheses of previously mentioned limit theorems. Some of their computational implementations include the Thrust Network Analysis (TNA) [11, 12, 34, 36, 44] and the Thrust Surface Analysis (TSA) [20], both based on the safe theorem, or the Rigid Block Analysis [13, 24, 31] and the Discrete Element Method (DEM) [30, 49], which employ the unsafe theorem. Worth mentioning are recent proposals in which a fracture mechanics-based analytical method with elastic-softening of masonry is applied to analyse the structural behaviour of arch bridges and show how the arch thrust line is affected by crack formation [1, 2].

Although these methods are all capable to analyze structures characterized by complicated geometries [35, 46] and by unusual construction techniques [19, 37], specific

***Corresponding Author: Francesco Marmo:** Dipartiment of Structures for Engineering and Architecture, University of Naples Federico II, Naples, Italy; Email: f.marmo@unina.it
** Paper included in the Special Issue entitled: Shell and Spatial Structures: Between New Developments and Historical Aspects

tools for the analysis of masonry arches are still of scientific interest [6, 38, 45]. Methods based on the kinematic approach are used to determine the collapse of arches undergoing spreading of supports [16, 22]. Methods based on Heyman principles are employed to compute the minimum thickness [4, 40] and the later load bearing capacity [5, 14] of arches.

Interestingly, the thrust line analysis can also be used to determine the optimal shape of masonry arches subjected to vertical and horizontal loads [39, 41]. In particular, Michiels and Adriaenssens [39] compute a unique thrust line by defining its span and height. The determination of this thrust line is obtained by iteratively adjusting the horizontal and vertical forces applied to the nodes of the thrust line. This thrust line is then mirrored with respect to the vertical axis so that the pair of mirrored thrust lines are used to define the profile of the arch, while an iterative procedure minimizes its total volume. The approach proposed by Nikolic [41], instead, employs an analytical modeling of a catenary arch of constant and finite thickness, for which the horizontal thrust is determined. Then a thrust line is analytically determined for this arch. Nikolic shows that this thrust line is not coincident with the catenary curve that defines the axis of the arch because of the altered position of the center of gravity of voissors due to their finite thickness and curvature.

Worth mentioning are the two MATLAB tools `ArchNURBS` [15] and `FRS_Method` [23]. The approach implemented in `ArchNURBS` is based on a nonuniform rational B-splines (NURBS) representation of arch geometry. A preliminary isogeometric finite-element elastic analysis of the arch is performed in order to determine the structural response under service loads, provided that the corresponding thrust line fulfills assumed geometric constraints. Successively a limit analysis based on the safe theorem by Heyman is carried on by considering equilibrium and yielding conditions of blocks interfaces. These equations and constraints are solved as a linear optimization problem that maximizes the load multiplier. The method employed in `FRS_Method`, insted, computes the line of thrust by solving an optimization problem that looks for the funicular polygon closest to the geometrical axis of the arch [48]. This means that a unique line of thrust is determined, which is neither one of the two limit configurations of minimum or maximum thrust, neither it obeys to the principle of least action. Additionally, the authors employed a new definition of geometric safety factor of the arch, which is alternative to the one given by Heyman.

The objective of this paper is the *Thrust Line Analysis* (TLA) of masonry arches, a specialization of the TNA to the two-dimensional case. As the TNA represents a discrete implementation of the no-tension membrane model [21] the TLA represents a discrete version of the funicular curve. The specialization to a planar line of thrust simplifies enormously the solution of the horizontal equilibrium equations that are used to compute the horizontal components of thrust. Actually, in the proposed version of the method, reference values of horizontal thrust are computed in closed form at each iteration.

The proposed version of the TLA has been implemented in a MATLAB code, freely downloadable from https://bit.ly/3krlVxH. The method is implemented within the function `ArchLab` that takes as input a table containing the geometric and loading data of the arch. Also some ready-to-run example files are provided together with a plotting function, useful to graphically visualize the solutions.

After describing the general method, this paper illustrates in detail how each portion of the algorithm has been implemented in the MATLAB function `ArchLab`. Results regarding the analysis of a pointed and a lowered arch are also reported in order to show its performance.

2 Thrust Line Analysis of masonry arches

A discretized thrust line, or funicular polygon, is represented in Figure 1. It is described by means of B segments and N vertices. Being the thrust line an open polygon, it is always $B = N - 1$.

Borrowing the nomenclature from the TNA, segments and vertices that form the funicular polygon are called branches and nodes. Branches and nodes are ordered form left to right; hence, branch b_j connects nodes n_j and n_{j+1}, while branches b_{j-1} and b_j share the same node n_j.

The geometry of the thrust line is described by the coordinates $\mathbf{x}_j = (x_j,\ y_j)$ of nodes, defined in a two-dimensional Cartesian reference frame. According to Heyman's principles, the thrust line is required to be contained within the thickness of the arch, hence the vertical coordinates of nodes are required to fulfill the inequalities $y_{j,\min}$ and $y_{j,\max}$, where $y_{j,\min}$ and $y_{j,\max}$ are the heights of the arch intrados and extrados. However, different choices can be done when setting $y_{j,\min} \le y_j \le y_{j,\max}$, depending on the specific needs of the user. For instance they can be set equal to the thirds of the arch thickness if the limit condition implies a fully compressed arch.

The generic j-th node is loaded by external forces $\mathbf{f}_j = (f_{jx},\ f_{jy})$, while the first and last nodes are also subjected to the reactions $\mathbf{r}_l$ and $\mathbf{r}_r$ of left and right springers of the arch. Nodes are also subjected to the thrust forces transmitted

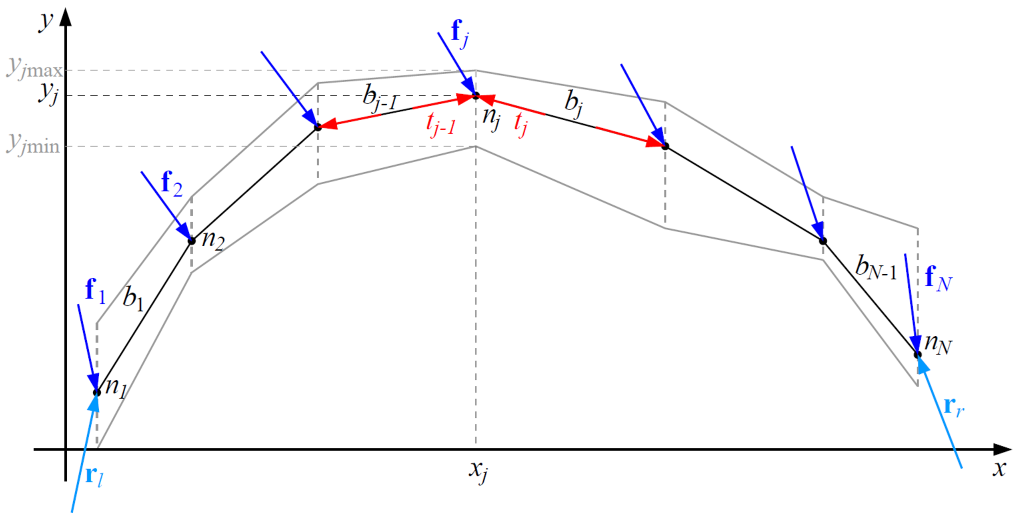

Figure 1: A schematic view of the thrust line

by the adjacent branches. In order to model these internal forces, a thrust value t_j is associated to the generic j-th branch of the thrust line. Since it represents a compressive force, t_j must be positive, i.e. $t_j > 0$, $j = 1, \dots, B$.

Within the TLA framework, the horizontal position of nodes and the external forces are known. Thrusts associated to branches, vertical position of all nodes and the reactions of springers are unknown. In the spirit of the Heyman's safe theorem, these unknowns are computed by employing equilibrium equations only.

The equilibrium of the generic j-th node of the thrust line is

$$\mathbf{t}_j^{j-1} + \mathbf{t}_j^j + \mathbf{f}_j = \mathbf{0} \qquad (1)$$

where $\mathbf{t}_j^{j-1}$ and $\mathbf{t}_j^j$ are the thrust forces that branches b_{j-1} and b_j transmit to node n_j. These forces have modulus equal to the thrust values t_{j-1} and t_j associated to branches b_{j-1} and b_j and they have direction parallel to the corresponding branches. Thus, they are computed as

$$\mathbf{t}_j^{j-1} = t_{j-1}\frac{\mathbf{x}_j - \mathbf{x}_{j-1}}{|\mathbf{x}_j - \mathbf{x}_{j-1}|} = t_{j-1}\frac{\mathbf{x}_j - \mathbf{x}_{j-1}}{\ell_{j-1}}, \qquad (2)$$
$$\mathbf{t}_j^{j} = t_{j}\frac{\mathbf{x}_j - \mathbf{x}_{j+1}}{|\mathbf{x}_j - \mathbf{x}_{j+1}|} = t_{j}\frac{\mathbf{x}_j - \mathbf{x}_{j+1}}{\ell_{j}}$$

where $\ell_j = |\mathbf{x}_{j+1} - \mathbf{x}_j| = \sqrt{(x_{j+1}-x_j)^2 + (y_{j+1}-y_j)^2}$ is the length of branch b_j.

Employing previous formulas, equilibrium of node n_j becomes

$$\frac{t_{j-1}}{\ell_{j-1}}(\mathbf{x}_j - \mathbf{x}_{j-1}) + \frac{t_j}{\ell_j}(\mathbf{x}_j - \mathbf{x}_{j+1}) + \mathbf{f}_j = \mathbf{0} \qquad (3)$$

Two exceptions shall be considered for the generic equilibrium equation (3). Actually, the first and last nodes of the thrust line are loaded by the thrust forces associated to the first and last branches, respectively, and by the springers' reactions $\mathbf{r}_l$ and $\mathbf{r}_r$. Accordingly, the equilibrium equations of these two nodes read

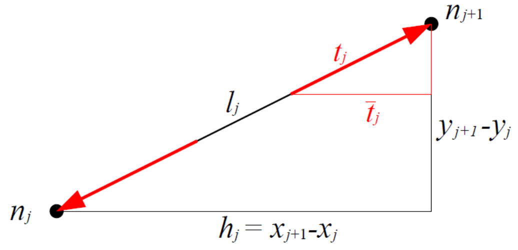

Figure 2: Horizontal thrust

$$\mathbf{r}_l + \frac{t_1}{\ell_1}(\mathbf{x}_1 - \mathbf{x}_2) + \mathbf{f}_1 = \mathbf{0}\,, \qquad \mathbf{r}_r + \frac{t_B}{\ell_B}(\mathbf{x}_N - \mathbf{x}_{N-1}) + \mathbf{f}_N = \mathbf{0} \quad (4)$$

These equations are easily solved for $\mathbf{r}_l$ and $\mathbf{r}_r$ as

$$\mathbf{r}_l = -\frac{t_1}{\ell_1}(\mathbf{x}_1 - \mathbf{x}_2) - \mathbf{f}_1\,, \qquad \mathbf{r}_r = -\frac{t_B}{\ell_B}(\mathbf{x}_N - \mathbf{x}_{N-1}) - \mathbf{f}_N \quad (5)$$

Thus, springers' reactions are easily computed after the thrust values of all branches and vertical coordinates of all nodes are determined. To this end, the set of equilibrium equations associated to internal nodes is solved first.

In order to simplify this set of equations, it is useful to introduce the so-called *reference thrusts* $\hat{t}_j = R\bar{t}_j$. They are proportional to the horizontal projection $\bar{t}_j$ of t_j by means of the positive scalar parameter $R > 0$. Such reference thrusts are introduced within equation (3) by setting

$$\frac{t_j}{\bar{t}_j} = \frac{\ell_j}{h_j} \quad \Leftrightarrow \quad t_j = \frac{\ell_j}{h_j}\bar{t}_j = \frac{\ell_j}{h_j}\frac{\hat{t}_j}{R} \qquad (6)$$

where $h_j = x_{j+1} - x_j$. Previous equation has been obtained by invoking the similitude of triangles in Figure 2.

Employing previous formula within Eq. (3), the equilibrium of the j-th internal node becomes

$$\frac{\hat{t}_{j-1}}{h_{j-1}}(\mathbf{x}_j - \mathbf{x}_{j-1}) + \frac{\hat{t}_j}{h_j}(\mathbf{x}_j - \mathbf{x}_{j+1}) + R\mathbf{f}_j = \mathbf{0} \tag{7}$$

This is the generic element of the set of equations used to perform the thrust line analysis.

Notice that the set of equations that is obtained by writing down (7) for all internal nodes is composed of $2(N-1) = 2N-2$ equations. The number of unknowns, that is represented by B reference thrusts, N vertical coordinates of nodes, and the value of R, exceeds by 2 the number of available equations. Actually, $B + N + 1 = 2N$. However, these unknowns are also subjected to a set of inequalities introduced before, i.e. $t_j > 0$, $y_{j,\min} \le y_j \le y_{j,\max}$ and $R > 0$.

This means that the set of equations and inequalities can admit infinite solutions, one unique solution or even no solutions. In the first case one is interested in determining two extreme thrust line configurations, respectively associated to a minimum and maximum value of the horizontal component of thrust at springers. Accordingly, these solutions are called solutions of minimum and maximum thrust. Recall that horizontal components of thrusts are inversely proportional to the parameter R, see, e.g., Eq. (6). Hence, these solutions are also associated to a maximum and minimum value of R, respectively. Furthermore, these two solutions correspond to the deepest and shallowest geometry of the thrust line. Hence, they are also called *deepest* and *shallowest* solutions.

In case the solution is unique, one can imagine that the interval between the deepest and the shallowest solutions is reduced in such a way that the two extreme cases coincide. This situation represents a limit condition for the equilibrium of the arch since only one equilibrated distribution of internal forces is possible. A small modification of the arch geometry or a modest alteration of the loading condition can result in an unsafe structure.

Finally, in case no solution exists, then there is no distribution of internal forces that equilibrate applied loads and fulfill the material assumptions. This means that the arch is unsafe.

The procedure for solving the set of equations (7), for $j = 2, \ldots, N-1$, and the corresponding inequalities, is particularly simplified in case all nodal forces are vertical. Actually, in this case it is possible to set up a procedure in which thrusts and vertical coordinates of nodes are uncoupled. This simpler case is described first, while the more general case of thrust lines subjected to both vertical and horizontal loads is presented later on.

```
 1  % --- No horizontal forces ---
 2
 3  % Inizialize reference thrusts of branches
 4  Tref=ones(N-1,1)*T0;
 5
 6  % Minimum thrust solution (deepest solution)
 7  [Yd,Rd,Td]=VertEq(X,Ymin,Ymax,Tref,Fy,-1);
 8
 9  % Maximum thrust solution (shallowest solution)
10  [Ys,Rs,Ts]=VertEq(X,Ymin,Ymax,Tref,Fy,1);
```

Figure 3: Solving procedure in absence of horizontal forces. The function `VertEq` is coded as in Figure 4.

2.1 Solving procedure for thrust lines subjected to vertical loads

In case all nodes of the thrust line are loaded exclusively by vertical forces, the determination of the thrust line configurations of minimum and maximum thrust is relatively simple. It is implemented by the few lines of MATLAB code reported in Figure 3.

In absence of horizontal forces, i.e. when $f_{jx} = 0$, $j = 1, \ldots, N$, the x component of equation (7) reads

$$\frac{\hat{t}_{j-1}}{h_{j-1}}(x_j - x_{j-1}) + \frac{\hat{t}_j}{h_j}(x_j - x_{j+1}) = 0 \tag{8}$$

Recalling that $h_{j-1} = x_j - x_{j-1}$ and $h_j = x_{j+1} - x_j$, the previous equation simplifies to

$$\hat{t}_{j-1} - \hat{t}_j = 0 \tag{9}$$

that leads to uniform distribution of reference thrusts in all branches. Accordingly, all reference thrusts can be set equal to an arbitrary small positive value $\hat{t}_0$. This is done at line 4 in Figure 3.

After the reference thrusts are assigned, the algorithm solves twice the vertical equilibrium equations to compute the solutions of minimum and maximum thrusts, respectively. This is done by invoking the function `VertEq` at lines 7 and 10 of Figure 3. The MATLAB code of this function is reported in Figure 4.

The vertical equilibrium of the generic j-th node is obtained by selecting the y component of equation (7), which reads

$$\frac{\hat{t}_{j-1}}{h_{j-1}}(y_j - y_{j-1}) + \frac{\hat{t}_j}{h_j}(y_j - y_{j+1}) + Rf_{jy} = 0 \tag{10}$$

The set of $N-1$ vertical equilibrium equations of all internal nodes are written in matrix form as

$$\mathbf{D}\mathbf{y} + R\mathbf{f}_y = \mathbf{0} \tag{11}$$

where $\mathbf{y}$ is the vector collecting the y coordinates of all nodes, $\mathbf{f}_y$ collects the vertical component of the forces applied at nodes $n_2 \ldots, n_{N-1}$ and $\mathbf{D}$ is the $N-2 \times N$ matrix collecting the *reference thrust densities* $\hat{t}_j/h_j$ of all branches. It

```
1  function [y,R,T]=VertEq(X,Ymin,Ymax,Tref,Fy,pm)
2  % Solve vertical equilibrium of the thrust line
3  % pm = -1  -> Minimum thrust solution (deepest solution)
4  % pm = +1  -> Maximum thrust solution (shallowest solution)
5
6  N=length(X);
7
8  % Matrix of reference thrust densities
9  D0=zeros(N);
10 Db0=[1,-1;-1,1];
11 for j=1:N-1
12     jj=j:j+1;
13     D0(jj,jj)=D0(jj,jj)+Db0*Tref(j)/(X(j+1)-X(j));
14 end
15
16 % select internal nodes
17 D=D0(2:N-1,:);
18 fy=Fy(2:N-1);
19
20 % Constrained linear optimization
21 options=optimset('Display','none','Algorithm','dual-simplex');
22 F=[zeros(N,1);pm];
23 Aeq=[D,fy];
24 beq=zeros(N-2,1);
25 lb=[Ymin;0];
26 ub=[Ymax;Inf(1)];
27 [Sol,~,ExitFlag]=linprog(F,[],[],Aeq,beq,lb,ub,[],options);
28
29 if ExitFlag==1
30     % Output solution
31     y=Sol(1:end-1);
32     R=Sol(end);
33 else
34     % No solution found
35     y=-ones(N,1);
36     R=-1;
37 end
38
39 % Evaluate actual thrusts
40 T=Tref;
41 for j=1:N-1
42     Th=Tref(j)/R;
43     Tv=Th*(y(j+1,1)-y(j,1))/(X(j+1,1)-X(j,1));
44     T(j)=norm([Th,Tv]);
45 end
```

Figure 4: MATLAB function VertEq: solution of the vertical equilibrium of a thrust line

is obtained by assembling the contribution of each branch, that is

$$\mathbf{D}_j = \frac{\hat{t}_j}{h_j}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad (12)$$

It is assembled in rows j and $j+1$ and columns j and $j+1$ of $\mathbf{D}$ (lines 8 – 18 of Figure 4).

The unknowns $\mathbf{y}$ and R are determined by solving Eq. (11) together with the set of inequalities that express the corresponding constrains. Hence, $\mathbf{y}$ and R are computed by solving the following constrained linear optimization problem:

$$\min_{\mathbf{y},r} \pm R \text{ such that } \begin{cases} \mathbf{D}\mathbf{y} + R\mathbf{f}_y = \mathbf{0} \\ \mathbf{y}_{\min} \leq \mathbf{y} \leq \mathbf{y}_{\max} \\ R > 0 \end{cases} \qquad (13)$$

by means of the Dual-Simplex Algorithm [42] already available in MATLAB. It is invoked at lines 20 – 27 of Figure 4.

Notice that the objective function in (13) is either $-R$ or $+R$, depending whether the optimization problem is used to maximize or minimize R. Either one of the two cases is selected by setting `pm = -1` or `pm = +1` to the last input variable of `VertEq` (lines 7 and 10 in Figure 3). Actually, from equation (6) it is clear that higher values of R are associated to lower values of thrust and *vice versa*. Accordingly, when the objective function $-R$ is used, the value of R is maximized and the optimization procedure returns the so-called *solution of minimum thrust* or *deepest solution*. Similarly, the *solution of maximum thrust* or *shallowest solution* is obtained when R is minimized, i.e. when the objective function is set as $+R$.

```
1  % --- Horizontal forces ---
2
3  % ++ Minimum thrust solution (deepest solution) ++
4  Rd=1e-8;
5  for k=2:100
6
7      % Assign reference thrusts that fulfill horizontal equilibrium
8      Tref=RefThr(Fx,N,T0,Rd);
9
10     % Vertical equilibrium
11     Rold=Rd;
12     [Yd,Rd,Td]=VertEq(X,Ymin,Ymax,Tref,Fy,-1);
13
14     % Check convergence
15     if (Rd==-1)||(abs(Rd-Rold)<1e-3*Rd)
16         break
17     end
18 end
19
20 % ++ Maximum thrust solution (shallowest solution) ++
21 Rs=1e-8;
22 for k=2:100
23
24     % Assign reference thrusts that fulfill horizontal equilibrium
25     Tref=RefThr(Fx,N,T0,Rs);
26
27     % Vertical equilibrium
28     Rold=Rs;
29     [Ys,Rs,Ts]=VertEq(X,Ymin,Ymax,Tref,Fy,1);
30
31     % Check convergence
32     if (Rs==-1)||(abs(Rs-Rold)<1e-3*Rs)
33         break
34     end
35 end
```

Figure 5: Solving procedure in presence of horizontal forces. The code of functions `VertEq` and `RefThr` are reported in Figure 4 and 6, respectivley.

2.2 Solution procedure for thrust lines subjected to horizontal and vertical loads

When nodes are loaded by horizontal and vertical forces, both the horizontal and vertical components of the equilibrium equations of nodes contain the unknown R. Hence, the horizontal and vertical equations are coupled and cannot be solved in sequence. Thus an iterative procedure that solves repeatedly the horizontal and vertical equilibrium of the thrust line needs to be adopted. The corresponding MATLAB code is reported in Figure 5 : lines 3 – 18 compute the deepest solution and lines 20 – 35 the shallowest one. Similarly to the case of null horizontal forces the only difference between these two portions of code regards the sign of the objective function used to solve the vertical equilibrium equations of nodes.

The mentioned iterative procedures start by choosing an arbitrary tentative value of R so that the horizontal equilibrium equation can be solved for the reference thrusts $\hat{t}_j$ of nodes. This is done at lines 4 and 21.

For any iterative estimate of R, including this first guess, the horizontal equilibrium equation of the generic j-th node becomes

$$\hat{t}_{j-1} - \hat{t}_j + Rf_{jx} = 0 \qquad \Leftrightarrow \qquad \hat{t}_j = \hat{t}_{j-1} + Rf_{jx} \tag{14}$$

Accordingly, the reference thrust $\hat{t}_j$ associated to a generic branch can be computed as a function of the reference thrust $\hat{t}_{j-1}$ of the previous branch. Hence, after choosing an arbitrary value of reference thrust for the very first branch, all successive thrusts can be evaluated in sequence. Of course, this solution does not necessarily fulfill the constraint $t_i > 0,\ i = 1, \ldots, B$. This issue is easily solved by modifying the tentative distribution of reference thrusts by adding $t_{i\ \min}$ to all thrust values.

This set of positive reference thrusts that fulfill the horizontal equilibrium equations and the relevant constraints are computed by invoking the function `RefThr` (lines 8 and 25 in Figure 5). The corresponding MATLAB code is reported in Figure 6.

Once reference thrusts are assigned, it is possible to evaluate the nodal heights $\mathbf{y}$ and a new estimate of R by solving the constrained linear optimization (13). This is done by invoking the function `VertEq` (lines 12 and 29 of Figure 5), which has been already described in the previous section.

This new estimate of R is then used again to evaluate a new estimate of reference thrusts by repeating the same set of operations.

```
function T=RefThr(fx,N,T0,R)
% Assign reference thrusts

T=ones(N-1,1)*T0;

for j=2:N-1
    T(j)=T(j-1)+fx(j)*R;
end

if min(T)<=0
    T=T-min(T)+T0;
end
```

Figure 6: Evaluation of positive reference thrusts that fulfill the horizontal equilibrium equations.

The procedure is iterated until the relative difference between two successive estimates of R is lower than a given tolerance, that is

$$\frac{R_{new} - R_{old}}{R_{new}} < tol_R \tag{15}$$

Such a convergence check is coded at lines 15 and 32 of Figure 5.

3 Numerical examples

Two numerical examples are reported hereafter, regarding the thrust line analysis of both a pointed arch and a lowered circular arch. Both problems are solved by the code contained in the MATLAB scripts `ExampleArch`, also available from the link reported earlier. The function `ArchLab` implements the TLA by the algorithm described in section 2, the two table files `PointedArch.csv` and `LoweredArch.csv` contain the input data for the examples regarding the analysis of the lowered and pointed arches and the plotting function `PlotArch` is used to graphically represent the solutions.

The function `ArchLab` takes as input the horizontal position x of the nodes of the thrust line, their height limits, $y_{\min}$ and $y_{\max}$, and the applied forces, f_x and f_y. These data are included in the two table files `PointedArch.csv` and `LoweredArch.csv`, respectively. The geometry and the loading condition of these arches is determined as a function of a few geometric and loading parameters that are (Figure 7): the arch span S, rise H and thickness t; its unitary weight F_{sw}; uniformly distributed load F_u; the height Y_f and unitary weight of filling F_f.

Geometric and loading parameters of the two models are reported in Table 1. Being planar models, unitary weights and distributed loads are assigned as force per unit area and force per unit length, respectively. In order to stress out the algorithm, the thrust lines of both arches has been generated by using a horizontal spacing between nodes equal to one hundredth of the span. This resulted in a thrust model composed of 122 nodes and 121 branches.

The intrados of the pointed arch is the composition of two circles of radius $D/2 = H^2/S + S/4 \approx 2.08$ m. Their centers are positioned at the height of the arch springers and distant $D/2 - S/2 \approx 0.58$ m on both sides of the axis of symmetry. The arch is loaded by horizontal forces which are pro-

Table 1: Geometric and loading parameters of the pointed and lowered arches taken as examples.

	Pointed arch	Lowered arch
S [m]	3.0	3.0
H [m]	2.0	1.0
t [m]	0.3	0.3
F_{sw} [kN/m^2]	20.0	20.0
F_u [kN/m]	0.0	10.0
Y_f [m]	1.8	1.5
F_f [kN/m^2]	15.0	15.0

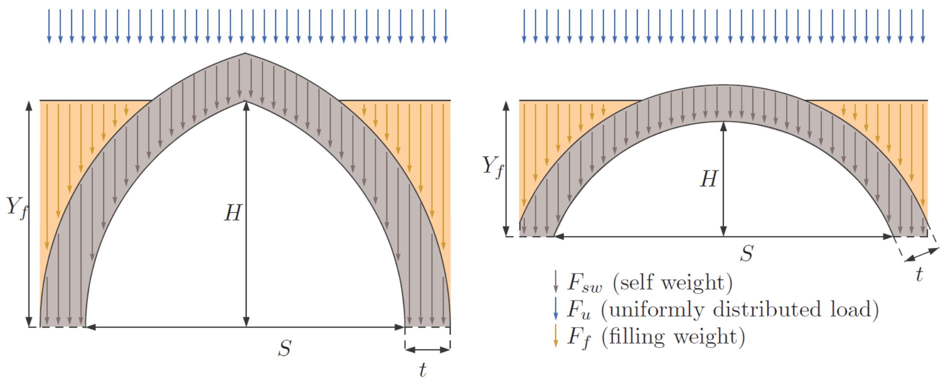

Figure 7: Pointed and lowered arches

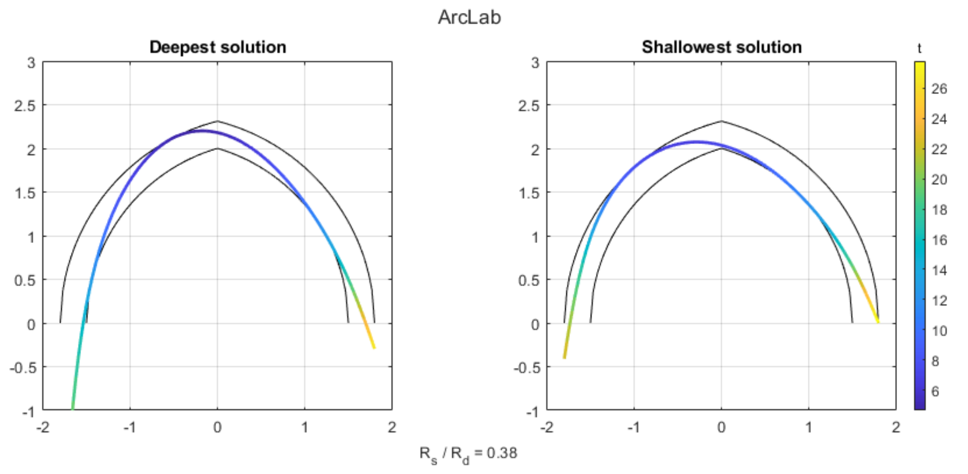

Figure 8: Deepest and shallowest solutions for a pointed arch subjected to vertical and horizontal loads

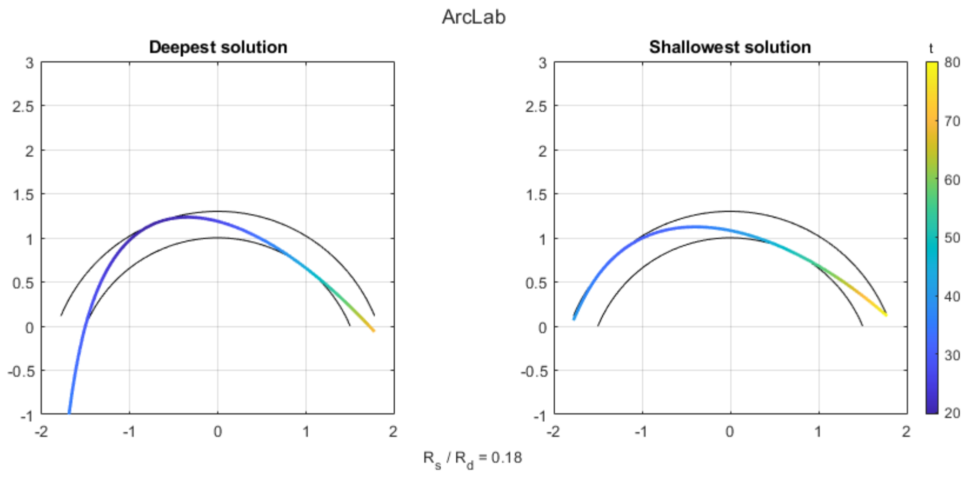

Figure 9: Deepest and shallowest solutions for a lowered arch subjected to vertical and horizontal loads

portional to the vertical ones by a factor $\lambda = f_x/f_y = 0.15$. The corresponding solutions of minimum and maximum thrust are reported in Figure 8. Within this figure, the values of the scalar parameter R associated to the deepest and shallowest solutions are respectively indicated by R_d and R_s. The ratio $R_s/R_d = 0.38$ is an estimate of the difference between the two solutions. The more this ratio is different from unity, the larger is the geometric safety factor of the arch.

The intrados of the lowered circular arch is described by a unique circle of radius $D/2 = H/2 + S^2/8H = 1.625$ m, while its center lies $H - D/2 = 0.625$ m below the arch springers. Similarly to the previous case, horizontal forces are set as proportional to the vertical ones by a factor $\lambda = f_x/f_y = 0.5$. The deepest and shallowest configurations of the thrust line are diagrammed in Figure 9. Also in this case, the even lower value of the ratio $R_s/R_d = 0.18$ shows the capability of lowered arches to withstand higher values of horizontal forces.

4 Conclusions

The function `ArchLab` is used to determine thrust line configurations of minimum and maximum thrust of masonry arches subjected to both vertical and horizontal loads. It represents a specialization of the Thrust Network Analysis to the case of a planar line of thrust. The implemented algorithm has been described in full detail in section 2 where a precise reference to the implemented MATLAB code is also reported. When only vertical loads are applied, the algorithm is capable to determine the thrust line configurations without the need of any recursive method. Conversely, when also horizontal loads are applied, the procedure re-

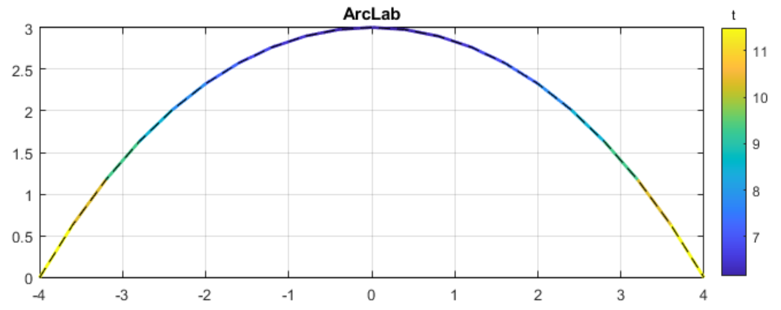

Figure 10: Optimal shape of an arch under its own self-weight and comparison with a catenary curve (dashed)

quires iterations. In both cases, the algorithm computes the horizontal components of reference thrust in closed form at each iteration, while the height of nodes is determined by solving a constrained linear optimization problem by using the MATLAB builtin implementation of the Dual-Simplex algorithm.

By implementing additional few lines of code, not commented in this paper for brevity, it is possible to employ the same functions described earlier to solve a form-finding problem. It is sufficient to set $y_{\min} = 0$ and $y_{\max} = H$ for all nodes of the model and compute the deepest configuration of the thrust line. Here H represents the arch axis height. An iterative procedure can be used to update the self-weight of each branch of the thrust line and compute updated nodal forces accordingly. An example of such an iterative algorithm is coded within the script `ExampleFF`, which finds the optimal shape of an arch of span $S = 8.0$ m and rise $H = 3.0$ m, subjected to its own self-weight. The solution is obtained after only 4 iterations and is then plotted by using the function `PlotFF`. It is reported in Figure 10, where it is compared with the catenary curve $y(x) = H + a - \text{Cosh}(x/a)$, with $a = 3.0668$ m, showing perfect agreement.

Funding information: The author states no funding involved

Conflict of interest: The author states no conflict of interest

References

[1] Accornero F, Lacidogna G. Safety Assessment of Masonry Arch Bridges Considering the Fracturing Benefit. Appl Sci (Basel). 2020;10(10):3490.
[2] Accornero F, Lacidogna G, Carpinteri A. Medieval arch bridges in the Lanzo Valleys, Italy: incremental structural analysis and fracturing benefit. J Bridge Eng. 2018;23(7):05018005.
[3] Addessi D, Marfia S, Sacco E, Toti J. Modeling Approaches for Masonry Structures. Open Civ Eng J. 2014;358(1):288–300.
[4] Alexakis H, Makris N. Minimum thickness of elliptical masonry arches. Acta Mech. 2013;224(12):2977–91.
[5] Alexakis H, Makris N. Limit equilibrium analysis and the minimum thickness of circular masonry arches to withstand lateral inertial loading. Arch Appl Mech. 2014;84(5):757–72.
[6] Alexakis H, Makris N. Limit equilibrium analysis of masonry arches. Arch Appl Mech. 2015;85(9-10):1363–81.
[7] Angelillo M, Cardamone L, Fortunato A. A numerical model for masonry-like structures. J Mech Mater Struct. 2010;5(4):415–583.
[8] Angelillo M, Babilio E, Fortunato A. Singular stress fields from masonry-like vaults. Contin Mech Thermodyn. 2013;25(2-4):423–41.
[9] Benvenuto E. Vaulted Structures and Elastic Systems. An introduction to the history of structural mechanics. Volume II. NY: Springer-Verlag. 1991.
[10] Block P, Ciblac T, Ochsendorf J. Real-time limit analysis of vaulted masonry buildings. Comput Struc. 2006;84(29-30):551841–52.
[11] Block P, DeJong M, Ochsendorf J. As hangs the flexible line: equilibrium of masonry arches. Nexus Netw J. 2006;8(2):13–24.
[12] Block P, Ochsendorf J. Thrust network analysis: a new methodology for threedimensional equilibrium. J Int Assoc Shell Spat Struct. 2007;48:167–73.
[13] Cascini L, Gagliardo R, Portioli F. LiABlock_3D: a software tool for collapse mechanism analysis of historic masonry structures. Int J Archit Herit. 2018;14(1):75–94.
[14] Cavalagli N, Gusella V, Severini L. Lateral loads carrying capacity and minimum thickness of circular and pointed masonry arches. Int J Mech Sci. 2016;115:70645–56.
[15] Chiozzi A, Malagù M, Tralli A, Cazzani A. ArchNURBS: NURBS-based tool for the structural safety assessment of masonry arches in MATLAB. J Comput Civ Eng. 2016;30(2):04015010.
[16] Coccia S, Di Carlo F, Rinaldi Z. Collapse displacements for a mechanism of spreading-induced supports in a masonry arch. Int. J. Advanc. Struct. Eng. 2015;7(3):307–20.
[17] Como M. Statics of historic masonry constructions. 2nd ed. NY: Springer. 2016.
[18] Cuomo M, Ventura G. A complementary energy formulation of no tension masonry-like solids. Comput Methods Appl Mech Eng. 2000;189(1):313–39.
[19] Forgács T, Sarhosis V, Bagi K. Minimum thickness of semi-circular skewed masonry arches. Eng Struct. 2017;5140:317–36.
[20] Fraddosio A, Lepore N, Piccioni MD. Thrust Surface Method: an innovative approach for the three-dimensional lower bound Limit Analysis of masonry vaults. Eng Struct. 2020;202:109846.

[21] Fraternali F. A thrust network approach for the equilibrium problem of unreinforced masonry vaults via polyhedral stress functions. Mech Res Commun. 2010;37(2):198–204.

[22] Galassi S, Misseri G, Rovero L, Tempesta G. Failure modes prediction of masonry voussoir arches on moving supports. Eng Struct. 2018;173:706–17.

[23] Galassi S, Tempesta G. The Matlab code of the method based on the Full Range Factor for assessing the safety of masonry arches. MethodsX. 2019 Jun;6:1521–42.

[24] Gilbert M, Melbourne C. Rigid-block analysis of masonry structures. Struct Eng. 1994;72(21):356–61.

[25] Heyman J. The stone skeleton. Int J Solids Struct. 1966;2(2):249–79.

[26] Heyman J. Structural Analysis, a Historical approach. Cambridge Univ. Press. 1998.

[27] Huerta S. Mechanics of masonry vaults: The equilibrium approach. Historical Constructions. Possibilities of numerical and experimental techniques. Guimaraes, Portugal: Universidade do Minho. 2001;47–69.

[28] Huerta S. The Analysis of Masonry Architecture: A Historical Approach. Archit Sci Rev. 2008;51(4):297–328.

[29] Kooharian A. Limit analysis of voussoir (segmental) and concrete archs. J Am Concr Inst. 1952;24(4):317–28.

[30] Lemos JV. Discrete element modeling of masonry structures. Int J Archit Herit. 2007;1(2):190–213.

[31] Livesley RK. Limit analysis of structures formed from rigid blocks. Int J Numer Methods Eng. 1978;12(12):1853–71.

[32] Lourenco PB, Milani G, Tralli A, Zucchini A. Analysis of masonry structures: review of and recent trends in homogenization techniques. Can J Civ Eng. 2007;34(11):1443–57.

[33] Milani G, Lourenco PB. 3D non-linear behavior of masonry arch bridges. Comput Struc. 2012;110:133–50.

[34] Marmo F, Rosati L. Reformulation and extension of the thrust network analysis. Comput Struc. 2017;182:104–18.

[35] Marmo F, Masi D, Rosati L. Thrust network analysis of masonry helical staircases, Int J Arch Her. 502018,12(5):828-848.

[36] Marmo F, Masi D, Mase D, Rosati L. Thrust network analysis of masonry vaults. Int J Masonry Research Innovat. 2019;4(1-2):64–77.

[37] Marmo F, Ruggieri N, Toraldo F, Rosati L. Historical study and static assessment of an innovative vaulting technique of the 19th century. Int J Archit Herit. 2019;13(6):799–819.

[38] Michiels T, Napolitano R, Adriaenssens S, Glisic B. Comparison of thrust line analysis, limit state analysis and distinct element modeling to predict the collapse load and collapse mechanism of a rammed earth arch. Eng Struct. 2017;148:145–56.

[39] Michiels T, Adriaenssens S. Form-finding algorithm for masonry arches subjected to in-plane earthquake loading. Comput Struc. 2018;195:85–98.

[40] Nikolić D. Thrust line analysis and the minimum thickness of pointed masonry arches. Acta Mech. 2017;228(6):2219–36.

[41] Nikolić D. Catenary arch of finite thickness as the optimal arch shape. Struct Multidiscipl Optim. 2019;60(5):1957–66.

[42] Nocedal J, Wright SJ. Numerical Optimization, 2nd ed., Springer Ser. in Op. Research, Springer-Verlag. 2006.

[43] Nodargi NA, Bisegna P. Thrust line analysis revisited and applied to optimization of masonry arches, Int J Mech Sci. 2020;179;105690.

[44] O'Dwyer D. Funicular analysis of masonry vaults. Comput Struc. 1999;73(1-5):187–97.

[45] Ricci E, Fraddosio A, Piccioni MD, Sacco E. A new numerical approach for determining optimal thrust curves of masonry arches. Eur J Mech A, Solids. 2019;75:426–42.

[46] Rigó B, Bagi K. Discrete element analysis of stone cantilever stairs. Meccanica. 2018;53(7):1571–89.

[47] Roca P, Cervera M, Gariup G, Pela L. Structural Analysis of Masonry Historical Constructions. Classical and Advanced Approaches. Arch Comput Methods Eng. 2010;17(3):299–325.

[48] Tempesta G, Galassi S. Safety evaluation of masonry arches. A numerical procedure based on the thrust line closest to the geometrical axis. Int J Mech Sci. 2019;155:206–21.

[49] Tóth AR, Orbán Z, Bagi K. Discrete element analysis of a stone-masonry arch. Mech Res Commun. 2009;36(4):469–80.