Limit analysis of frictional block assemblies as a mathematical program with complementarity constraints

M.C. Ferris
Computer Sciences Department
University of Wisconsin
1210 West Dayton Street, Madison
Wisconsin 53706, U.S.A.

F. Tin-Loi*
School of Civil and Environmental Engineering
University of New South Wales
Sydney 2052, Australia

February 15, 1999

Abstract

The computation of the collapse loads of discrete rigid block systems, characterized by frictional (nonassociative) and tensionless contact interfaces, is formulated and solved as a special constrained optimization problem known as a Mathematical Program with Equilibrium Constraints (MPEC). In the present instance, some of the essential constraints are defined by a complementarity system involving the orthogonality of two sign-constrained vectors. Due to its intrinsic complexity, MPECs are computationally very hard to solve. In this paper, we investigate a simple numerical scheme, involving appropriate relaxation of the complementarity term, to solve this nonstandard limit analysis problem. Some computational results are presented to illustrate potentialities of the method.

Keywords: Limit analysis, friction, mathematical programming.

1 Introduction

The analysis of masonry structures has been the subject of a rich literature spanning over the last few hundred years, as indicated by Heyman in his classical treatises on the subject [1, 2]. Of particular importance is the limit analysis of block structures with a

*Corresponding author (email: f.tinloi@unsw.edu.au).

nonassociative type of contact interface law, typical of the commonly assumed Coulomb friction behavior.

Drucker [3] was perhaps the first to point out the problem of applying the classical bound theorems of plasticity to frictional problems, just after, it appears, his student Kooharian [4] had described the behavior of segmental arches under hinging only in terms of limit analysis. However, without doubt, the precursor of systematic and modern computational methods to deal with the collapse load evaluation of block structures with nonassociative material is the seminal paper of Livesley [5] who attempted to solve the problem as a Linear Programming (LP) problem using the classical lower bound formulation of limit analysis. That work also showed that adoption of a simplified associated constitutive law not only, as expected, runs the risk of providing an incorrect collapse mechanism, but more importantly may give an overestimate of the true collapse load. Livesley suggested a "postprocessing" of the results to provide what he believes to be a correct mechanism but did not offer any remedy for limit load overestimation. Following Livesley's work, which was subsequently extended to three-dimensional block systems [6], a number of related investigations have been carried out. Most notable in recent years are: the work of Boothby and Brown [7, 8, 9] in establishing stability criteria based on extremum characterization of some energy functional; experimental and theoretical (albeit assuming normality by treating sliding as Drucker's plastic shearing analogy) corroboration by Melbourne and Gilbert [10, 11] that frictional considerations are especially important in multiring arches; an excellent thesis by Fishwick [12] concerned with automatic numerical schemes for limit analysis of rigid block structures involving nonassociative friction; and, similarly, computer-oriented mathematical programming approaches by Baggio and Trovalusci [13] for carrying out the same task.

In spite of vigorous research, as illustrated by the foregoing representative achievements, the computation of the collapse load under nonassociative slip is still an open problem. Fishwick's enumerative method [12] to solve the underlying Mixed Complementarity Problem (MCP) — a mathematical program involving a system of orthogonal sign-constrained (or complementary) vectors, see e.g. [14] — appears capable of providing the absolute (global) minimum collapse load but only for a small number of blocks. Baggio and Trovalusci [13], instead of searching for the minimum load factor of an MCP as in [12], attempted a direct minimization under complementarity constraints and experienced severe computational difficulties for reasonable size block systems. They had to resort to an *a priori* assumption on the distribution of contact forces to achieve convergence. In fact, they even finally recommend use of an associated law, leading to a more tractable (but in our view inappropriate) LP problem.

The primary objective of the present paper is to outline a simple numerical scheme suitable for solving the limit analysis problem for large-scale block structures. Our discrete formulation is straightforward; the difficulty lies in its solution. Using a nodal approach for ease of automatic assembly of appropriate matrix-vector quantities (rather than the perhaps more compact but equivalent mesh formulation), we gather as constraints all governing conditions (statics, kinematics, nonassociative constitution and the requirement of positive dissipation by the live loads) associated with our problem in point, and set up an optimization problem involving minimization of the load factor. By itself, the set of constraints is fully equivalent to the mesh-based MCP considered by Fishwick [12], whereas our optimization problem, known in the mathematical programming literature as a Mathematical Program with Equilibrium Constraints (MPEC) [16] is in effect another form of the optimization problem considered by Baggio and Trovalusci [13]. The feature (and difficulty) of an MPEC lies in the presence of nonconvex complementarity constraints, with the consequence that the limit analysis problem may have multiple local minima.

This paper is organized as follows. In the next section, we present the governing relations of our discrete model leading naturally, in Section 3, to a number of mathematical programming formulations, depending on the associativity assumption. In particular, for a nonassociative law, the governing relations yield an MCP whose solution provides an upper bound on the collapse load. The search for the best (minimum) load factor can be cast as an MPEC. Moreover, when normality is assumed, we also show, using standard mathematical programming theory, how the MCP splits into a pair of classical dual LP problems. Motivated by simplicity and our recent, successful experiences in solving other types of MPEC-related structural problems [17, 18], we then propose (Section 4) a numerical algorithm capable of solving the MPEC. The key idea is suitable relaxation of the complementarity term. In the following Section 5, we give an idea of the potentialities of the algorithm by presenting computational results on a number of reasonably large problems. Comparative solution times for solving the relevant MPECs, MCPs and LPs are provided as well as collapse load values and sketches of collapse mechanisms. We also briefly describe the tools and environments used for modeling and solving our mathematical programming based problems. Finally, we conclude with some pertinent remarks in Section 6.

A note regarding notation: column vectors are assumed throughout; vectors and matrices are denoted by boldface lower case and upper case symbols, respectively; transposition is indicated by the superscript T; a null vector is represented by $\mathbf{0}$; kinematic quantities (displacements and strains) are assumed to be in rate form but are denoted, for clarity, without the normal superimposed dot.

2 Discrete model and governing relations

The discrete block model we adopt is a popular and often the most appropriate idealization for masonry-type structures. Its main mechanical features are: rigid blocks; contact interfaces that cannot resist tension; provision for blocks to slide (without separation, if desired, as required by a nonassociated law) and/or to overturn when some limits are reached; and unlimited compressive strength at interfaces. Two comments are worthy of note. First, some of these features are assumptions that we have adopted rather than shortcomings of the model; it would be easy to incorporate, for instance, the ability to carry tension, limited compressive strength and even partial contact at the interfaces. Second, as mentioned in [19], such models are particularly appropriate for analyzing ancient, historical masonry structures characterized by a complex system of stones either dry-assembled or connected by poor quality mortar. Experimental tests [20] validate the use of such a discrete, rather than homogeneous and isotropic, model as it was observed that the global behavior of such assemblies is strongly influenced by their discrete nature, namely, size, disposition and orientation of essentially rigid blocks in frictional-unilateral contact with one another.

We now proceed to develop the governing equations for our frictional block structure. For this purpose, consider the representative discrete model shown in Fig. 1. As in [5], we treat the blocks as nodes and the interfaces as elements of a conventional finite element discretization. A nodal approach is adopted, in preference to the usually more compact mesh formulation [12], for ease of automatic generation of problem data.

Assume that three degrees of freedom are associated with the centroid of each block. In turn, three pairs of equal and opposite stress resultants act at each contact interface, leading to the force system shown in Fig. 1 for a typical block j. For each interface, the stress resultants are the transverse (shear) force t, the normal force n, and a bending moment measure $\bar{m}$ (defined as the bending moment m per half the corresponding contact length w, i.e. $\bar{m} = m/w$). For a model with b blocks and c contacts, let $\mathbf{f}$ be the $3b$-vector of applied nodal forces and $\mathbf{x}$ the vector of length $3c$ that collects (in the order of contact interface numbering) all stress resultants. Then, equilibrium of the whole structure can be expressed, through the constant $3b \times 3c$ equilibrium matrix $\mathbf{A}$ (whose transpose is known as the "compatibility" matrix), as

$$\mathbf{A}\mathbf{x} = \mathbf{f} = \mathbf{f}_D + \alpha\mathbf{f}_L, \tag{1}$$

where the nodal loads $\mathbf{f}$, as indicated, are conceived as the sum of known dead loads $\mathbf{f}_D$ and unknown live loads $\alpha\mathbf{f}_L$, in which α is an unknown (scalar) proportional load factor that amplifies the known vector $\mathbf{f}_L$ of basic live loads. We need not detail the calculation of matrix $\mathbf{A}$, but simply mention that this can be automatically carried out in conventional finite element fashion through assembly, using say location vectors identifying the contact interfaces, of elemental equilibrium matrices pertaining to individual blocks. Incidentally, for the model shown in Fig. 1, three block types can be clearly identified, namely, a full base-course block with 5 contacts, a full block with 6 contacts, and a half block with 4 contacts.

We now consider the kinematics of the collapse. Let $\mathbf{u}$ be the $3b$-vector of nodal unconstrained displacement rates corresponding to the nodal loads $\mathbf{f}$. Also, the stress vector $(t, n, \bar{m})$ for each contact interface is related (in a virtual work sense) to a strain rate vector $(\gamma, \varepsilon, \bar{\theta})$ describing, in order, the corresponding relative joint sliding, separation and rotation ($\bar{\theta} = \theta w$). We can thus define a $3c$-vector $\mathbf{q}$, ordered as for $\mathbf{x}$, which collects all such contact strain rates. For the assumed small displacement regime, geometric compatibility is then ensured at the structure level if

$$\mathbf{q} = \mathbf{A}^T\mathbf{u}. \tag{2}$$

Crucial to the formulation is a proper description of the constitutive laws that govern the behavior of the contact interfaces. This follows [12] classical Coulomb friction laws and can be elegantly described in the same fashion as classical plasticity relations (e.g. [21, 22]). For a generic contact interface, we can thus, in direct analogy to plasticity, define in the space of the static (stress) variables a set of limit (yield) conditions that delineate failure due to sliding and/or rocking. For clarity, we map these limit surfaces, pertaining to the two types of failure modes, separately as shown in Fig. 2. Any stress state contained within the cone formed by the limit surfaces for sliding and rocking represents a combination that is considered safe. On the other hand, a stress state on a limit surface will lead to a critical condition for which the contact interface is active and has developed (or about to develop) positive strain rates. The possible directions of such strain rates are also indicated in Fig. 2, for the case of activation in the positive quadrants.

¿From Fig. 2, with the angles ϕ and ψ (the latter normally assumed to be 45°) defined as indicated, the limit conditions for a generic i-th contact interface can be written explicitly as

$$\begin{bmatrix} y_{s+} \\ y_{s-} \\ y_{r+} \\ y_{r-} \end{bmatrix} = - \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ -\cos\phi & -\sin\phi & 0 \\ 0 & -\sin\psi & \cos\psi \\ 0 & -\sin\psi & -\cos\psi \end{bmatrix} \begin{bmatrix} t \\ n \\ \bar{m} \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

or more compactly as

$$\mathbf{y}^i = -\mathbf{N}^{iT}\mathbf{x}^i \geq \mathbf{0}. \tag{3}$$

The nonnegative vector $\mathbf{y}^i$ can be considered to be a vector of yield functions (with subscript s indicating sliding, r rocking, $+$ positive rocking and sliding, and $-$ negative sliding and rocking); geometrically it represents a vector of orthogonal distances from a stress point to the limit hyperplanes.

Further, for contact i, the strain rates contained in $\mathbf{q}^i$ are related (Fig. 2) to the respective (obviously) nonnegative resultant strain rates (analogous to plastic multipliers in classical plasticity) in $\mathbf{z}^i$ as follows:

$$\begin{bmatrix} \gamma \\ \varepsilon \\ \bar{\theta} \end{bmatrix} = \begin{bmatrix} \cos\phi_o & -\cos\phi_o & 0 & 0 \\ -\sin\phi_o & -\sin\phi_o & -\sin\psi & -\sin\psi \\ 0 & 0 & \cos\psi & -\cos\psi \end{bmatrix} \begin{bmatrix} z_{s+} \\ z_{s-} \\ z_{r+} \\ z_{r-} \end{bmatrix}, \quad \begin{bmatrix} z_{s+} \\ z_{s-} \\ z_{r+} \\ z_{r-} \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

or,

$$\mathbf{q}^i = \mathbf{V}^i\mathbf{z}^i, \quad \mathbf{z}^i \geq \mathbf{0}, \tag{4}$$

where we have assumed that a nonassociative "flow" law, in accordance with a Coulomb-type frictional model, governs sliding behavior (i.e. the resultant strain rate z_s is not normal to the sliding limit surface; it is only so if $\phi_o = \phi$, whereas $\phi_o = 0$ models ideal, nonassociated Coulomb friction). Normality, however, is adopted for rocking behavior so that the strain rate z_r is perpendicular (Fig. 2) to the limit surface for rocking.

To complete the description of the interface constitutive laws, we need to impose a condition on the vectors $\mathbf{y}^i$ and $\mathbf{z}^i$ so that any component of $\mathbf{z}^i$ is positive only when the stress point lies on the corresponding yield plane. Again, as in plasticity [21], this can be expressed through the complementarity relation

$$\mathbf{y}^{iT}\mathbf{z}^i = 0 \tag{5}$$

which also holds componentwise, in view of the nonnegativity of all elements of vectors $\mathbf{y}^i$ and $\mathbf{z}^i$.

Extension of relations (3)–(5) to the whole structure is straightforward. This can be done by a simple reinterpretation of the quantities involved as new indexless symbols, in particular elemental vectors and matrices as appropriate concatenated vectors and block diagonal matrices, respectively. For example, $\mathbf{y}^T \equiv (\mathbf{y}^{1T} \ldots \mathbf{y}^{cT})$, $\mathbf{V} \equiv \mathrm{diag}(\mathbf{V}^1, \ldots, \mathbf{V}^c)$, etc.

Finally, the last condition needed to describe collapse of the structure requires that the displacement rates $\mathbf{u}$, which define the onset of the collapse motion, must cause positive dissipation to be produced by the live loads $\mathbf{f}_L$, i.e. $\mathbf{f}_L^T\mathbf{u} > 0$. This requirement can be conveniently normalized [23] as

$$\mathbf{f}_L^T\mathbf{u} = 1. \tag{6}$$

3 Mathematical programming formulations

We are now in a position to formulate precisely the limit analysis problem for frictional rigid block assemblages. This is achieved by simply collecting all conditions (statics, kinematics, constitutive relations, and positivity of dissipated work) that describe the collapse of such systems. Thus, from the relations developed in the previous section, we obtain, after some rearrangement, the following system:

$$\begin{bmatrix} . & . & \mathbf{f}_L^T & . \\ . & . & -\mathbf{A}^T & \mathbf{V} \\ -\mathbf{f}_L & \mathbf{A} & . & . \\ . & -\mathbf{N}^T & . & . \end{bmatrix} \begin{bmatrix} \alpha \\ \mathbf{x} \\ \mathbf{u} \\ \mathbf{z} \end{bmatrix} - \begin{bmatrix} . \\ . \\ . \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \\ \mathbf{f}_D \\ \mathbf{0} \end{bmatrix}, \tag{7}$$
$$\mathbf{y} \geq \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0}, \quad \mathbf{y}^T\mathbf{z} = 0,$$

where dots (.) represent zero quantities (scalars, vectors or matrices) of appropriate size. This particular problem is known is an MCP [14] — a class of mathematical programming problems that has been vigorously researched over the last decade or so, from both theoretical and computational viewpoints. In fact, apart from the relationship

$\mathbf{y}^T\mathbf{z} = 0$, this problem only involves linear relationships amongst the variables and is thus called a linear mixed complemenarity problem. However, adaptations of Lemke's method [15], the standard technique for linear complementarity problems, is only known to process (7) when $\mathbf{V} = \mathbf{N}$. In our case, uniqueness of the load multiplier α is not guaranteed and any solution of the MCP will yield an upper bound to the collapse limit.

In the case of fully associated contact laws, however, normality of the resultant strain rates are ensured so that $\mathbf{V} = \mathbf{N}$, leading to a (skew) symmetric system for MCP (7). As is well-known [24], the static and kinematic variables become uncoupled and the MCP can be recognized as being the necessary and sufficient optimality (Karush-Kuhn-Tucker) conditions of a pair of dual LP problems with common (unique) optimal values of α. Mechanically, the LPs are well-known expressions of the bounds theorems of plasticity. In particular [23], the LP related to the static theorem is given by

$$\begin{aligned} &\text{maximize} && \alpha \\ &\text{subject to} && -\alpha\mathbf{f}_L^T + \mathbf{A}\mathbf{x} = \mathbf{f}_D, \\ &&& -\mathbf{N}^T\mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{8}$$

whereas the LP arising from the kinematic theorem is

$$\begin{aligned} &\text{minimize} && -\mathbf{f}_D^T\mathbf{u} \\ &\text{subject to} && \mathbf{f}_L^T\mathbf{u} = 1, \\ &&& -\mathbf{A}^T\mathbf{u} + \mathbf{N}^T\mathbf{z} = \mathbf{0}, \\ &&& \mathbf{z} \geq \mathbf{0}. \end{aligned} \tag{9}$$

Let us return to MCP (7). Since the collapse limit is not unique, then it would be desirable to calculate the minimum value of the set of load factor solutions to the MCP. For small-size problems, it may be possible to find the best solution to the MCP by exhaustive enumeration [12]. However, this technique is not possible for large-size MCPs. A better solution is to attempt a direct minimization (e.g. [13]). This can be posed as the following optimization problem:

$$\begin{aligned} &\text{minimize} && \alpha \\ &\text{subject to} && \mathbf{f}_L^T\mathbf{u} = 1, \\ &&& -\mathbf{A}^T\mathbf{u} + \mathbf{V}^T\mathbf{z} = \mathbf{0}, \\ &&& -\alpha\mathbf{f}_L^T + \mathbf{A}\mathbf{x} = \mathbf{f}_D, \\ &&& \mathbf{y} = -\mathbf{N}^T\mathbf{x}, \\ &&& \mathbf{y} \geq \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0}, \quad \mathbf{y}^T\mathbf{z} = 0, \end{aligned} \tag{10}$$

which is a special case of an MPEC [16] in which the equilibrium system takes the form of a complementarity condition. Clearly, the constraints in (10) are exactly the MCP given by (7). At variance with Baggio and Trovalusci [13], we do not attempt to simplify the constraints of the MPEC (as they do through a Gauss-Jordan transform) since we intend to use sophisticated mathematical programming tools (modeling systems and associated solvers) to automatically carry out the reduction and account for any sparsity patterns.

Finally a note regarding MPECs is appropriate. Whilst an extensive theory of first and second order optimality conditions for MPECs has been developed in [16], still relatively little is known about the numerical solution of practical, large-scale MPECs likely to arise in realistic applications. The most prominent feature of an MPEC, and one that distinguishes it from a standard nonlinear program, is the presence of complementarity constraints. These constraints classify this class of mathematical programs as a nonlinear disjunctive (or piecewise) program and therefore carries with it a "combinatorial curse". This makes it very difficult to solve, especially if one wishes, as ideally required in the present instance, to compute a global optimal solution. A branch-and-bound technique can be adopted to perform an exhaustive enumeration in the search for a global optimum, but, as mentioned, is obviously severely limited in the size of problem it can handle. Nearly all methods proposed to date [16] are aimed at finding stationary solutions and/or local optima, and are categorized roughly by the way the complementarity condition is handled.

4 A relaxation algorithm for solving the MPEC

We propose, in the following, a simple and intuitive reformulation of (10) involving the use of standard, readily available nonlinear programming (NLP) solvers. A primary motivation behind this scheme is to exploit the availability of state-of-the-art and industry-standard solvers such as CONOPT2 [25], especially from within the powerful GAMS (an acronym for General Algebraic Modeling System) modeling environment [26] adopted in this work to facilitate the modeling and solution process.

The attempt to formulate and solve an MPEC as a nonlinear program, it must be noted, is carried out in spite of the fact that traditional constraint qualifications are never satisfied [16], with the implication that the usual numerical methods for solving NLP problems may be expected to have some difficulties in their solution. Also, whilst there is no guarantee that the solution provided represents a local minimum to the MPEC (let alone a global minimum), we wish to investigate numerically if our simple scheme is computationally feasible for large-size structures and can provide reasonable solutions in practice.

As indicated earlier, the difficulty in solving the MPEC lies in the presence of the nonconvex complementarity constraints. The basic idea underlying the relaxation method for solving MPEC (10) consists in simply relaxing the complementarity term, allowing $\mathbf{y}^T\mathbf{z} \leq \mu$, for some relaxation parameter μ. The MPEC is thus converted to the following standard NLP problem:

$$
\begin{aligned}
\text{minimize} \quad & \alpha \\
\text{subject to} \quad & \mathbf{f}_L^T \mathbf{u} = 1, \\
& -\mathbf{A}^T \mathbf{u} + \mathbf{V}^T \mathbf{z} = \mathbf{0}, \\
& -\alpha \mathbf{f}_L^T + \mathbf{A}\mathbf{x} = \mathbf{f}_D, \qquad (11) \\
& \mathbf{y} = -\mathbf{N}^T \mathbf{x}, \\
& \mathbf{y} \geq \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0}, \\
& \mathbf{y}^T \mathbf{z} \leq \mu.
\end{aligned}
$$

The relaxed problem is solved for successively smaller values of μ to force the complementarity term, which is nonnegative at feasible points of (11), to approach zero. The attraction of this method is that each subproblem is a standard nonlinear program and general purpose codes such as CONOPT2 [25] can be used.

An alternative penalty problem method that also solves a sequence of nonlinear programs has been successfully used in solving some minimum weight [18] and parameter identification problems [17]. This technique could also be used in this case, although some limited computational testing showed the relaxation method to perform better on the class of problems described here.

The following pseudocode further clarifies the algorithm:

Set initial μ (e.g. 10^{-3}), maximum number of iterations (*maxiter*), and solve the MCP (7) to determine initial values for the variables.

```
for i = 1 to maxiter
    if y^T z ≤ 10^-10 exit
    solve nonlinear program (11)
    set μ = μ/2
end
```

At the start of the solution, we solve the MCP (7) to determine initial values for the problem variables. The remainder of the algorithm can be considered as a local neighbourhood improvement mechanism. At the solution of the MCP, the complementarity error is zero. The algorithm relaxes this constraint, allowing the nonlinear programming code to search in the neighbourhood of the given complementary point for a point with better objective value. Typically, a new point is found that has a lower value for α but which is no longer complementary. The algorithm then slowly drives the parameter μ to zero to recover a new complementary solution. Since a modeling language efficiently implements restarts from a given solution, the nonlinear programs are typically solved fairly quickly. Whilst this heuristic does not guarantee a global minimum, our computational experiments indicate that improvements on the collapse limit can be achieved in some instances.

5 Computational results

In this section, we report on some computational results concerning the limit analyses of block assemblages. We implemented the models within the GAMS mathematical programming modeling environment and solved them using various GAMS solvers. Before detailing our results, a note concerning both GAMS and the solvers we used would be useful. For more detailed information on GAMS and its associated solvers, the interested reader is referred to the GAMS Corporation website: http://www.gams.com.

It is commonly stated that data manipulation requirements limit mathematical programming applications more than optimization requirements. The typical end-user is generally more concerned with model formulation, representation and solution than with the details of the mathematical techniques involved. There is thus a strong case for making the solution phase as simple as possible while at the same time allowing for easy construction of large and complex models. This aim provided the impetus for the development of modeling languages of which GAMS is one.

GAMS [26] had its origin at the Development Research Center of the World Bank in Washington. It is a high level declarative language for formulating small to very large mathematical programming models using simple and concise algebraic statements which mirror the actual mathematical constructs involved. A GAMS model is transparent to both human and computer, is easily modified and moved across different computing platforms from notebooks to mainframes, and is independent of the solution algorithm of the mathematical programming solvers. It not only frees the model builder from the burdens imposed by the solution phase but also takes over the steps required for generation of the model. In addition to providing simplicity and compactness of model construction, it possesses important capabilities such as an internal efficient sparse data representation and automatic differentiation.

A number of mathematical programming problems types can be solved via GAMS. In addition to the LP, MCP and NLP models — problem classes which concern the present work — solution procedures are available for MIP (mixed integer programming), RMIP (relaxed mixed integer programming), MINLP (mixed integer nonlinear programming), RMINLP (relaxed mixed integer nonlinear programming) and CNS (constrained nonlinear systems). GAMS is continually evolving and adapted as new algorithms and problem classes have been explored.

Paucity of space precludes us from detailing the structure and construction of GAMS models. We refer the interested reader to the extensive GAMS library of models (from such diverse areas as economics, chemical engineering, trade, etc.) accessible from the GAMS website, and to [17, 18] for simple examples of GAMS models pertaining to the penalty approach for solving an MPEC. Inspection of GAMS models, even by someone not familiar with the syntax, will immediately show a close resemblance to the actual formulations, such as the MCP (7). A GAMS file is written using a standard text editor and executed through a simple "`gams <file name>`" command. The "`solve`" statement (e.g. "`solve block using lp maximizing obj`" where "`block`" is the name of the model and "`obj`" is the objective function to be maximized) invokes the appropriate solver, in this case an LP solver. In our case, we used, from within the GAMS (version 2.50.094) environment, GAMS/CPLEX (version 6.0) to solve the LP problem (8), GAMS/PATH (version 4.0) to solve the MCP (7) and GAMS/CONOPT2 (version 2.070F) to solve the NLP problem (11). All three solvers are large-scale, industry-standard optimizers.

CPLEX can solve LP problems using several alternative algorithms (primal simplex, dual simplex, or barrier) which are all designed for large, difficult problems where other LP solvers fail or are unacceptably slow. The CPLEX solvers have the reputation of being exceptionally fast and robust, providing high reliability even for poorly scaled or numerically difficult problems. We used the default state-of-the-art modified primal simplex [27] option with default settings. PATH is an implementation of a stabilized Newton method for the solution of the suitably transformed MCP as a set of nonsmooth equations [28, 29]. It uses standard large-scale simplex technology to help in the path search for the solution. PATH has become, since its introduction in 1995, the standard against which new large-scale MCP solvers are compared. CONOPT2, the newer version of an NLP code [25] based on the generalized reduced gradient idea, has powerful preprocessing features and maintains feasibility during its iterations, making it particularly robust and efficient.

For our computational testing, we developed a generic GAMS model to carry out the limit analysis of two-dimensional block assemblages such as those considered by Baggio and Trovalusci [13]. This allowed the three mathematical programming problems, namely, LP problem (8), MCP (7) and NLP problem (11), to be solved for various structural arrangements of blocks.

Basic details of all models are: blocks of (full) size 4×1.75 and (half) size 2×1.75; $\phi = \tan^{-1} 0.65$; $\psi = 45\,^\circ$; $\phi_o = \phi$ (associated, for LP problems) or $\phi_o = 0\,^\circ$ (nonassociated, for MCP and MPEC problems). All blocks are subjected to vertical (downward) self weight and horizontal (left to right) live loads, simulating an earthquake-type loading. In particular, for each j-th full block $\mathbf{f}_D^{jT} = (0, -1, 0)$ and $\mathbf{f}_L^{jT} = (\alpha, 0, 0)$; and hence for each j-th half block $\mathbf{f}_D^{jT} = (0, -1/2, 0)$ and $\mathbf{f}_L^{jT} = (\alpha/2, 0, 0)$.

All runs were carried out on a Win95-based 333MHz Pentium-II. We report on six different sets of runs representing different structural configurations (for conciseness, we omit diagrams of initial configurations but give deformed configurations for the MPEC runs later) very similar to those in [13]. Table 1 summarizes the results obtained. Reported are problem sizes (number of blocks b and number of contacts c), limit loads α and total computing time in secs, corresponding to solutions of LP problem (8), MCP (7) and NLP (labelled as "MPEC") problem (11). Also, the percentage difference (of MPEC limit loads) between MPEC and LP solutions are indicated by the column headed "% diff".

Table 1: Computational results

Example	Size $b \times c$	LP α	LP secs	MCP α	MCP secs	MPEC α	MPEC % diff	MPEC secs
1	33 × 83	0.64286	1.5	0.64285	1.3	0.63898	0.6	4.9
2	55 × 141	0.58000	2.0	0.56368	4.0	0.55742	4.1	9.3
3	46 × 102	0.37383	1.6	0.31078	1.5	0.31078	20.3	4.8
4	55 × 116	0.33195	1.7	0.26374	2.0	0.26374	25.9	5.6
5	61 × 120	0.23964	1.7	0.21584	2.4	0.20863	14.9	6.1
6	146 × 345	0.34782	5.7	0.29725	35.6	0.29577	17.6	232.0

The limit loads obtained by solving an MPEC are generally smaller than the corresponding MCP formulation. In turn, the MCP results are smaller than those given by the LP formulation which assumes associativity. We should note, however, that it may be possible to improve on the MCP solutions by using different starting vectors, as was done in [30] in the context of capturing multiplicity of solutions in quasibrittle fracture processes. We have not carried this out; our starting vectors for the MCP runs were set to zero in all cases.

All algorithms were run using their default parameters, and the relaxation algorithm was coded (in GAMS) precisely as indicated in the previous section. Even though there is no theoretical guarnatee of convergence of the MCP and MPEC approaches for (7) and (11) respectively, the codes solved every instance of the problems presented to them. We believe that the solution process we outline in this paper, although tailored to the problem instance, is generically implemented, and has great potential for use in other problems.

The assumption of associativity (in the LP problems) produced higher collapse loads (up to about 25.9% higher for Example 4), but an advantage is that CPLEX can carry out the limit analyses very efficiently. Moreover, it is interesting to note that the PATH execution times for solving the MCPs are comparable to those of CPLEX for the smaller problems (Examples 1–5). For Example 6, CPLEX is about six times faster, although in absolute terms PATH can still be considered as being very efficient (about 36 secs to solve that example). As expected, the solution of the relaxed NLP problems is computationally more demanding, especially in the case of the larger size Example 6. However, the absolute times for solving the MPECs are still remarkably good, considering that they include an initial MCP solve (time as indicated under the "MCP" column) as well. The computational results are particularly encouraging especially in view of the difficulties encountered for similar problems by Baggio and Trovalusci [13].

The collapse mechanisms extracted from the nonassociated MPEC solutions are shown in Figs. 3–8. These plots (as well as visual checks of input GAMS data) were carried out within MATLAB using a recently developed GAMS-MATLAB link [31]. This useful facility enables MATLAB users to access the optimization capabilities of GAMS, and allows visualization of GAMS models directly within MATLAB.

6 Conclusions

This paper is concerned with an important and difficult class of limit analysis problems involving rigid block assemblages in frictional contact. The problem is cast in the first instance as an MCP. The search for the best upper bound then leads to an optimization problem involving complementarity constraints, or an MPEC.

Motivated by the need for simple, yet robust, approaches to solve this problem for practical, often large-scale structures, we attempt to take advantage of the increased availability of advanced and powerful software (and hardware) by proposing a simple algorithm with the potential of solving our problem via the GAMS modeling language and an associated nonlinear programming solver CONOPT2.

The algorithm is based on a relaxation approach that attempts to drive the complementarity term to zero. Computational testing within the GAMS environment indicates the viability of this approach. Comparison with the results of an MCP formulation shows that the MPEC formulation is likely give better solutions, albeit at some computational expense. Assumption of associativity leads to easy to solve LP problems but furnishes higher collapse loads, as expected.

This paper has been primarily concerned with solving the proposed MPEC. Useful extensions of the present work, made possible by the positive conclusions reached regarding the MPEC approach, include: extensive parametric studies regarding different block sizes and dispositions; consideration of other structural types such as arch bridges; modeling of actual masonry-type structures; and extension to three-dimensional structures — a task which should pose formal rather than conceptual difficulties. From the computational viewpoint, it would be worthwhile to carry out more extensive testing of the MPEC algorithm on similar and other problem types, and to investigate use of the more efficient MCP formulation, coupled with some robust and efficient search strategy. Of course, a challenging goal will always be the search for the global minimum of the MPEC, especially for structures with a large number of blocks.

Acknowledgments

This research was supported by the Australian Research Council and the National Science Foundation. We would also like to thank Dr. R.J. Fishwick (University of Portsmouth) for kindly providing us with a copy of his PhD thesis and Dr. P. Trovalusci (University of Rome "La Sapienza") for supplying references [19, 20].

References

[1] Heyman, J. The masonry arch. Chichester: Ellis Horwood, 1982.

[2] Heyman, J. The stone skeleton: structural engineering of masonry architecture. Cambridge: Cambridge University Press, 1995.

[3] Drucker, D.C. Coulomb friction, plasticity and limit loads. Journal of Applied Mechanics, 1953, 21:71–74.

[4] Kooharian, A. Limit analysis of voussoir (segmental) and concrete arches. ACI Journal, 1952, 23:317–328.

[5] Livesley, R.K. Limit analysis of structures formed from rigid blocks. International Journal of Numerical Methods in Engineering, 1978, 12:1853–1871.

[6] Livesley, R.K. A computational model for the limit analysis of three-dimensional masonry structures. Meccanica, 1992, 27:161–172.

[7] Boothby, T.E. and Brown, C.B. Stability of masonry piers and arches. Journal of Engineering Mechanics, ASCE, 1992, 118:367–383.

[8] Boothby, T.E. and Brown, C.B. A general lower and upper bound theorem of static stability. Engineering Structures, 1993, 15;189–196.

[9] Boothby, T.E. Stability of masonry piers and arches including sliding. Journal of Engineering Mechanics, ASCE, 1994, 120:304–319.

[10] Gilbert, M. and Melbourne, C. Rigid-block analysis of masonry structures. The Structural Engineer, 1994, 72:356–360.

[11] Melbourne, C. and Gilbert, M. The behaviour of multiring brickwork arch bridges. The Structural Engineer, 1995, 73:39–47.

[12] Fishwick, R.J. Limit analysis of rigid block structures. PhD thesis, Department of Civil Engineering, University of Portsmouth, 1996.

[13] Baggio, C. and Trovalusci, P. Limit analysis for no-tension and frictional three-dimensional discrete systems. Mechanics of Structures and Machines, 1998, 26:287–304.

[14] Dirkse, S.P. and Ferris, M.C. MCPLIB: a collection of nonlinear mixed complementarity problems. Optimization Methods and Software, 1995, 5:319–345.

[15] Lemke, C.E. and Howson, J.T. Equilibrium Points of Bimatrix Games. SIAM Journal on Applied Mathematics, 1964, 12:413–423.

[16] Luo, Z.Q., Pang, J.S. and Ralph, D. Mathematical programs with equilibrium constraints. Cambridge: Cambridge University Press, 1996.

[17] Ferris, M.C. and Tin-Loi, F. Nonlinear programming approach for a class of inverse problems in elastoplasticity. Structural Engineering and Mechanics, 1998, 6:857–870.

[18] Ferris, M.C. and Tin-Loi, F. On the solution of a minimum weight elastoplastic problem involving displacement and complementarity constraints. Computer Methods in Applied Mechanics and Engineering, 1999 (in print).

[19] Baggio, C. and Trovalusci, P. Discrete models for jointed masonry walls. In: Proceedings, The Sixth North American Masonry Conference, Vol. 2, Philadelphia, 1993, 939–949.

[20] Trovalusci, P. No-tension discrete model with friction for jointed block masonry walls using interface elements. In: Proceedings, ANSYS Technology Conference and Exhibition, Vol. II, Pittsburgh, 1992, 11.73–11.82.

[21] Maier, G. A matrix structural theory of piecewise linear elastoplasticity with interacting yield planes. Meccanica, 1970, 5:121–130.

[22] Maier, G. Shakedown theory in perfect elastoplasticity with associated and nonassociated flow laws: a finite element, linear programming approach. Meccanica, 1969, 4:250–260.

[23] Maier, G. and Nappi, A. A theory of no-tension discretized structural systems. Engineering structures, 1990, 12:227–234.

[24] Lloyd Smith, D. Plastic limit analysis. In: Lloyd Smith, D., editor. Mathematical programming methods in structural plasticity. New York: Springer-Verlag, 1990, 61–82.

[25] Drud, A. CONOPT – a large-scale GRG code. ORSA Journal on Computing, 1994, 6:207–216.

[26] Brooke, A., Kendrick, D. and Meeraus, A. GAMS: a user's guide, release 2.25. Massachusetts: Boyd & Fraser Publishing Company, 1992.

[27] Bixby, R.E. Implementing the simplex method: the initial basis. ORSA Journal on Computing, 1992, 4:267–284.

[28] Dirkse, S.P. and Ferris, M.C. The PATH solver: a non-monotone stabilization scheme for mixed complementarity problems. Optimization Methods and Software, 1995, 5:123–156.

[29] Ferris, M.C. and Munson, T.S. Complementarity problems in GAMS and the PATH solver. Journal of Economic Dynamics and Control, 1999 (forthcoming).

[30] Tin-Loi, F. and Ferris, M.C. Holonomic analysis of quasibrittle fracture with nonlinear softening. In: Karihaloo, B.L., Mai, Y.W., Ripley, M.I. and Ritchie, R.O., editors. Advances in fracture research. Oxford: Pergamon, 1997, 2183–2190.

[31] Ferris, M.C. MATLAB and GAMS: interfacing optimization and visualization software. Technical Report TR98-19, Computer Sciences Department, University of Wisconsin, Madison.

LIST OF FIGURES

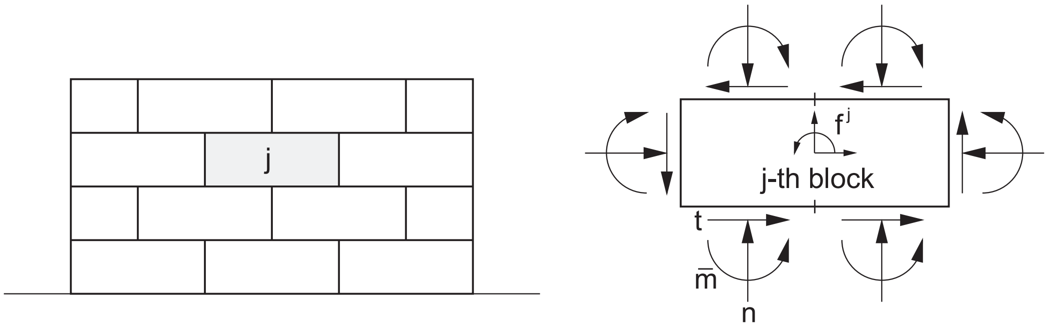
j
f^j
j-th block
t
$\bar{m}$
n

Fig. 1

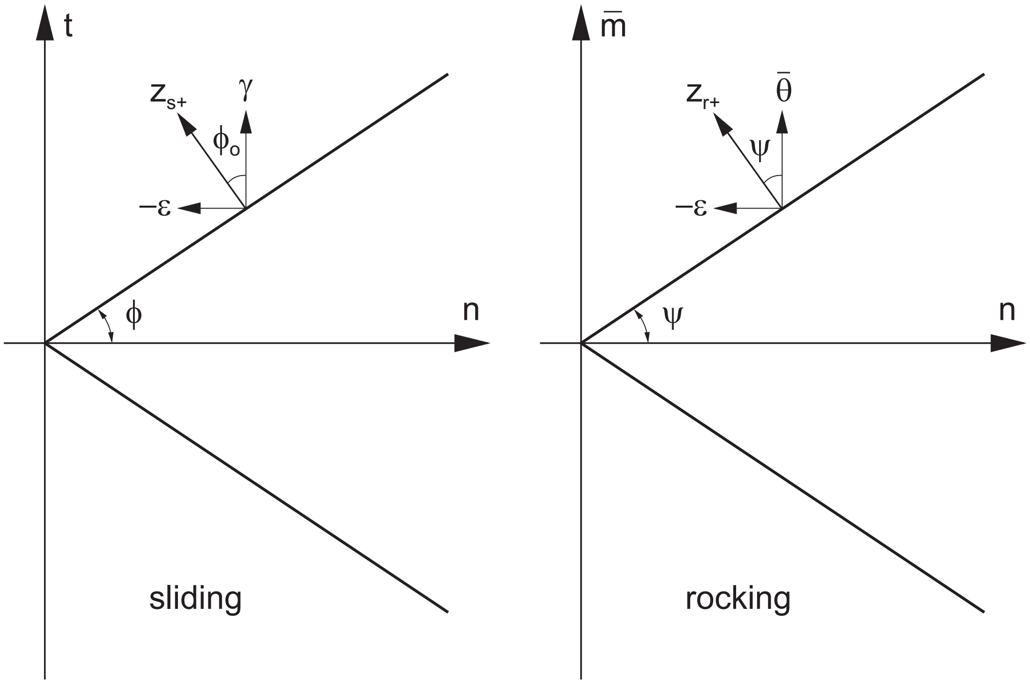
t
z_{s+}
γ
ϕ_o
$-\varepsilon$
ϕ
n
sliding
$\bar{m}$
z_{r+}
$\bar{\theta}$
ψ
$-\varepsilon$
ψ
n
rocking

Fig. 2

Nonassociated : α = 0.63898

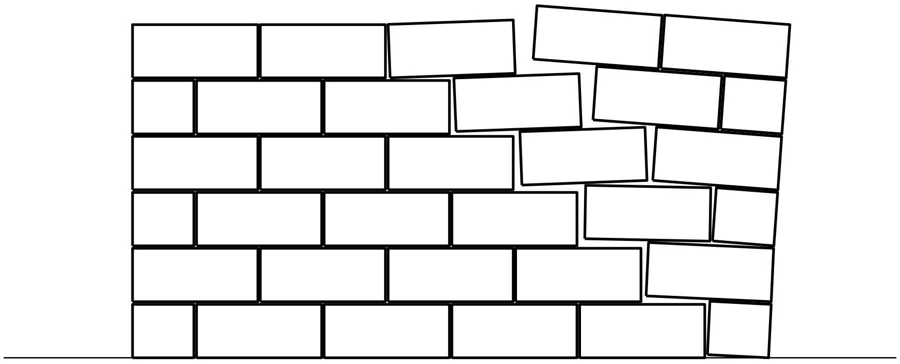

Fig. 3

Nonassociated : α = 0.55742

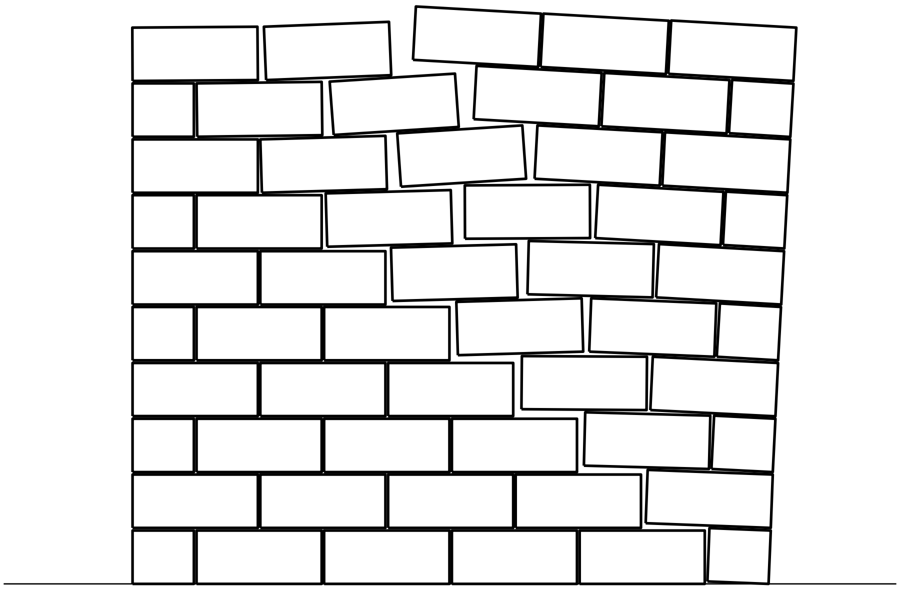

Fig. 4

Nonassociated : α = 0.31078

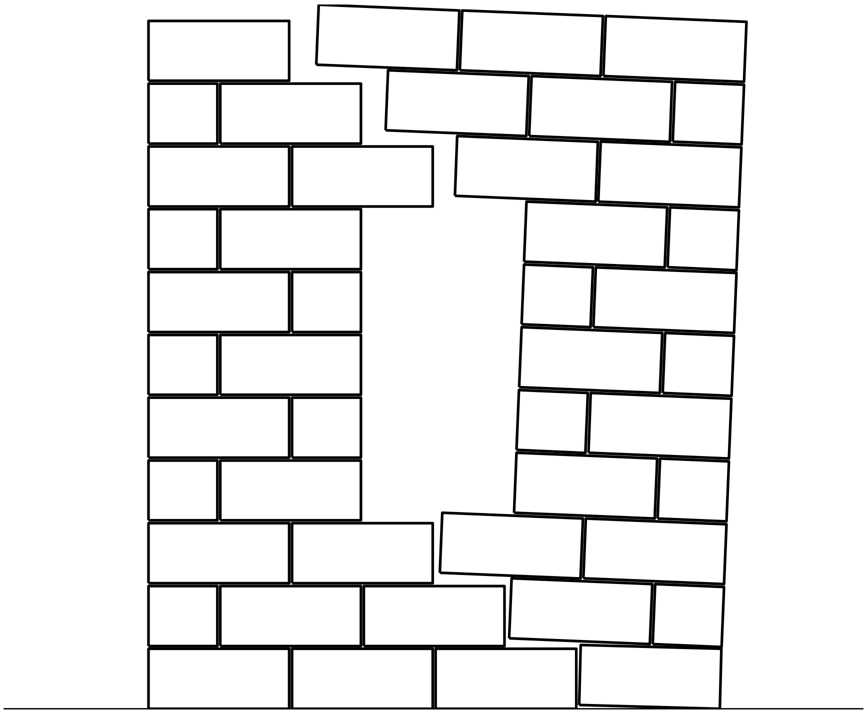

Fig. 5

Nonassociated : α = 0.26374

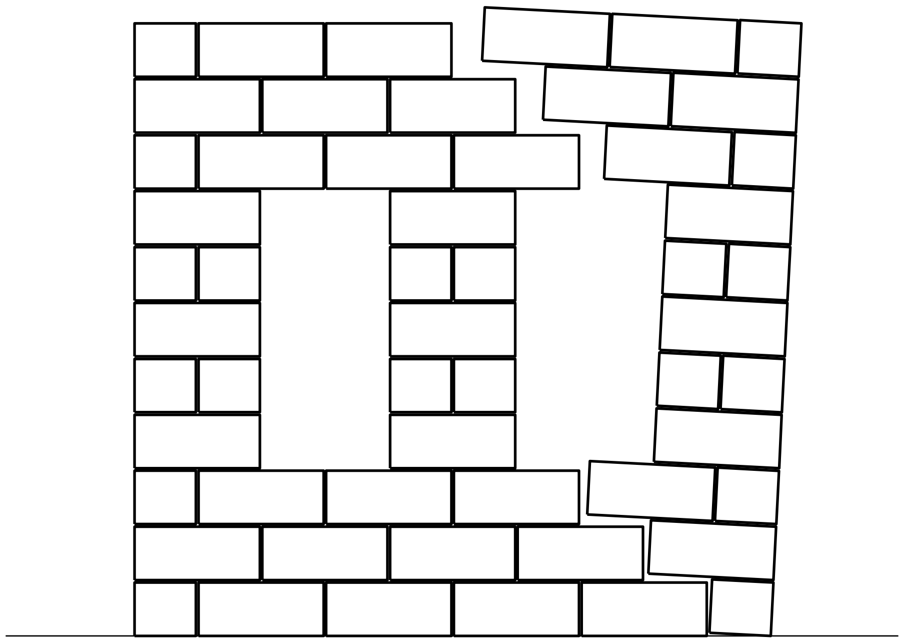

Fig. 6

Nonassociated : α = 0.20863

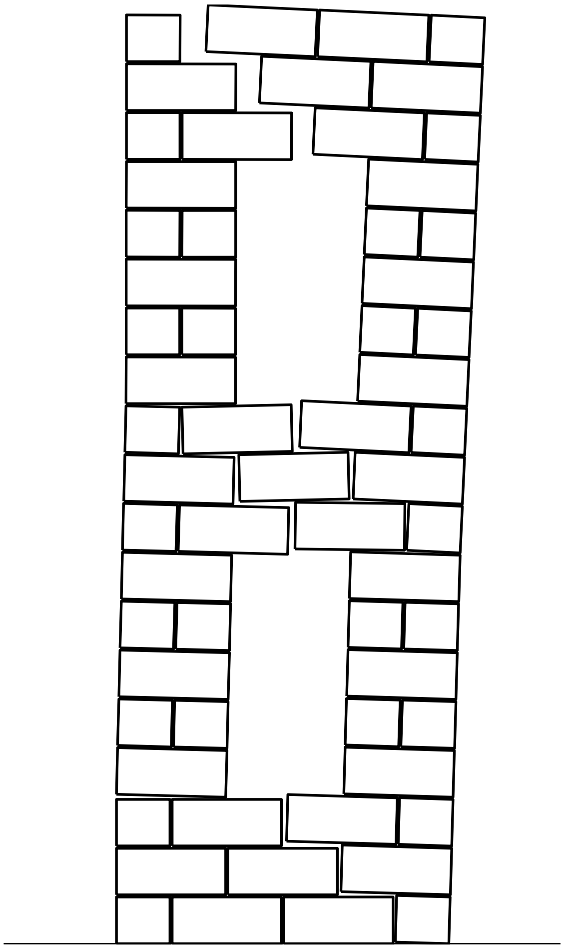

Fig. 7

Nonassociated : α = 0.29577

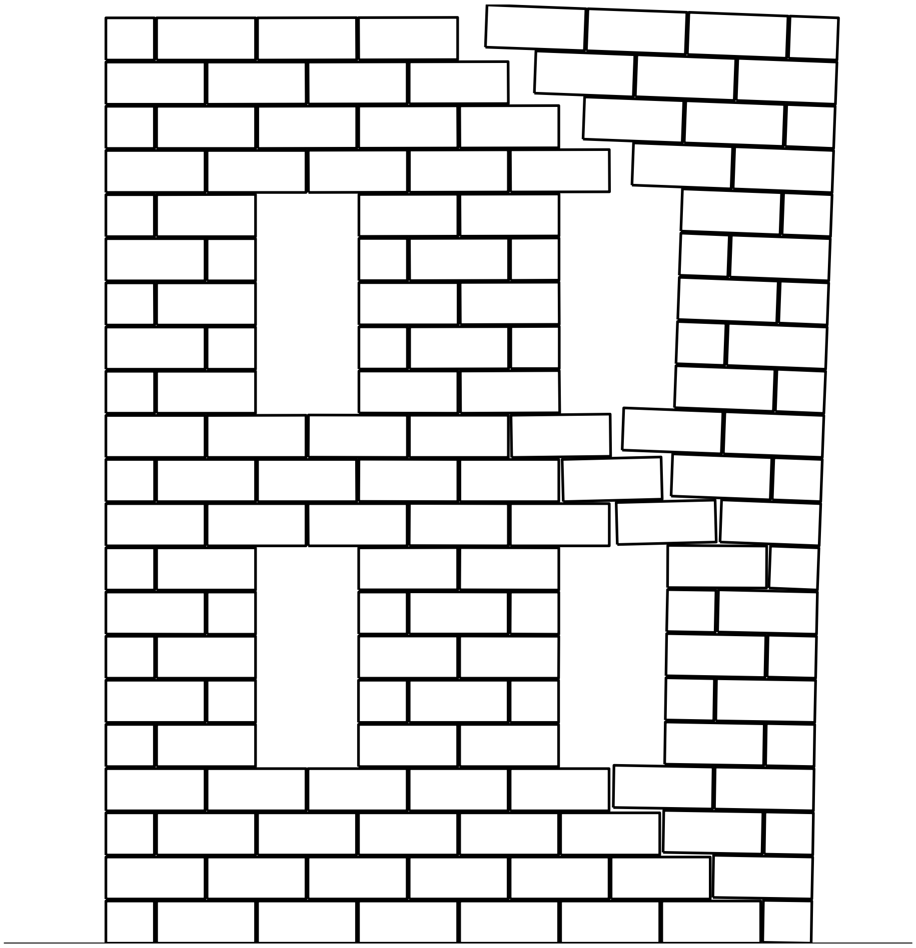

Fig. 8