A greedy algorithm for yield surface approximation

Jérémy Bleyer, Patrick De Buhan

▶ **To cite this version:**

HAL Id: hal-00850868

https://hal-enpc.archives-ouvertes.fr/hal-00850868

Submitted on 9 Aug 2013

A greedy algorithm for yield surface approximation

Jérémy Bleyer[a], Patrick de Buhan[a]

[a]*Université Paris-Est, Laboratoire Navier,*
Ecole des Ponts ParisTech-IFSTTAR-CNRS (UMR 8205),
6-8 av Blaise Pascal, Cité Descartes, 77455 Champs-sur-Marne, FRANCE

Abstract

This Note presents an approximation method for convex yield surfaces in the framework of yield design theory. The proposed algorithm constructs an approximation using a convex hull of ellipsoids such that the approximate criterion can be formulated in terms of second-order conic constraints. The algorithm can treat bounded as well as unbounded yield surfaces. Its efficiency is illustrated on two yield surfaces obtained using up-scaling procedures.

To cite this article: J. Bleyer, P. de Buhan, C. R. Mecanique 341 (2013), pp. 605-615

Résumé

Un algorithme glouton pour l'approximation des surfaces de rupture Cette Note présente une méthode d'approximation pour les convexes de résistance dans le cadre de la théorie du calcul à la rupture. L'algorithme proposé construit une approximation utilisant une union convexe d'ellipsoides de sorte que le critère approché puisse être formulé à l'aide de contraintes coniques du second ordre. L'algorithme est capable de traiter le cas de surfaces bornées ou non bornées. Son efficacité est illustrée sur deux surfaces de rupture obtenues par des procédures de type changement d'échelle.

Pour citer cet article : J. Bleyer, P. de Buhan, C. R. Mecanique 341 (2013), pp. 605-615

Email addresses: `jeremy.bleyer@enpc.fr` (Jérémy Bleyer), `patrick.debuhan@enpc.fr` (Patrick de Buhan)

Version française abrégée

1. Introduction

Yield design theory has proved to be a very efficient tool for assessing the yield strength of different types of structures without performing cumbersome elastoplastic analyses. It relies, indeed, on checking the compatibility between equilibrium equations and satisfaction of the local strength criterion. As regards heterogeneous periodic media, yield design-based homogenization procedures have been proposed by Suquet [1] or de Buhan [2] in the context of reinforced soil mechanics. The solution of an auxiliary yield design problem formulated on the unit periodic cell leads to the construction of a macroscopic strength criterion.

In the case of elastic homogenization procedures, the homogenized quantity to be determined is the macroscopic stiffness tensor which can entirely be described by the value of its components. On the contrary, the homogenized quantity of interest in yield design is a macroscopic yield surface. Such yield surfaces can be very complex (due to the heterogeneous yield criterion, strength anisotropy...) and, therefore, can no longer be easily described by analytic formulas. For this simple reason, computation of limit loads on homogeneous equivalent structures has been limited to simple cases where analytic formulas were available.

Fortunately, the development of efficient algorithms for optimization problems now makes it possible to tackle problems involving a quite large number of variables [3]. Being able to use complex macroscopic yield surfaces associated to efficient optimization solvers would then be of paramount interest as regards engineering applications.

The aim of this Note is to propose a simple formulation and related computational tools describing such complex numerically determined yield surfaces. In section 2, the proposed formulation using a convex hull of ellipsoids is presented and justified in the scope of mathematical programming techniques currently at hand. Section 3 is devoted to the description of a constructive algorithm to approximate a general yield surface (determined for example by a homogenization or *up-scaling* procedure) with the previous formulation. Finally, two examples illustrating the algorithm efficiency will be presented in section 4.

2. Yield surface approximation

The main difficulty arising in the numerical formulation of yield design problems lies in the verification of the yield criterion at each point of the structure. When considering the upper bound kinematic approach, it corresponds to the implementation of the (usually non-linear) support function of the yield criterion in the global optimization problem.

As a consequence, simplified descriptions of the yield criterion (or, equivalently, approximations of the support function) are often required to implement the problem in an appropriate numerical solver.

2.1. Numerical challenges

Piecewise linearization (PWL) of yield criteria has often been proposed in the literature to overcome this difficulty [4, 5, 6]. In this case, the support function can be expressed using N_v linear inequalities, where N_v corresponds to the number of vertices of the approximating polytope. Replacing the original yield criterion by a piecewise linear approximation, the corresponding optimization problem reduces to a linear programming (LP) problem. This formulation has been highly attractive due to the performance of LP solvers using interior point algorithms. However, PWL approximations of yield surfaces can require an important number of vertices to obtain a good accuracy [7]. Since the support function has to be evaluated at each Gauss points of the structure, the total number of inequalities can become very important for complex structures.

Interior points algorithm have been extended to the case of second-order cone programming (SOCP) problems which include non-linear conic constraints. Remarkably, SOCP encompasses a large class of convex optimization problems (including LP) and most usual strength criteria can be formulated using SOCP constraints [8]. Thus, yield design problems in plane strain, plane stress [9, 10] or for plates in bending [11, 12] have been successfully solved using SOCP formulations without any linearization procedure. Therefore, two important advantages can be foreseen from approximating yield surfaces using conic constraints : first, it is expected that *conic* approximation would require less constraints than *linear* approximation allowing to tackle more complex problems; secondly, highly efficient SOCP solvers can be used to treat the global optimization problem.

The following subsection proposes a formulation using a convex hull of ellipsoids, which can be treated using conic constraints.

2.2. An interesting formulation using a convex hull of ellipsoids

Let $\mathcal{E} \subset \mathbb{R}^3$ be an ellipsoid centered at point c, the principal axes of which are given by u, v and w forming an orthogonal basis. Its support function is then given by :

$$\pi_{\mathcal{E}}(d) = \sup_{\sigma \in \mathcal{E}} \sigma \cdot d = \sqrt{(u \cdot d)^2 + (v \cdot d)^2 + (w \cdot d)^2} + c \cdot d = \|A \cdot d\| + c \cdot d$$

with $A = {}^T[u\,v\,w]$. Moreover, if $\mathcal{G}$ is a convex hull of n ellipsoids $\mathcal{E}_i$ with matrix A_i and centers c_i, then its support function is given by :

$$\pi_{\mathcal{G}}(d) = \max_{i=1,\dots,n} (\pi_{\mathcal{E}_i}(d)) = \max_{i=1,\dots,n} (\|A_i \cdot d\| + c_i \cdot d)$$

This formulation is quite interesting in the scope of numerical yield design, since it can be expressed in terms of conic constraints by introducing $n+1$ auxiliary scalar variables $(t_0, t_1, \dots, t_n)$ such that :

$$\begin{aligned} t_i + c_i \cdot d &\leq t_0 \\ \|A_i \cdot d\| &\leq t_i \end{aligned} \qquad \forall i = 1, \dots, n$$

3

This new formulation involves n linear inequalities and n second-order cone inequalities such that, when minimizing t_0 under the previous constraints, we have $t_0 = \pi_\mathcal{G}(d)$ at the optimum.

As a consequence, if $\mathcal{G}$ can be expressed as a convex hull of ellipsoids, the corresponding minimization problem of the kinematic approach can be written as a second-order cone programming problem and efficiently solved using dedicated solvers. This formulation deserves two comments :
- the cost to pay is obviously related to the number of additional optimization variables t_i and to the linear or conic inequality constraints;
- the same kind of results can be established when considering the lower bound static approach of yield design. Therefore, with such a formulation, both approaches can be efficiently tackled using SOCP solvers.

3. An algorithm to approximate a yield surface by a convex hull of ellipsoids

This section is aimed at deriving a relatively simple algorithm which computes a series of optimal ellipsoids such that their convex hull is an approximation of the initial yield surface. This algorithm uses the dual description of the convex yield surface characterized by its support function π. It is therefore particularly well suited to macroscopic yield surfaces obtained from a homogenization upper bound approach.

3.1. General assumptions and notations

The considered yield surfaces are supposed to be a convex bounded set $\mathcal{G} \subset \mathbb{R}^3$ containing the origin. It is to be noted that this algorithm can be extended without difficulty to the case of unbounded convex sets, as discussed in the second illustrative example of section 4.2.
The intuitive algorithm constructs an approximation from the inside of the original yield surface. However, provided that this approximation is sufficiently accurate, it is always possible to expand the inner approximation by an appropriate factor to obtain an outer approximation. Both approximations can be valuable regarding their use in a yield design problem at the structure scale, considering either the lower bound static approach or the upper bound kinematic approach.
In the following, it will be assumed that M values of the support function π_j at directions [1] d_j on the unit sphere $\mathbb{S}$ are known. The directions d_j are supposed to be uniformly distributed on the unit sphere. We will note D the $M \times 3$ matrix of the directions d_j and π the $M \times 1$ vector made by the corresponding values of the support function.
A two-dimensional example will serve to illustrate the principle of the algorithm

---

1. We will use the term *directions* for the values at which support functions are evaluated. These values correspond, indeed, to normal vectors of supporting hyperplanes.
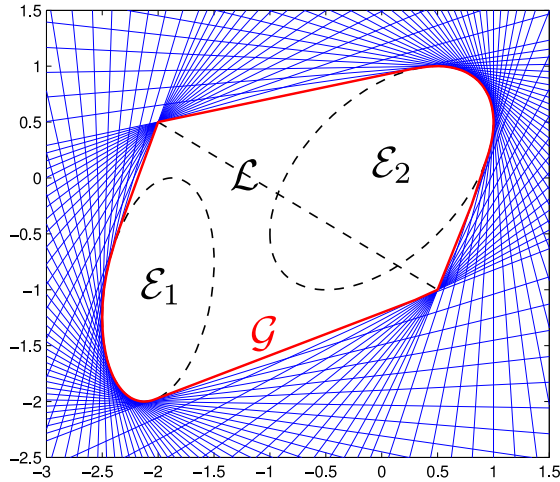
Figure 1: Illustrative 2D example. The considered convex set $\mathcal{G}$ (in red) is the convex hull of two ellipses ($\mathcal{E}_1$ and $\mathcal{E}_2$) and a segment $\mathcal{L}$. The supporting planes for different directions d_j are represented in blue.

throughout the discussion. The considered convex set $\mathcal{G}$ is obtained as the convex hull of two ellipses ($\mathcal{E}_1$ and $\mathcal{E}_2$) and a segment $\mathcal{L}$ (figure 1). Its support function has the following simple analytical expression :

$$\pi_{\mathcal{G}}(d_x, d_y) = \max\left\{\pi_{\mathcal{E}_1}(d_x, d_y), \pi_{\mathcal{E}_2}(d_x, d_y), \pi_{\mathcal{L}}(d_x, d_y)\right\}$$

where the support functions of the ellipses and the segment are given by :

$$\pi_{\mathcal{E}_1}(d_x, d_y) = \tfrac{1}{2}\sqrt{d_x^2 + 4d_y^2 + d_x d_y} - 2d_x - d_y$$
$$\pi_{\mathcal{E}_2}(d_x, d_y) = \sqrt{d_x^2 + d_y^2 + d_x d_y}$$
$$\pi_{\mathcal{L}}(d_x, d_y) = \max\{-2d_x + \tfrac{1}{2}d_y, \tfrac{1}{2}d_x - d_y\}$$

3.2. Outline of the algorithm

At each step, the proposed iterative algorithm constructs an ellipsoid approximating $\mathcal{G}$ in a given region, which is then added to the convex hull (CH) approximation of the previous step. Supposing that at step n, an approximation $\mathcal{P}_n = CH(\mathcal{E}_i)$ with ellipsoids $\mathcal{E}_i$ for $1 \leq i \leq n$ is available, the support function Π_n of $\mathcal{P}_n$ has a simple analytical expression in function of the geometric parameters of the different $\mathcal{E}_i$. Therefore, it is possible to compute the difference between the original support function and the current approximation $\pi_j - \Pi_n(d_j)$ for all j in order to find the direction d_0 for which the difference is the largest.

Given this direction d_0, the decisive step of the algorithm will be to find a new

ellipsoid $\mathcal{E}_{n+1}$ approximating $\mathcal{G}$ a neighborhood[2] of d_0. Once $\mathcal{E}_{n+1}$ is found, it is added to the convex hull to form the new approximation at step $n + 1$: $\mathcal{P}_{n+1} = CH(\mathcal{P}_n; \mathcal{E}_{n+1})$.

3.3. Using the support function to characterize the local geometry

The key ingredient of the algorithm is based on a characterization of the local geometry of $\mathcal{G}$ in the neighborhood of d_0. For this purpose, the support function provides very useful information on the local geometry. All properties on the curvature can, indeed, be obtained from the Weingarten map W (curvature tensor), the eigenvectors and eigenvalues of which are the principal directions of curvatures and principal curvatures of $\mathcal{G}$ respectively. Moreover, the following relation between the support function $\pi(d)$ and the Weingarten map holds true [13, 14] :

$$W = -\left(\text{Hess}_{\mathbb{S}}\,\pi(d) + \pi(d)\text{Id}\right)^{-1}$$

with $\text{Hess}_{\mathbb{S}}\,\pi(d)$ being the Hessian of $\pi(d)$ with respect to the unit sphere $\mathbb{S}$ if $\pi(d)$ is of class C^2. Therefore, if λ is an eigenvalue of $\text{Hess}_{\mathbb{S}}\,\pi(d)$, then its associated eigenvector is a principal direction of curvature with a radius of curvature given by $\rho = |\lambda + \pi(d)|$.

Thus, provided that the Hessian can be computed, all information concerning the local curvature are available. A natural idea is then to fit an ellipsoid with the same curvature to obtain a local approximation of $\mathcal{G}$ with an higher order of accuracy than just a point.

3.4. Computation of the best ellipsoid with prescribed curvature

Let σ_0 be a tangent point between $\mathcal{G} \in \mathbb{R}^3$ and the supporting plane with normal d_0. Denoting by d_1 and d_2 the principal directions of curvature, we will look for an ellipsoid of principal axes $\{d_0, d_1, d_2\}$ contained in $\mathcal{G}$, tangent to $\mathcal{G}$ at σ_0 and such that its radii of curvature at σ_0 are given by $\rho_1 = |\lambda_1 + \pi(d_0)|$ and $\rho_2 = |\lambda_2 + \pi(d_0)|$.

Besides, if the principal axes semi-lengths of the ellipsoid are denoted by a_0, a_1 and a_2 and choosing σ_0 as the point of coordinates $(a_0, 0, 0)$ in the local frame of the ellipsoid with the origin at its center, the radii of curvature of the ellipsoid at σ_0 are given by the following relations :

$$\rho_1 = \frac{a_1^2}{a_0} \quad \text{and} \quad \rho_2 = \frac{a_2^2}{a_0}$$

Therefore, ρ_1 and ρ_2 being kept fixed, the only remaining degree of freedom is, for example, a_0 (see figure 2). Admissible values of a_0 are those corresponding to ellipsoids contained in $\mathcal{G}$. For such values, we determine the set of

---

2. Here again, due to the retained dual description via support functions, a neighborhood of d_0 will represent supporting planes whose normal directions are close to d_0 (in the sense of the scalar product for example). Hence, a neighborhood of d_0 does not necessarily correspond to a neighborhood of a given point on $\mathcal{G}$.
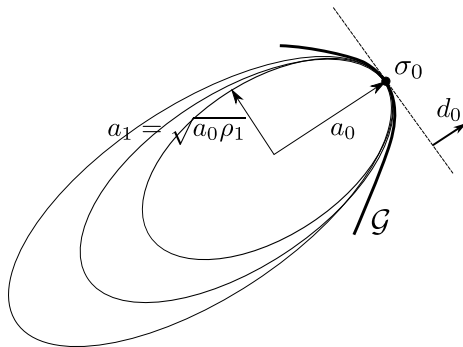
Figure 2: Series of ellipsoids with prescribed curvatures at σ_0

directions where the corresponding ellipsoid improves the previous approximation $\mathcal{P}_n$. Then, we compute the root mean squared (RMS) error between its support function and the original one on this given set. The best ellipsoid is the one for which the RMS error is minimal.

3.5. Implementation details

    – *Initialization :* Initialization can be performed using any point in $\mathcal{G}$: the origin, the *center* of $\mathcal{G}$ defined as the point minimizing the distance between all supporting planes, etc... This choice did not seem to influence the quality of the approximation.

    – *Neighborhood of d_0 :* The determination of the neighborhood of d_0 is performed by sorting the scalar products $d_j \cdot d_0 \; \forall j$ by descending order and taking the first m directions except d_0 itself.

    – *Computation of Hess$_{\mathbb{S}}\,\pi(d_0)$:* We perform a second order Taylor expansion of $\pi(d)$ for all d in the neighborhood of d_0. However, we recall that directions d are required to belong to the unit sphere so that the Taylor expansion has the following expression [15] :

$$\pi(d) \approx \pi(d_0) + \langle \text{grad}_{\mathbb{S}}\,\pi(d_0), \text{Log}_{d_0}(d)\rangle + \frac{1}{2}\,\langle \text{Log}_{d_0}(d), \text{Hess}_{\mathbb{S}}\,\pi(d_0)\text{Log}_{d_0}(d)\rangle$$

where $\langle\cdot,\cdot\rangle$ is the scalar product on the tangent plane to $\mathbb{S}$ at d_0, $\text{grad}_{\mathbb{S}}\pi(d_0)$ is the gradient of $\pi(d)$ with respect to $\mathbb{S}$ and $\text{Log}_{d_0}(d)$ is the logarithmic mapping (see figure 3) from d to the tangent plane such that $\|\text{Log}_{d_0}(d)\| = \text{dist}_{\mathbb{S}}(d, d_0)$ where $\text{dist}_{\mathbb{S}}(d, d_0)$ is the geodesic distance on $\mathbb{S}$. On $\mathbb{S}$, this reduces to [16] :

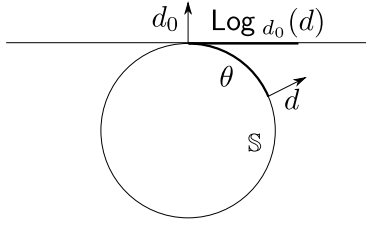$$\text{Log}_{d_0}(d) = \left(d \cdot e_1 \frac{\theta}{\sin\theta}, d \cdot e_2 \frac{\theta}{\sin\theta} \right)$$

7

Figure 3: Logarithmic mapping on $\mathbb{S}$

with $\mathrm{dist}_{\mathbb{S}}(d, d_0) = \theta = \arccos(d \cdot d_0)$ and $\{e_1, e_2\}$ an orthonormal basis of the tangent plane.

Forming vectors $y = (y_1, y_2) = \mathrm{Log}_{d_0}(d)$, $d\pi = \pi(d) - \pi(d_0)$ and matrix $B = [y_1 \quad y_2 \quad y_1^2/2 \quad y_2^2/2 \quad y_1 y_2]$, we solve the following least-square problem :

$$r^* = \arg\min \|Br - d\pi\|$$

This operation can be efficiently performed in MATLAB with the back-slash \ operator. The Hessian is then given by the following matrix $H = \begin{bmatrix} r_3^* & r_5^* \\ r_5^* & r_4^* \end{bmatrix}$.

– *Tangent point σ_0 :* The tangent point σ_0 between $\mathcal{G}$ and the supporting plane with normal d_0 is found by solving the following linear programming optimization problem :

$$\sigma_0 = \arg \max_{x, Dx \leq \pi} x \cdot d_0$$

This operation can be easily performed with MATLAB linear programming solver `linprog` or other software packages like Mosek [17] for example.

– *Optimal value of a_0 :* The retained strategy for finding the optimal value of a_0 consists in estimating the maximal value which can take a_0. For example, we can choose the half of the diameter of $\mathcal{G}$ in the direction d_0 given by :

$$a_{max} = \frac{1}{2}(\pi(d_0) + \pi(-d_0))$$

Then, the segment $[0; a_{max}]$ is discretized in N_d values. N_d different ellipsoids tangent at σ_0 and with prescribed curvature are obtained. The optimal ellipsoid is determined as discussed before. This stage is represented in figure 4.

– *Actualization before next step :* Once the optimal ellipsoid $\mathcal{E}$ has been determined, its geometric parameters (center $c_0 = \sigma_0 - a_0 d_0$ and principal
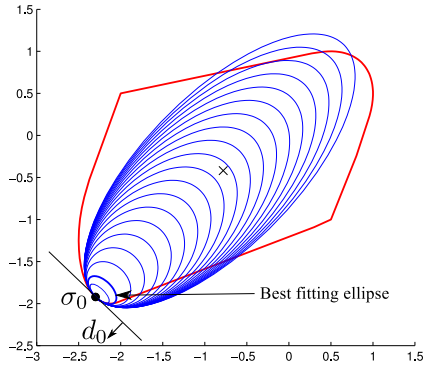
Figure 4: Different ellipses with the same curvatures as $\mathcal{G}$ at σ_0.

axis) are saved and $\mathcal{E}$ is added to the previous approximation for the next step. Hence, the support function of the new approximation will be :

$$\Pi_{n+1}(d) = \max(\Pi_n(d), \pi_{\mathcal{E}}(d))$$

where $\pi_{\mathcal{E}}(d)$ is the support function of the optimal ellipsoid given by :

$$\pi_{\mathcal{E}}(d) = \sqrt{(a_0 d_0 \cdot d)^2 + (a_1 d_1 \cdot d)^2 + (a_2 d_2 \cdot d)^2} + c_0 \cdot d$$

The first 6 steps of the algorithm are illustrated in figure 5. We can observe that, starting with the center of $\mathcal{G}$ (represented by a black cross), the algorithm constructs, step-by-step, optimal ellipsoids in the regions which are the farthest away from the previous approximation.

It can clearly be seen that the local curvature is correctly computed and that in regions with corners, the determination of the Hessian correctly accounts for the fact that the radius of curvature is zero, which results in elongated ellipsoids.

Finally, we can also mention that this algorithm is greedy since, at each stage, it performs optimal operations but cannot find the globally optimal solution, which consists, here, in the two ellipsoids and the segment used to construct $\mathcal{G}$. Nevertheless, this example illustrates its performance, since it can approximate $\mathcal{G}$ with a very good accuracy in 6 steps only.

4. Illustrative examples

In this section, the efficiency of the proposed algorithm is demonstrated on two different numerically computed yield surfaces. In the first example, a bounded yield surface is approximated whereas the second example involves an unbounded yield surface.
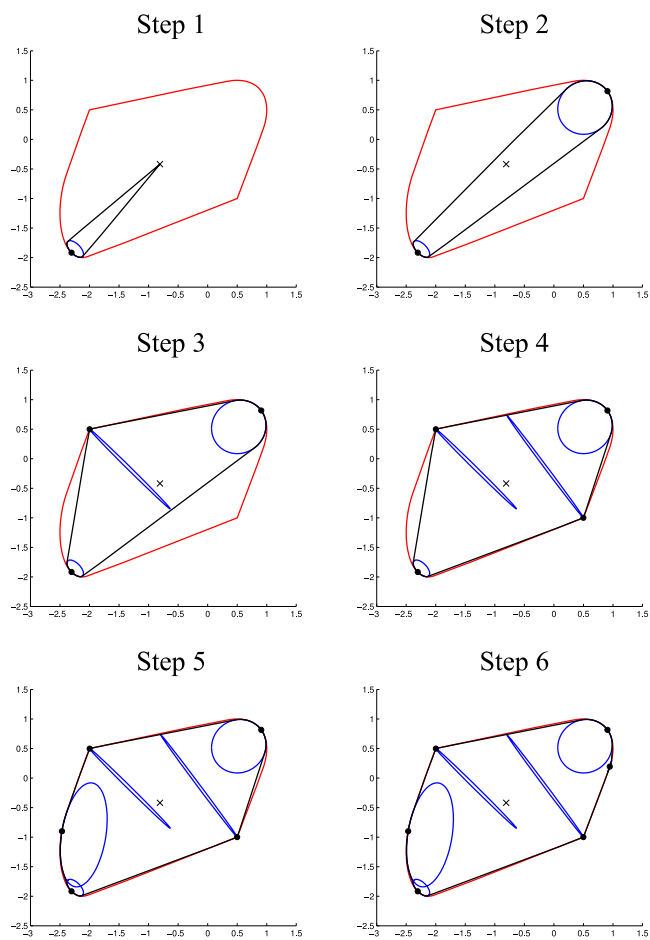
9

Figure 5: First six steps of the algorithm on the illustrative example of figure 1. The black line represents the approximating convex at each step, obtained by taking the convex hull of the computed ellipsoids. Black dots represent the computed tangent points σ_0.

4.1. Interaction surface of a reinforced concrete beam section

The first example considers the interaction surface of a beam section subject to combined bending moments M_y, M_z and axial force N. The considered section is a L-shaped concrete section reinforced by steel bars (figure 6).
The bending moments and axial force in equilibrium with a stress field $\underline{\underline{\sigma}}$, are given by :

$$N = \int_S \sigma_{xx}(y, z)dS$$

$$M_y = -\int_S z\sigma_{xx}(y, z)dS$$

$$M_z = -\int_S y\sigma_{xx}(y, z)dS$$

where S is the transverse section of the beam.
Let S_1 (resp. S_2) denote the region occupied by the concrete (resp. steel reinforcement), σ_c^1 (resp. σ_c^2) the uniaxial compressive strength in the x-direction and σ_t^1 (resp. σ_t^2) the uniaxial tensile strength in the x-direction of the concrete (resp. steel reinforcement). The interaction surface $\mathcal{G}$ is obtained by considering statically admissible uniaxial stress fields of the form : $\underline{\underline{\sigma}} = \sigma(y, z)\underline{e}_x \otimes \underline{e}_x$ with :

$$\sigma(y, z) = \begin{cases} \sigma_t^i & \text{if } \delta - z\chi_y - y\chi_z > 0 \text{ and } (y, z) \in S_i \\ -\sigma_c^i & \text{if } \delta - z\chi_y - y\chi_z < 0 \text{ and } (y, z) \in S_i \end{cases} \quad \forall i = 1, 2$$

where χ_y, χ_z and δ are the generalized strains (curvatures and axial strain) associated with M_y, M_z and N. Hence, the support function of $\mathcal{G}$ is given by :

$$\pi(\chi_y, \chi_z, \delta) = \sum_{i=1,2} \int_{S_i} \sup \left\{ \sigma_t^i \left(\delta - z\chi_y - y\chi_z \right) ; -\sigma_c^i (\delta - z\chi_y - y\chi_z) \right\} dS$$

The corresponding interaction surface is represented on figure 7(a) in the $(m_y, m_z, n)-$space with $m_y = M_y/M_{y0}$, $m_z = M_z/M_{z0}$ and $n = N/N_0$ where $\Sigma_0 = (\max \Sigma - \min \Sigma)/2$ for $\Sigma = M_y, M_z$ or N. Numerical values of strength were $\sigma_c = 30$ MPa and $\sigma_t = 1.8$ MPa for concrete and $\sigma_c = \sigma_t = 435$ MPa for steel reinforcement.

The above-described algorithm has been tested on this surface by taking $m = 10$ for the number of directions in a neighborhood and $N_d = 200$ for the number of discretization values for a_0. The surface has been approximated using up to 50 ellipsoids. The surface obtained with $N = 30$ ellipsoids is represented on figure 7(b). One can clearly see that the quality of reconstruction obtained with 30 ellipsoids is very good. More precisely, the evolution of the relative error with respect to the original support function is represented in figure 8(a) as a function of the number of ellipsoids. Two different error norms were plotted : the maximal error and the root mean squared (RMS) error. It can clearly be seen that the error is rapidly decreasing with the number of ellipsoids used in the approximation. For instance, with $N = 30$ ellipsoids the maximum

relative error made on the support function is less than 4% whereas the RMS error is around 0.5%. With $N = 50$ ellipsoids, the maximal error is less than 2%.

In order to compare the speed-up gained with this approximate description of the initial convex set, an artificial optimization problem, characteristic of those arising in numerical yield design computations, has been imagined, involving either the initial convex set of figure 7(a) consisting of approximately $N_p = 5100$ points a_j or the approximate set of N ellipsoids. Fixing a value N_S characterizing the size of the problem and random vector F of size $3N_S$, the optimization problem can be written as :

$$\min_{D \in \mathbb{R}^{3N_S}} \quad \sum_{i=1}^{N_S} \pi(D_{3i-2 \to 3i})$$
$$\text{s.t.} \quad F \cdot D = 1$$

where $\pi(d)$ represents the support function of the considered convex set. In the case of the initial convex set, the optimization problem reads as the following linear programming problem :

$$\min_{D \in \mathbb{R}^{3N_S}} \quad \sum_{i=1}^{N_S} t_i$$
$$\text{s.t.} \quad F \cdot D = 1$$
$$a_j \cdot D_{3i-2 \to 3i} \leq t_i \quad \forall j = 1, \ldots, N_p$$

whereas, with the previous notations, the problem with the approximate set of N ellipsoids will read as the following SOCP problem :

$$\min_{D \in \mathbb{R}^{3N_S}} \quad \sum_{i=1}^{N_S} t_{0,i}$$
$$\text{s.t.} \quad F \cdot D = 1$$
$$\begin{aligned} \|A_j \cdot D_{3i-2 \to 3i}\| &\leq t_{j,i} \\ t_{j,i} + c_j \cdot D_{3i-2 \to 3i} &\leq t_{0,i} \end{aligned} \quad \forall j = 1, \ldots, N$$

Considering different values of N_S, both problems are solved with the Mosek software package and the corresponding interior-point computation times are reported in table 1. It can be observed that the approximate problems with $N = 30$ or $N = 50$ ellipsoids are solved approximately three times faster than the initial problem without any approximation procedure. Besides, it was impossible to solve the largest considered optimization problem ($N_S = 500$) with the initial convex set because too much memory space was required for the problem to be formulated. Therefore, the proposed approximation procedure is advantageous in terms of computational time saving as well as memory requirement.

Besides, the algorithm can also be degenerated to compute an approximation of the yield surface using points [3] only. A convergence analysis between

---
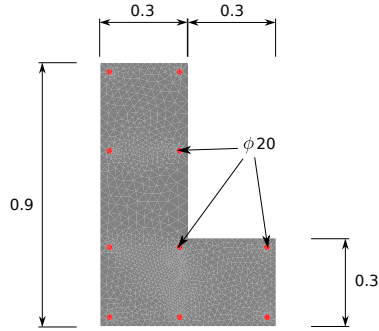
3. The points correspond to σ_0 at each step.

Figure 6: Geometry in the $(y, z)-$plane of the RC section : concrete in gray, steel reinforcements in red (units : meters, millimeters for reinforcement diameters)
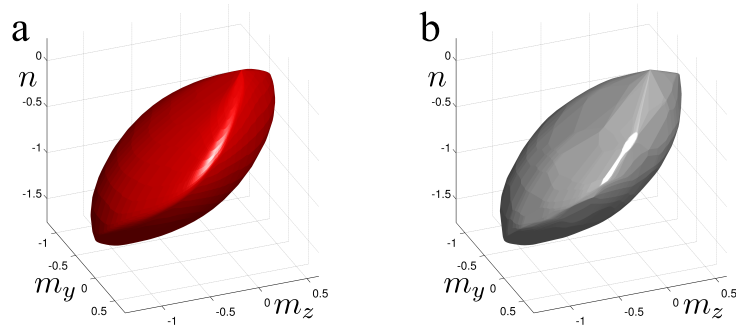

Figure 7: Reconstruction of the interaction surface for the reinforced concrete L section of figure 6 : original surface in red, approximated surface with $N = 30$ ellipsoids in gray

ellipsoid-based and point-based approximations was performed on this example. Evolution of relative error norms are plotted in figure 8(b) with respect to the number of primitives (ellipsoids or points). As expected, the approximation using ellipsoids presents higher convergence rates (almost 1.5 higher for both error norms) than approximation using points only, thereby illustrating the efficiency of this method.

Finally, it is to be noted that, in this particular example, both primal (in generalized stress space) and dual (via support function) descriptions of $\mathcal{G}$ were available. Nevertheless, in both descriptions, it was impossible to derive a simple analytical formula describing $\mathcal{G}$, to be used in an optimization tool. Thus, an approximation procedure is required to use this interaction surface, in order to compute the limit load of a structure made of such sections.

13
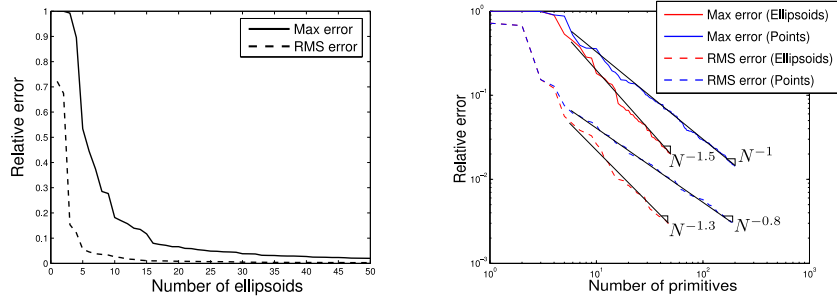
(a) Evolution of RMS and max relative error (b) Comparison of convergence rates be-
with the number of ellipsoids   tween ellipsoid-based and point-based ap-
                                                  proximations

Figure 8: Convergence analysis of the approximation procedure on the reinforced concrete L
section interaction surface

Problem size	Initial	Approximate $N = 30$	Approximate $N = 50$
$N_S = 50$	3.93	0.64	1.07
$N_S = 100$	7.83	1.85	2.23
$N_S = 200$	15.76	5.03	6.16
$N_S = 500$	*	12.4	33.6

Table 1: Computing time (in seconds) for the interior-point solver on a Intel-P4 2.4 GHz with
Mosek v6.0. The symbol "*" signifies that Mosek was unable to solve the problem due to
insufficient available memory.

The second example involves an unbounded yield surface corresponding to the macroscopic strength properties of a purely cohesive soil reinforced by a periodic distribution of stone columns made of a highly frictional granular material. This macroscopic yield surface was determined in [18] on the basis of a yield design homogenization approach using a series of numerical elastoplastic simulations performed on a unit cell of the reinforced soil.
As an example, the determination of such a complex criterion is the first step to the evaluation of the bearing capacity of stone columns reinforced foundations. However, due to the criterion complexity, an approximation procedure is necessary to implement a numerical formulation of the corresponding yield design problem on the equivalent homogeneous reinforced soil.

As already mentioned, the proposed algorithm can easily be extended to the case of unbounded yield surfaces. Indeed, instead of considering directions of the supporting plane normals describing the whole unit sphere $\mathbb{S}$, one has only to consider the subset of directions for which the yield surface is bounded. The only other modification relies on the evaluation of a_{max} since the diameter in the direction d_0 can be infinite for some d_0. For example, in the case when the yield surface is bounded in direction d_0 but not in $-d_0$, it was decided to replace $-d_0$ in the definition of a_{max} by $-\widehat{d_0}$ which is the closest direction to $-d_0$ for which the yield surface is bounded.

Using the same optimization parameters as before, the algorithm was tested on the yield surface numerically determined in [18]. The original yield surface as well as the obtained approximation using $N = 30$ ellipsoids were represented in the space of "plane-strain" macroscopic stresses Σ_{xx}, Σ_{yy} and $2\Sigma_{xy}$ (figure 9). Evolution of the error norms was also plotted in figure 10. One can observe that the maximal error is around 4% with $N = 30$ ellipsoids whereas the RMS error is around 1%.

5. Conclusions and future work

An efficient algorithm has been proposed to approximate general yield surfaces using a convex hull of ellipsoids. This algorithm uses the dual description of a convex yield surface by its support function, to compute the local curvature of the surface and to find an optimal ellipsoid. The main characteristics of this algorithm are the following.
- Bounded and unbounded surfaces can be approximated by controlling the number of ellipsoids.
- the resolution of a linear programming problem is the only stage requiring a specific solver. In particular, this algorithm does not require any nonlinear solver for which convergence issues may appear.
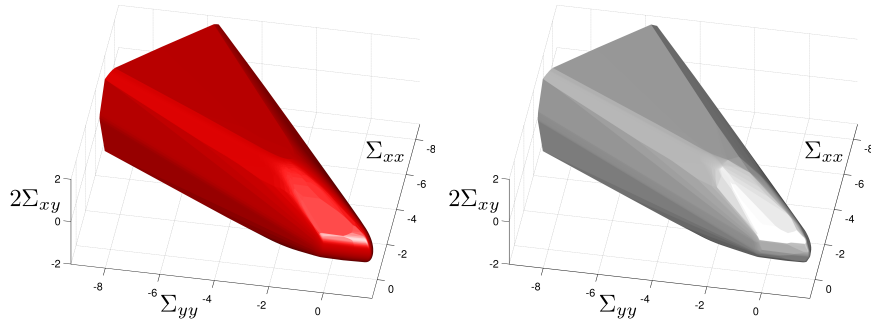- Approximations from the inside are obtained.

Figure 9: Reconstruction of the macroscopic unbounded yield surface for a stone column reinforced soil (computed from [18]) : original surface in red, approximated surface with $N = 30$ ellipsoids in gray
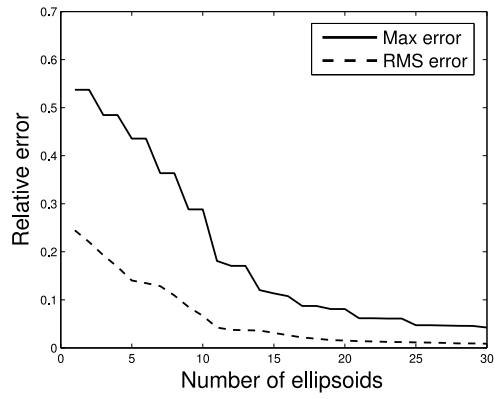

Figure 10: Evolution of RMS and max relative error with the number of ellipsoids for the macroscopic yield surface

16

– The support function of the obtained approximation can be analytically computed from the geometrical parameters determined by the algorithm. Moreover, it can be easily formulated in terms of second-order cone constraints, which allows to formulate corresponding yield design problems as SOCP, which have recently proved to be very efficient to solve these problems.

It was also illustrated on one example that the convergence rate of the error was sensibly higher when using a convex hull of ellipsoids rather than a convex hull of points to approximate the yield surface. This is highly valuable if such yield surfaces are used on yield design problems at the structure level since such formulations will require less constraints compared to a piecewise linearization procedure.
Concerning the presented algorithm, further work could be conducted to investigate the influence of the number of directions in the neighborhood or to imagine more sophisticated strategies to improve its efficiency. Nevertheless, it has been demonstrated that its performances are highly satisfactory. Moreover, since our primary interest lies in the computation of limit loads on heterogeneous structures, our future work will therefore be aimed at using this algorithm to formulate yield design problems on complex structures made of heterogeneous materials, using SOCP.

References

[1] P. Suquet, Elements of homogenization for inelastic solid mechanics, Homogenization techniques for composite media 272 (1985) 193–278.

[2] P. de Buhan, A fundamental approach to the yield design of reinforced soil structures, Ph.D. thesis, Thèse d'Etat, Paris VI (1986).

[3] E. Christiansen, K. Andersen, Computation of collapse states with von mises type yield condition, International Journal for Numerical Methods in Engineering 46 (8) (1999) 1185–1202.

[4] E. Anderheggen, H. Knöpfel, Finite element limit analysis using linear programming, International Journal of Solids and Structures 8 (12) (1972) 1413–1431.

[5] S. Sloan, Lower bound limit analysis using finite elements and linear programming, International Journal for Numerical and Analytical Methods in Geomechanics 12 (1) (1988) 61–77.

[6] S. Sloan, Upper bound limit analysis using finite elements and linear programming, International Journal for Numerical and Analytical Methods in Geomechanics 13 (3) (1989) 263–282.

[7] A. Makrodimopoulos, C. Martin, Upper bound limit analysis using simplex strain elements and second-order cone programming, International journal

for numerical and analytical methods in geomechanics 31 (6) (2007) 835–865.

[8] A. Makrodimopoulos, Remarks on some properties of conic yield restrictions in limit analysis, International Journal for Numerical Methods in Biomedical Engineering 26 (11) (2010) 1449–1461.

[9] A. Makrodimopoulos, C. Martin, Lower bound limit analysis of cohesive-frictional materials using second-order cone programming, International Journal for Numerical Methods in Engineering 66 (4) (2006) 604–634.

[10] H. Ciria, J. Peraire, J. Bonet, Mesh adaptive computation of upper and lower bounds in limit analysis, International journal for numerical methods in engineering 75 (8) (2008) 899–944.

[11] C. Le, H. Nguyen-Xuan, H. Nguyen-Dang, Upper and lower bound limit analysis of plates using fem and second-order cone programming, Computers & Structures 88 (1-2) (2010) 65–73.

[12] J. Bleyer, P. de Buhan, On the performance of non-conforming finite elements for the upper bound limit analysis of plates, International Journal for Numerical Methods in Engineering 94 (3) (2013) 308–330. doi:10.1002/nme.4460.
URL http://dx.doi.org/10.1002/nme.4460

[13] J. Gravesen, Surfaces parametrized by the normals, Computing 79 (2) (2007) 175–183.

[14] Z. Sir, J. Gravesen, B. Juttler, Curves and surfaces represented by polynomial support functions, Theoretical Computer Science 392 (13) (2008) 141 – 157.

[15] P. Absil, R. Mahony, R. Sepulchre, Optimization algorithms on matrix manifolds, Princeton University Press, 2009.

[16] S. Sen, Classification on manifolds, Ph.D. thesis, University of North Carolina (2008).

[17] Mosek, The mosek optimization toolbox for matlab manual (2008).
URL http://www.mosek.com/

[18] G. Hassen, M. Gueguin, P. de Buhan, A homogenization approach for assessing the yield strength properties of stone column reinforced soils, European Journal of Mechanics - A/Solids 37 (0) (2013) 266 – 280. doi:10.1016/j.euromechsol.2012.07.003.