A linear programming-based algorithm for the signed separation of (non-smooth) convex bodies

G. Johnson a,⇑, M. Ortiz a, S. Leyendecker b

⇑ Corresponding author. Address: 1200 E. California Blvd., MC 105-50, Pasadena, CA 91125, USA.
E-mail address: gwen@caltech.edu (G. Johnson).

Abstract

A subdifferentiable global contact detection algorithm, the Supporting Separating Hyperplane (SSH) algorithm, based on the signed distance between supporting hyperplanes of two convex sets is developed. It is shown that for polyhedral sets, the SSH algorithm may be evaluated as a linear program, and that this linear program is always feasible and always subdifferentiable with respect to the configuration variables, which define the constraint matrix. This is true regardless of whether the program is primal degenerate, dual degenerate, or both. The subgradient of the SSH linear program always lies in the normal cone of the closest admissible configuration to an inadmissible contact configuration. In particular if a contact surface exists, the subgradient of the SSH linear program is orthogonal to the contact surface, as required of contact reactions. This property of the algorithm is particularly important in modeling stiff systems, rigid bodies, and tightly packed or jammed systems.

1. Introduction

The objective of this paper is to develop a contact detection algorithm and contact potential for non-smooth convex bodies. The proposed contact detection algorithm can be concisely described as a supporting separating hyperplane (SSH) test for interpenetration, and is based on standard separation theorems for compact convex sets. We develop this test in detail for polyhedral sets, where the SSH test can be effectively reformulated as a linear programming problem—the SSH LP. We further show that the subgradient of the SSH LP can be readily evaluated and that it supplies the force system at the time of contact.

1.1. Previous work

A large body of literature exists on the efficient detection of collisions, driven in large part by advances in computational geometry, computer graphics and robotics [1–13]. While several advances have taken place in the past several years, the 2001 review by Jimenez and others [14] and the 2005 book by Ericson [11] effectively summarize the essential state of the art. We follow the same track as much of this literature, which develops collision detection algorithms for use with convex polyhedra and for solid models that have been discretized by polyhedra. However, we note that the collision detection algorithm presented here is general enough to extend to an interpenetration test between any two smooth or non-smooth convex bodies, but the linear programming solution methodology is not, in general, extensible to these situations.

The most popular software packages make use of hierarchical volume bounding to organize oriented bounding bodies (OBB’s) and axis-aligned bounding bodies (AABB’s) into rapidly searchable data structures (Octrees, K-D trees etc.) [6,7,11]. Intended for discretized surfaces, these algorithms can quickly compute candidate areas for contact, and refine those areas to determine which simplices are actually involved in a collision. However, when the objects get close enough for the bounding volumes to suggest that contact might have taken place, a finer detection test must be used to conclusively declare that a collision has taken place. Another option that works on the coarse and fine levels, proposed by Chung and Wang, detects collision based on the existence (or not) of a separating vector [9]. While our interpenetration function is universal and robust, in that it can be evaluated for any pair of convex bodies, and is certainly capable of serving as a collision detection test, we would rather suggest its use as a final test in conjunction with one of the coarser tests referenced in this paragraph.

The interpenetration function here can be seen as an alternative to the heuristic search for a separating vector developed by Chung and Wang [9], and also as an alternative to other linear programming approaches such as those proposed by Akgunduz and others [10] and Aliyu and Al-Sultan [8], to which Seidel [15] made key contributions. The advantage of our proposed linear programming
approach to collision detection is that it provides extremely useful additional information for physics-based dynamics simulations and closest-point projection (CPP) operations. In the present work, we follow a well known path to collision detection and undertake a search for a separating vector. However, we seek for this vector in an optimal way so that it always lies in the normal cone of the closest admissible configuration to an inadmissible contact configuration and respects any symmetries present in the geometry of the contact configuration. Furthermore, unlike previously proposed methods, our approach ensures that the subgradient of the SSH linear program is only non-zero with respect to degrees of freedom directly involved in the collision and also respects the geometry of the contact configuration. For example, if a contact surface exists, the subgradient is orthogonal to the contact surface.

1.2. Motivation

The aforementioned shape useful additional information is the key motivation for this work. By way of illustration, we may consider a widely accepted treatment of contact in the equations of motion: the introduction of a contact potential into the action functional reads

\[ \text{I} \quad \text{(x)} = \int_{0}^{T} \mathcal{L}(x, \dot{x}) dt, \]

for the Lagrangian

\[ \mathcal{L}(x, \dot{x}) = \dot{x}^T M \dot{x} - I_d(x), \]

where \( M \) is an appropriate mass matrix and

\[ I_d(x) = \left\{ \begin{array}{ll} 0 & \text{if } x \in \mathcal{A} \\ \infty & \text{otherwise.} \end{array} \right. \]

The equations of motion can be recovered by requiring stationarity of \( \mathcal{I} \)

\[ M \dot{x} + \partial I_d(x) \geq 0. \]

In (4), \( \partial I_d(x) \) denotes the generalized differential of the indicator function (c.f. [16,17]). It is readily shown that the generalized differential of the indicator function of a set is given by the normal cone, \( N(x) \), of the set

\[ \partial I_d(x) = N(x). \]

It follows from (4) that the contact forces \( f_{\text{con}} \) are related to the normal cone by

\[ f_{\text{con}} \in -N(x). \]

The normal cone is defined precisely in Section 2 of this paper. For our introductory example it is sufficient to understand that (6) is a statement that the contact forces must be orthogonal to a contact surface in an admissible configuration. Alternatively, we may consider the contact time as an additional variable, leading to the action functional [19,20]

\[ \text{I}(x, t_c) = \int_{0}^{t_c} \mathcal{L}(x, \dot{x}) dt + \int_{t_c}^{T} \mathcal{L}(x, \dot{x}) dt. \]

Where \( \mathcal{L}(x, \dot{x}) \) is the same as the expression in (2). In this case, the equations of motion at the time of contact read as jump conditions on the change of momentum \( p = M \dot{x} \) and kinetic energy during the collision

\[ p^T M^{-1} p|_{t_c} = 0 \]

\[ p|_{t_c} \in N_x(x(t_c)). \]

Eqs. (8a) and (8b) describe the conservation of energy and momentum during the collision, respectively. In practice, the restriction that the forces from (6) and the change in momentum in (8a) be in the normal cone of the admissible set are accomplished by constraining the configuration variables to be in \( A \subset Q \) via an interpretation function \( g(x) \) that is negative if two bodies are not overlapping and positive if they are, that is \( A = \{ x \in Q | g(x) \leq 0 \} \).

For example, consider a point mass in two dimensions falling onto a flat surface coincident with the \( x_1 \)-axis (see Fig. 1). In this case, the admissible set of configurations for the mass are described by \( A = \{ x \in \mathbb{R}^2 | (-n, x) \leq 0 \} \) with \( n = (0, 1)^T \), and the normal cone of the admissible set has the unique values \( N_x(x) = -n \) if \( x_2 > 0 \) and \( N_x(x) = 0 \) if \( x_2 > 0 \) and \( N_x(x) = 0 \) otherwise.

In this case, (8a) can be expressed as [20]

\[ p^T A p|_{t_c} = \lambda \nabla g, \]

which for our example is equal to

\[ p|_{t_c} \in -\lambda n. \]

if \( x_2 = 0 \). Here, \( \lambda \in \mathbb{R} \) is a scalar parameter (see [20] for details). The simplest re-expression of (6) is accomplished through a smooth approximation of the indicator function

\[ I_d(x) \approx V_d(x) = \begin{cases} 0 & \text{if } g(x) < 0 \\ \frac{1}{2} g(x)^2 & \text{otherwise.} \end{cases} \]

This simple example lends itself to a straightforward geometric interpretation of \( N_x = \partial I_d \), in that the contact forces must be normal to the contact surface (see (12)) and that the change of momentum must also be normal to the surface (see (9)). Thus, to accurately conserve momentum and approximate the continuous equations of motion, a constraint function \( g(x) \) should have the property that \( \nabla g(x) \approx N_x(x) \), as in the point-mass example.

Some time integration schemes based on Lagrangian mechanics use a contact potential in the action functional and require a continuous interpenetration function that is at least sub-differentiable [19–23]. Such a potential is easy to construct for models of geometrically simple bodies and admissible configurations, but a good choice for this potential is much less obvious for complex geometries.

The preferred function for use in these applications has been a test for overlapping oriented simplices (OOS); i.e. tetrahedra in

\[ p^T A p|_{t_c} = \lambda \nabla g, \]

which for our example is equal to

\[ p|_{t_c} \in -\lambda n. \]

if \( x_2 = 0 \). Here, \( \lambda \in \mathbb{R} \) is a scalar parameter (see [20] for details). The simplest re-expression of (6) is accomplished through a smooth approximation of the indicator function

\[ I_d(x) \approx V_d(x) = \begin{cases} 0 & \text{if } g(x) < 0 \\ \frac{1}{2} g(x)^2 & \text{otherwise.} \end{cases} \]

This simple example lends itself to a straightforward geometric interpretation of \( N_x = \partial I_d \), in that the contact forces must be normal to the contact surface (see (12)) and that the change of momentum must also be normal to the surface (see (9)). Thus, to accurately conserve momentum and approximate the continuous equations of motion, a constraint function \( g(x) \) should have the property that \( \nabla g(x) \approx N_x(x) \), as in the point-mass example.

Some time integration schemes based on Lagrangian mechanics use a contact potential in the action functional and require a continuous interpenetration function that is at least sub-differentiable [19–23]. Such a potential is easy to construct for models of geometrically simple bodies and admissible configurations, but a good choice for this potential is much less obvious for complex geometries.

The preferred function for use in these applications has been a test for overlapping oriented simplices (OOS); i.e. tetrahedra in

\[ p^T A p|_{t_c} = \lambda \nabla g, \]

which for our example is equal to

\[ p|_{t_c} \in -\lambda n. \]

if \( x_2 = 0 \). Here, \( \lambda \in \mathbb{R} \) is a scalar parameter (see [20] for details). The simplest re-expression of (6) is accomplished through a smooth approximation of the indicator function

\[ I_d(x) \approx V_d(x) = \begin{cases} 0 & \text{if } g(x) < 0 \\ \frac{1}{2} g(x)^2 & \text{otherwise.} \end{cases} \]

This simple example lends itself to a straightforward geometric interpretation of \( N_x = \partial I_d \), in that the contact forces must be normal to the contact surface (see (12)) and that the change of momentum must also be normal to the surface (see (9)). Thus, to accurately conserve momentum and approximate the continuous equations of motion, a constraint function \( g(x) \) should have the property that \( \nabla g(x) \approx N_x(x) \), as in the point-mass example.

Some time integration schemes based on Lagrangian mechanics use a contact potential in the action functional and require a continuous interpenetration function that is at least sub-differentiable [19–23]. Such a potential is easy to construct for models of geometrically simple bodies and admissible configurations, but a good choice for this potential is much less obvious for complex geometries.

The preferred function for use in these applications has been a test for overlapping oriented simplices (OOS); i.e. tetrahedra in
three dimensions, and triangles in two dimensions (c.f. [17,20]). The OOS test can accurately provide a function $g(x)$, which indicates whether $I_q(x)$ is zero, but it is obvious from Fig. 2 that the gradient of the OOS test does not approximate $N_q(x)$. This is due to the non-global nature of the OOS test, which does not consider the body as a whole, but rather considers overlapping segments that compose triangles in two dimensions or overlapping triangles that compose tetrahedra in three dimensions. This has the further effect that the application of forces according to (12) may be appropriate to correct the overlap of segments in two dimensions or triangles in three dimensions, but it may not effectively correct the overlap between the bodies composed of these simplices. Thus, the OOS test is inadequate in particular for systems that do not have the capacity to absorb these spurious forces through deformations or motions; for example, in crowded or tightly-packed systems, stiff systems, and rigid body dynamics an accurate gradient $Vg(x) \approx Nq(x)$ is essential.

The OOS test has another key shortcoming. Before it can be evaluated, the algorithm must first determine whether the two segments or triangles of interest are indeed overlapping. Other versions of the test call for determining the type of contact first (face-edge, face-corner, face-face). These different ‘switches’ that precede the actual evaluation of the OOS test are particularly cumbersome when the time-integration algorithm is implicit or calls for optimization to resolve the contact configuration as they amount to changing the potential energy function or contact constraints as the algorithm is trying to converge.

Our function, which we have introduced as the supporting separation hyperplane (SSH) algorithm and is outlined in Algorithm 1 in Section 6 compares favorably to the OOS approach because (1) it is global in that it considers whole bodies and it does not require any ‘switches’ or additional calculations to classify types of contact that have taken place, and (2) it always supplies a gradient direction in the normal cone of the contact configuration, i.e., the closest admissible configuration to the present inadmissible configuration. Furthermore, the force system described by $-\nabla V_s(x)$ in a contact configuration is local to the features on each body involved in the collision. Finally, the dual solution to the SSH linear program can be used to determine closest-feature information and to determine an excellent approximation to the exact point of contact.

### 1.3. Organization

The development of the SSH algorithm, including the derivation of its subderivative is illustrated in the workflow diagram in Fig. 3. The algorithm is based on concepts and theorems from convex and affine geometry, and can be reduced to a quadratically constrained linear program (QCLP) for polyhedral bodies with isolated extreme points. However, the structure of this QCLP is such that an equivalent linear program (QCLP) for polyhedral bodies with isolated extreme points. Moreover, the force system described by $-\nabla V_s(x)$ in a contact configuration is local to the features on each body involved in the collision. Finally, the dual solution to the SSH linear program can be used to determine closest-feature information and to determine an excellent approximation to the exact point of contact.

#### 2. Hyperplanes and affine geometry

We use this section to review key definitions and theorems related to hyperplanes and affine sets in $\mathbb{R}^n$, which we have adapted from Rockafellar [26] and Urruty [27], and we refer the reader to these resources for precise definitions of bounded, closed, and compact sets, and the convex hull of a set. For this and the following sections, unless otherwise stated, we denote the inner product between two vectors in $\mathbb{R}^n$ as $\langle \cdot, \cdot \rangle$, the Euclidean norm of a vector in $\mathbb{R}^n$ as $\| \cdot \|$, and the cardinality of a set as $| \cdot |$.

##### 2.1. Affine sets

A subset $M \subset \mathbb{R}^n$ is called an affine set if for every $x \in M$, $y \in M$, and $\lambda \in \mathbb{R}$, $(1-\lambda)x + \lambda y \in M$.

##### 2.2. Convex sets

A subset $C \subset \mathbb{R}^n$ is convex if for all distinct points $x \in C$, $y \in C$, and all $0 \leq \lambda \leq 1$, $(1-\lambda)x + \lambda y \in C$. Clearly, all affine sets are also convex sets.

##### 2.3. Hyperplanes

Following [27], we use the notation $H(x)$ for the hyperplane in $\mathbb{R}^n$, with $x \in \mathbb{R}^n$ and $a \in \mathbb{R}$, as the set of points such that
which is an affine (and convex) set. We recognize that $\mathbf{a}$ is the normal vector to the plane, and that $H_{xa}$ has two distinct sides. An equivalent (point-normal) representation, with $\mathbf{a} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$, is

$$H_{xa}: = \{ \mathbf{x} \in \mathbb{R}^n | \langle \mathbf{x}, \mathbf{a} \rangle - a = 0 \},$$

(14)

where we can identify $a = \langle \mathbf{a}, \mathbf{x} \rangle$. For any choice of $\mathbf{a}$ and $\mathbf{b}$, the hyperplane associated with $\mu \mathbf{a} + \mathbf{c} \in H_{xa}$ is equivalent to $H_{xa}$ for all $\mu > 0 \in \mathbb{R}$. Therefore, without loss of generality, we can restrict $\mathbf{a} \in S^{n-1}$. Thus, for the signed distance between a point $\mathbf{y} \in \mathbb{R}^n$ and a hyperplane, we use the notation

$$H_{xa}(\mathbf{y}) := \langle \mathbf{y}, \mathbf{a} \rangle - a.$$

(16)

We further restrict the definition of parallel hyperplanes to denote two hyperplanes $H_{xa}$ and $H_{xb}$ for which $\mu = 1$. Thus, the signed distance from $H_{xa} \equiv H_{xb}$ to $H_{ya} \equiv H_{yb}$ is given by

$$d(H_{xa}, H_{xb}) = d(H_{ya}, H_{yb}) = b - a,$$

(17)

where the equivalence of the hyperplanes is due to $a = \langle \mathbf{a}, \mathbf{x} \rangle$ and $b = \langle \mathbf{b}, \mathbf{x} \rangle$, respectively.

2.4. Construction of convex sets

2.4.1. Intersection of half-spaces

Let us define the (closed) half-spaces associated with a hyperplane $H_{xa}$ as

$$H_{xa}^+: = \{ \mathbf{x} \in \mathbb{R}^n | H_{xa}(\mathbf{x}) \geq 0 \}$$

(18a)

and$\$$

$$H_{xa}^-: = \{ \mathbf{x} \in \mathbb{R}^n | H_{xa}(\mathbf{x}) \leq 0 \}.$$ (18b)

It may readily be shown that the intersection of convex sets is also convex (c.f. Theorem 2.1 in [26]). Thus, the following set $K$ is convex (and closed)

$$K = \cap \{ \mathbf{x} \in \mathbb{R}^n | H_{xa}(\mathbf{x}) \leq 0, i = 1 \ldots j \}.$$ (19)

If $K \neq \emptyset$ and $j \rightarrow \infty$, then some portion of the boundary of $K$ (bd$K$) is said to be smooth. Furthermore, $K$ is a compact set if it is bounded. For finite, $j, K$ as defined above is non-smooth and is called a polyhedral set. In the case that a polyhedral set is bounded (and therefore compact), we will alternately call it polytope in $\mathbb{R}^n$ or simply a polyhedral body (also in $\mathbb{R}^n$), in reference to the physical applications of the algorithm to be developed in this paper.

2.4.2. Convex hull of extreme points

An extreme point $\mathbf{z} \in K$ is a point such that there are no two different points $\mathbf{x} \in K$ and $\mathbf{y} \in K$ for which $\mathbf{z} = (1 - \gamma)\mathbf{x} + \gamma\mathbf{y}$. $0 < \gamma < 1$. That is, $\mathbf{z}$ is an extreme point of $K$ if and only if $\mathbf{z} = (1 - \gamma)\mathbf{x} + \gamma\mathbf{y}$. $0 < \gamma < 1 \Rightarrow \mathbf{x} = \mathbf{y} = \mathbf{z}$. Let us denote the set of extreme points of $K$ as ext$K$. For a compact set, ext$K \neq \emptyset$, and all extreme points of $K$ are on bd$K$. Thus, we can alternately describe a set $K$, which is convex and compact in $\mathbb{R}^n$, as the convex hull of its extreme points, $K = \text{co}(\text{ext}K)$. By this construction, for a finite number of extreme points, $K$ is a polytope. This allows us to describe all points in $K$ as a convex combination of its extreme points. That is, for all $\mathbf{x} \in K$, $\mathbf{y} \in \text{ext}K$, and $\lambda_i > 0$,

$$\mathbf{x} = \sum_{i \in \text{ext}K} \lambda_i \mathbf{y}_i,$$

(20)

where:

$$\sum_{i \in \text{ext}K} \lambda_i = 1.$$

For a polyhedral set, the set of vertices is equal to the set of extreme points.

2.5. Supporting hyperplanes

A hyperplane $H_{xa}$ is said to support the set $K$ when $K$ is entirely contained in either $H_{xa}^+$ or $H_{xa}^-$, and bd$K \cap H_{xa} \neq \emptyset$. It is said to support $K$ at $\mathbf{x} \in K$ when, in addition, $\mathbf{x} \in H_{xa}$.

2.6. Normal cone to a convex set

For the following analysis, it suffices to present a normal cone definition that is strictly relevant to convex sets. However, it is important to note that the idea can be extended to non-convex sets and is important in non-smooth analysis [16–19]. The normal cone $N_{K}(\mathbf{x})$ to a convex set $K$ at the point $\mathbf{x} \in K$ may be defined as the set of directions $\mathbf{v} \in \mathbb{R}^n$ for which $K$ is in the negative halfspace of $H_{xa}$, i.e.

$$N_{K}(\mathbf{x}) = \{ \mathbf{v} \in \mathbb{R}^n | H_{xa}(\mathbf{v}) < 0 \ \forall \mathbf{x} \in K \}$$

(21)

Thus, it is obvious that $N_{K}(\mathbf{x})$ will contain more than one direction $\mathbf{v}$ when $K$ is a polyhedral set of dimension $n$ in $\mathbb{R}^n$ if $\mathbf{x}$ is in a feature of $K$ of dimension less than $n - 1$. Fig. 4 shows two examples of normal cones.

As it will be useful later, we here make note that for convex sets there is a one-to-one correspondence between finding a hyperplane supporting a set at a given point, and finding a direction in the normal cone at the point.

2.7. Separation of convex sets

The strict separation of sets is defined as follows: for two non-empty closed convex sets, $K_1$ and $K_2$ for which $K_1 \cap K_2 = \emptyset$ and $K_2$ is bounded, there exists $\mathbf{x}$ such that

$$\sup_{\mathbf{y} \in K_2} \langle \mathbf{x}, \mathbf{y} \rangle < \inf_{\mathbf{y} \in K_1} \langle \mathbf{x}, \mathbf{y} \rangle.$$ (22)

Furthermore, if $K_1$ is also bounded, then there exists $\mathbf{x}$ such that

$$\max_{\mathbf{y} \in K_2} \langle \mathbf{x}, \mathbf{y} \rangle < \inf_{\mathbf{x} \in K_1} \langle \mathbf{x}, \mathbf{y} \rangle.$$ (23)

Two compact convex sets are properly separated if:.

$$\max_{\mathbf{y} \in K_2} \langle \mathbf{x}, \mathbf{y} \rangle < \inf_{\mathbf{x} \in K_1} \langle \mathbf{x}, \mathbf{y} \rangle,$$ (24a)

and

$$\max_{\mathbf{y} \in K_2} \langle \mathbf{x}, \mathbf{y} \rangle < \max_{\mathbf{x} \in K_1} \langle \mathbf{x}, \mathbf{y} \rangle.$$ (24b)

In other words, a vector $\mathbf{x}$ is a separating vector if it can be associated with a hyperplane $H_{xa}$ that properly classifies all of the points $K_1$ and $K_2$ such that $K_1 \subset H_{xa}^+$ and $K_2 \subset H_{xa}^-$. The converse is also true. For two compact convex sets, if $\mathbf{x}$ cannot be found such that

Fig. 4. Schematic of concepts for hyperplanes and convex sets for a compact polyhedral set. Points $a, b, c, d, e$ form the set of extreme points of $K$, and $\mathbf{z}$ is not an extreme point. The hyperplane $H_{za}$ supports the set $K$ at $\mathbf{y}$ such that $K \subset H_{za}^-$. The normal cone of $K$ is shown at two points: $\mathbf{x}$ such that $N_K(\mathbf{x})$ is not unique, and $\mathbf{z} \in \text{ext}K$ where $N_K(\mathbf{z})$ is unique.


max_{y \in K_2}(x, y) \leq \min_{x \in K_1}(x, y), then K_1 \cap K_2 \neq \emptyset. Fig. 5 illustrates the
definition of proper (a) and strict (b) ... hyperplanes associated with these vectors.

3. Supporting separating hyperplane algorithm

In this section, we develop the Supporting Separating Hyperplane (SSH) algorithm as a test for interpenetration. We begin by
developing the algorithm for general compact convex sets as a constrained optimization problem with a linear objective function, and
then show that, for compact polyhedral sets, it can always be reformulated so that all constraints are linear, i.e., as a linear programming problem.

To begin, let us define \{H_{x,a}\}_{x \in K_1} as the set of supporting hyperplanes of K_1 at x \in bdK_1 such that K_1 \subset H_{x,a} and \{H_{b,b}\}_{b \in K_1} as the set of supporting hyperplanes of K_1 at y \in bdK_2 such that K_2 \subset H_{y,b}. Again, all normal vectors of \{H_{x,a}\}_{x \in K_1} and \{H_{b,b}\}_{b \in K_1} are unit vectors (i.e., \|a\| = 1). Let us define the function h(K_1, K_2) as follows, where the function d(H_{x,a}, H_{b,b}) was defined in (17) and is the distance between parallel hyperplanes:

\[
\tilde{h}(K_1, K_2) = \max_{H_{x,a}(x, y) \in K_1 \cap K_2} d(H_{x,a}, H_{b,b})
\tag{25}
\]

The solution to (25) is the maximum signed distance between parallel supporting hyperplanes of each set, with the restriction that each hyperplane properly classify its associated set.

Remark 1. From the point-normal description of a hyperplane, we have for two non-empty compact convex sets, h(K_1, K_2) = \tilde{h}(extK_1, extK_2). Since d(H_{x,a}, H_{a,a}) = d(H_{x,a}, H_{a,b}) \forall b_1, a_1 \in H_{x,a}, i.e., b_1 - a_1 \in H_{a,a}, \Rightarrow H_{x,a} \cap H_{a,b} = H_{a,a}. Therefore, if H_{x,a} supports K_1 at a_1 \in extK_1, there is an equivalent hyperplane that supports K_1 at a_1 \in extK_1, and

![Fig. 5. An example of strictly separable sets. The vector \( \beta \) and associated hyperplane \( H_{x,a} \) strictly separate \( K_1 \) and \( K_2 \), whereas the vector \( x \) and associated hyperplane \( H_{x,a} \) properly separate \( K_1 \) and \( K_2 \). Note that the hyperplane drawn labeled \( H_{x,a} \) is not a unique choice for the vector \( \beta \).](image)

\[
\hat{h}(extK_1, extK_2) = h(K_1, K_2).
\tag{26}
\]

Remark 2. The problem (25) is always feasible. For two compact, convex sets, we can always find two parallel supporting hyperplanes such that \( H_{x,a} \in \{H_{x,a}\}_{x \in K_1} \) and \( H_{a,b} \in \{H_{a,b}\}_{b \in K_1} \). This remark is obvious if \( K_1 \cap K_2 = \emptyset \), and at least one hyperplane exists that strictly separates \( K_1 \) and \( K_2 \). To show that this is true if \( K_1 \cap K_2 \neq \emptyset \), consider a plane \( \beta \), that properly separates two sets \( \forall x \in H_{x,a}, y \in H_{a,b} \). If we now set \( b_2 = (y, y') \) and \( b_1 = (x, x') \) and undertake a translation of \( K_2 \) by \( \epsilon \), we can always find such a plane, and further define two equivalent planes for which \( H_{x,a}(y') = 0 \) for some \( y' \in bdK_1 \) and \( H_{a,b}(x) = 0 \) for some \( x \in bdK_1 \). If we now set \( b_2 = (y, y') \) and \( b_1 = (x, x') \) and undertake a translation of \( K_2 \) by \( \epsilon \), we can always find such a plane, and further define two equivalent planes for which \( H_{x,a}(y') = 0 \) for some \( y' \in bdK_1 \) and \( H_{a,b}(x) = 0 \) for some \( x \in bdK_1 \).

![Fig. 6. Fig. a through Fig. c illustrate the cases discussed in Remark 2. One of the many possible sets of supporting hyperplanes in the domain of (25) is shown for strictly separable sets in Fig. a. In Fig. b and c, the circular set remains \( K_1 \), and the triangular set remains \( K_2 \), but they are not explicitly labeled to simplify the drawing. Fig. 6b shows a hyperplane \( H_{x,a} \) which is clearly in the domain of (25), and two arbitrarily chosen points, \( x' \in bdK_1 \) and \( y' \in bdK_2 \), at which \( H_{x,a} \) supports each set, as described in Remark 2. Finally, Fig. c shows the translation of \( K_2 \) by \( \epsilon \), and the corresponding hyperplanes in the domain of (25).](image)

Before proceeding, we can define a constrained optimization problem over \( x \in S^{-1} \) and \( a_1, a_2 \in \mathbb{R} \) that has the same solution to (25) and is useful in proving Theorem 1.

\[
\begin{align*}
\max_{x \in S^{-1}} & \quad a_1 \cdot x - a_2, \quad \text{Subject to:} \\
\text{ext} K_1 \subset H_{x,a_1} \\
\text{ext} K_2 \subset H_{x,a_2}
\end{align*}
\tag{27}
\]

Theorem 1. Two compact convex sets \( K_1 \) and \( K_2 \) are strictly (properly) separable if and only if \( h > (> 0) \).

Proof of Theorem 1. We first show that \( h > 0 \Rightarrow K_1 \cap K_2 = \emptyset \), and by extension that \( h = 0 \) implies that \( K_1 \) and \( K_2 \) are properly separable. From the definition of a supporting hyperplane, we must have \( a_1 = (x, x) \) for \( x \in extK_1 \) and likewise that \( a_2 = (x, x) \) by \( y \in extK_2 \). It follows from the constraints in Eq. (27) that \( max_{x \in extK_1}(x, x) \) and that \( min_{x \in extK_2}(x, x) \) at the solution. Therefore, we can do a portion of the optimization explicitly and see that

\[
\begin{align*}
h(K_1, K_2) &= \max_{x \in extK_1}(x, x) - a_2(x, x), \\
&= \min_{x \in extK_1}(x, x) - \max_{y \in extK_2}(x, y),
\end{align*}
\tag{28}
\]

subject to the same constraints as (27). From the definition of strict (proper) separation of sets, if we can find \( x \) such that
that this constraint may be relaxed (to a restriction that we note that the structure of (29) is linear in the objective function, thus $h > 0 \Rightarrow K_1 \cap K_2 = \emptyset$, and that $h = 0 \Rightarrow K_1 \cap K_2 \subset H_{a_1} \equiv H_{a_2}$.

To show that $h > 0 \Leftrightarrow K_1 \cap K_2 = \emptyset$, we note that the domain has been restricted such that $H_{a_1}$ properly classifies $K_1 \subset H_{a_2}$, and $H_{a_2}$ properly classifies $K_2 \subset H_{a_1}$. In this domain, the maximum distance $a_1 - a_2$ is positive (non-negative) only if, in addition, $K_2 \subset H_{a_1}$, and $K_1 \subset H_{a_2}$, in which case, by the definition of strict (proper) separation of sets, $x$ is a separating vector and both hyperplanes are separating hyperplanes. Therefore $h > 0 \Leftrightarrow K_1 \cap K_2 = \emptyset$, and $h = 0 \Leftrightarrow K_1 \cap K_2 \subset H_{a_1} \equiv H_{a_2}$.\end{proof}

**Corollary to Theorem 1.** Two compact convex sets $K_1$ and $K_2$ are not separable if and only if $h < 0$.

**Proof.** Follows directly from Theorem 1. \end{proof}

**Remark 5.** If two compact, convex sets are separable with $\dim K_1 = \dim K_2 = n$, then there is a compact set of directions $x \in S^{n-1}$ which render the QP (29) non-negative. This follows from Theorem 1 due to $0 \geq \max d(H_{a_1}, H_{a_2}) > d(H_{a_1}, H_{a_2})$.

**Remark 4.** If two compact, convex sets are separable with $\dim K_1 = \dim K_2 = n$, then there is a compact set of directions $x \in S^{n-1}$ which render the QP (29) non-negative. This follows from the observation that there is a compact set of points in bd$K_1$ and bd$K_2$ at which a separating hyperplane can support each set. Furthermore, with the restriction that $\dim K_1 = \dim K_2 = n$, this set will be contained in less than a hemisphere of $S^{n-1}$. We will denote this set of separating directions $S(K_1, K_2) = \{x \in S^{n-1} | K_1 \subset H_{a_1}, K_2 \subset H_{a_2}, x \in \mathrm{bd} K_1, y \in \mathrm{bd} K_2 \} \subset S^{n-1}$. An example of $S(K_1, K_2)$ for two sets in $R^2$ is shown in Fig. 8.

**Remark 5.** The QP (29) always has a finite solution so long as all $x$, and $y$, are finite. This follows from (1) the observation that the objective function only has a non-zero gradient in the $a_1 - a_2$ subspace, and (2) the structure of the first two sets of constraints establishes an upper bound for $a_1$, and a lower bound for $a_2$ for any $x$. Thus, the objective function is always bounded in its increasing direction for any $x \in S^{n-1}$, as can be seen in Fig. 9.
one feasible solution for any constraint that \( \theta \). The inner optimization problem then amounts to an optimization problem which are feasible solutions to (29), sketched in the corresponding diagrams superimposed on the plot.

Remark 6. The LP (30) is feasible for any choice of \( b \). To show this, we note that the following problem is equivalent to (29) as \( S(K_1, K_2) = \{ x \in S^{n-1} \mid K_1 \subset H_{ex}, K_2 \subset H_{xy}, x \in bdK_1, y \in bdK_2 \} \subset S^{n-1} \), which will be equal to the empty set if \( K_1 \) and \( K_2 \) are not separable.

Theorem 2. An optimal solution, \( \bar{a}_{opt} \), to (30) is equivalent to an optimal solution, \( \bar{a}_{opt} \), to (29) for a given choice of \( b \in S^{n-1} \) if the following statements hold.

(i) The open hemisphere of \( S^{n-1} \) centered on \( b \) contains of \( S(K_1, K_2) \) and \( \bar{a}_{opt} \), with \( \gamma > 0 \), or
(ii) The corresponding equivalent solutions to the LP are bounded from above. It is still possible for another feasible solution direction, \( \bar{a}_{opt} \), of the linear program to satisfy \( ||\bar{a}_{opt}|| > \gamma ||\bar{a}_{opt}|| \), with \( \gamma > 0 \). Trivially, if \( \text{sign}(\gamma') = \text{sign}(\gamma) \), then \( \bar{a}_{opt} \) remains the optimal LP direction. If \( \text{sign}(\gamma') = \text{sign}(\gamma) \), then we can calculate the point, \( a_{crit} \), at which the ray \( a = a_{crit} + a \), with \( a = (a_1, a_2) \) as components of the corresponding feasible solution to (29) intersects the level set \( H_{ex}(\bar{a}_{opt}) \).

Proof of Theorem 2. By statement (i), the solution direction \( \bar{a}_{opt} \) has an equivalent solution direction \( x_{opt} \in H_{ex} \). Also by statement (i), as \( ||x|| \to \infty \), \( \gamma \) becomes negative for directions \( \bar{a}_{opt} \). So all feasible solutions to the LP are bounded from above. It is still possible for another feasible solution direction, \( \bar{a}_{opt} \), of the linear program to satisfy \( ||\bar{a}_{opt}|| > \gamma ||\bar{a}_{opt}|| \), with \( \gamma > 0 \). Trivially, if \( \text{sign}(\gamma') = \text{sign}(\gamma) \), then \( \bar{a}_{opt} \) remains the optimal LP direction. If \( \text{sign}(\gamma') = \text{sign}(\gamma) \), then we can calculate the point, \( a_{crit} \), at which the ray \( a = a_{crit} + a \), with \( a = (a_1, a_2) \) as components of the corresponding feasible solution to (29) intersects the level set \( H_{ex}(\bar{a}_{opt}) \).

The inner optimization problem then amounts to an optimization problem over \( (a_1, a_2) \) for a fixed vector \( a = \hat{a} \), which is always a bounded, feasible problem according to the preceding remark. Thus, removing the constraint that \( x \in S^{n-1} \) will result in a problem with at least one feasible solution for any \( \hat{a} \).

With these preliminaries established, let us call \( a_1 - a_2 = \gamma \) a feasible solution to the QP (29), and \( a_1 - a_2 = \gamma \) an optimal solution to the QP. The corresponding equivalent solutions to the LP (30) will have the values \( \gamma ||x_{opt}|| \) and \( \gamma ||x_{opt}|| \), respectively, with \( x_{opt} \) satisfying \( H_{ex}(x_{opt}) = 0 \). Furthermore, we will use the notation \( a = (a_1, a_2) \) to denote the solution to the QP (29) is the \( a_1 - a_2 \) subspace, and \( a_{opt} \) the corresponding solution to the LP (30), with the relationship \( a = a_{opt} \). Before proceeding with the key theorem of this section, we recall the definition of the set of separating vectors for the sets given in Remark 4 as \( S(K_1, K_2) = \{ x \in S^{n-1} \mid K_1 \subset H_{ex}, K_2 \subset H_{xy}, x \in bdK_1, y \in bdK_2 \} \subset S^{n-1} \), which will be equal to the empty set if \( K_1 \) and \( K_2 \) are not separable.
In practice, choosing a vector $\beta$ a priori to meet the first requirement of Theorem 2 is not hard to do, in particular as the optimal QP solution decreases to zero or becomes negative and $S(K_1, K_2)$ shrinks to the empty set, which renders (i) less and less restrictive. We will not present a concise result on this topic, although we note that $S(K_1, K_2)$ could be explicitly calculated and a central direction in this set chosen if needed to ensure that (i) is met so that the optimal LP solution is bounded. Furthermore, if an unbounded LP is encountered—which in itself implies that the sets are separable—$\beta$ can simply be reset based on the unbounded direction, and the new problem solved to determine a separating direction.

Perhaps more importantly, choosing $\beta$ to meet the second requirement of the theorem is actually not essential to determine whether or not two sets are separable, or even to resolve a separating direction if they are. By Remark 3, if $K_1$ and $K_2$ are not separable, then any choice of $\beta$ will indicate this fact, by rendering the objective function of the LP (30) negative. However, it is possible to select $\beta$ which gives a ‘false positive’, for which the solution to the LP is negative, but the solution to the QP is positive, implying that condition (i) in Theorem 2 has not been met. To avoid this, $\beta$ should be initially selected so that $\beta$ and some subset $S \subseteq S(K_1, K_2)$ are in the same open hemisphere.

In other words, one does not need to find an equivalent LP solution to the QP (29) in order to obtain an accurate result for the SSH test using a linear program. Rather, one simply needs to solve a bounded LP which avoids the aforementioned ‘false positive’. A good a priori choice for a vector $\beta$ satisfying condition (i) of Theorem 2—or barring that, a direction such that $\beta$ and $S$ are in the same open hemisphere—is a unit vector along a direction connecting a point $y \in K_2$ to another point $x \in K_1$. One such selection is presented in Section 6.4.

4. Linear programming

For completeness, we present a brief review of the well-studied topic of optimality conditions and solution strategies for linear programs. While this will likely be a review for many readers, the formulation of the problem is essential to the evaluation of its SSH linear program’s subderivative.

To simplify notation in this section, we establish that the number of inequality constraints is $m_1$, the number of equality constraints in a linear program is $m_2$, and the number of primal variables is $n$. All vectors are to be understood as column vectors and are denoted by bold lowercase letters, for example $a \in \mathbb{R}^n$. The combination of two vectors $a_1 \in \mathbb{R}^{m_1}$ and $a_2 \in \mathbb{R}^{m_2}$ into an extended vector $a \in \mathbb{R}^{m_1+m_2}$ is denoted, with a slight abuse of notation, as $a = (a_1, a_2)$. Subscript notation such as $a_i$, denotes the $i$th component of $a$. Matrices such as $A \in \mathbb{R}^{m \times n}$ are denoted by bold uppercase letters, and the combination of matrices of compatible dimensions into entries in block matrices is denoted by means of square brackets, e.g., $[A_1, A_2]$. Subscript notation such as $A_i$, denotes the $i$th component of the $i$th row of $A$.

4.1. Conditions for optimality

We wish to consider linear programs of the form

$$F(A, b, c) = \max_{x \in X} c^T x$$

with $X = \{x \in \mathbb{R}^n | A_1 x \leq b_1, A_2 x = b_2, x \geq 0 \}$, for which $X$ is a set of feasible solutions to the (primal) problem (31). We later define $X^* \subseteq X$ as the set of optimal solutions to (31). In this problem, the total number of constraints is $m = m_1 + m_2$, and $A_1 \in \mathbb{R}^{m_1 \times n}, A_2 \in \mathbb{R}^{m_2 \times n}$, with $b_1$ and $b_2$ also of compatible dimensions. We alternately adopt the notation that $A = [A_1, A_2]$ and $b = (b_1, b_2)$.

Let us denote a feasible problem of the form (31) as a problem for which $X \neq \emptyset$ and $F(A, b, c)$ is not infinity. From the primal problem (31), we can write the Lagrangian function

$$L_\lambda(x, \lambda, \mu) = c^T x - \lambda^T (A_1 x - b_1) - \mu^T (A_2 x - b_2) + \mu^T x$$

where $\lambda = (\lambda_1, \lambda_2)$ and $\mu$ are lagrange multipliers. The Karush–Kuhn–Tucker (KKT) conditions that are necessary for optimal primal $(x^*)$ and dual $(\lambda^*, \mu^*)$ variables are

$$x^* \in X$$

$$\lambda^T (A_1 x^* - b_1) = 0$$

$$\lambda^T (c - A^T \lambda^*) = 0$$

and

$$\lambda^* \lambda^* \geq c, \lambda^* \mu^* \geq 0$$

where the $i$th constraint is said to be active if $\lambda_i \neq 0$. This corresponds an inequality being an equality at the solution.

To state the dual problem to (31), we first set $y = (y_1, y_2)$. The dual program is then given by

$$G(A, b, c) = \min_{y \in Y} \ y^T b$$

with $Y = \{y_1 \in \mathbb{R}^{m_1}, y_2 \in \mathbb{R}^{m_2} | A_1^T y \geq c, y_1 \geq 0 \}$, with its own Lagrangian function

$$L_\eta(y, \pi, \eta) = y^T b - \pi^T (A_1^T y - c) - \eta^T y$$

in which $\pi$ and $\eta$ are the Lagrange multipliers of the dual problem. From (34), we have the KKT conditions for the dual problem

$$y^* \in Y$$

$$\pi^* (A_1^T y^* - c) = 0$$

$$y^T (b - A \pi^*) = 0$$

and

$$A_1 \pi^* \leq b_1, \quad A_2 \pi^* = b_2, \quad \pi^* \geq 0$$

From the KKT conditions for the primal and dual problem, we see that if a solution exists to (31), then a solution also exists to (33), and that at the solution, $y^* = \lambda^*$, $x^* = \pi^*$, and $F(A, b, c) = G(A, b, c)$ (i.e., $c^T x = y^T b$). Furthermore, equality constraints in the primal problem lead to unrestricted variables in the dual problem, and inequality constraints in the primal problem lead to restrictions on the sign of the dual variables associated with those constraints. This is important, because the SSH LP is not in standard form as all primal variables are unrestricted in sign. The primal problem of the SSH LP is of the form

$$F(A, b, c) = \max_{x \in X} c^T x$$

with $X = \{x \in \mathbb{R}^n | A_1 x \leq b_1, A_2 x = b_2 \}$, and therefore the dual problem has the form

$$G(A, b, c) = \min_{y \in Y} \ y^T b$$

with $Y = \{y_1 \in \mathbb{R}^{m_1}, y_2 \in \mathbb{R}^{m_2} | A_1^T y = c, y_1 \geq 0 \}$.

In many applications either the primal solution vector, the dual solution vector, or both, are not unique. Primal degeneracy (multiple dual solutions) is quite common in applications where the number of constraints is larger than the number of variables. In particular, for problems with weakly redundant constraints active at the primal optimum, there are multiple dual solutions because there is not a unique set of constraints that could be active at the solution (c.f. [28–31]). Dual degeneracy (multiple primal solutions) occurs when the dual problem has weakly redundant constraints and in particular when there are linearly dependent columns in $A$. For these cases, let us denote the sets of optimal solutions $X^*$ and $Y^*$:
Again, due to the nature of the constraints, we know that the primal and dual optimal solution sets are polyhedral in nature. Thus, if all vertices of these polyhedral sets are known, we may express any optimal solution vector as the convex combination of these vertices, according to Eq. (20).

These ideas have been explored in great detail, primarily in the areas of economics and operations research, where the primal and dual solutions have useful interpretations, c.f. [28,29,32,33,31]. For instance, in resource allocation when there is a unique primal solution that is degenerate, the minimum and maximum values of $y_i \in Y$ represent the highest buying price and lowest selling price of a given resource. In a convex analysis sense, it was shown by Ak-
gul [29] that if we allow $F$ to be a function of $b$, keeping $A$ and $c$ fixed, then $F(b)$ is a non-decreasing piecewise linear convex function and the set of subgradients $\partial F(b)$ of $F$ at $b$ is given by $\partial F(b) = Y$.

4.2. Linear program solution strategies

In this section we develop the solution to the SSH linear program based on the primal simplex algorithm [30]. The structure of this solution is particularly useful in evaluating the subgradient of the SSH program in cases that have not been treated in the literature. However, in order to exploit this algorithm, problems must be represented in extended standard form, and problems like the SSH LP that are in non-standard form [35] must first undergo a change of variables to be represented in standard form.

4.2.1. Extended standard form

The extended standard form of the primal program (31) is given by

$$F^{ext}(A, b, c) = \max \{c^T x | x \in R^n, Ax = b, x \geq 0, x = 0\}$$

with the dual problem (33) is also associated, and therefore (by the KKT conditions) the sets of primal and dual solution vectors to $F$ and $F^{ext}$ coincide, as does the value of the maximum. In the extended problem, $x_0$ and $x_1$ are known respectively as the slack and artificial variables, and $c^s = 0$ (c.f. [31,30]). In practice, the constraint that $x_0 = 0$ is never explicitly enforced. Rather, $x_0$ are set to zero by heavily penalizing violations of this constraint and setting $c = -M1$ (with $M \in R$ and $1$ being a vector in $R^n$ with ones in all entries) or by a two phase method in which the first phase attempts to eliminate the equality constraints from the system.

4.2.2. Variables unrestricted in sign

In our problem of interest, [30], all primal variables are unrestricted in sign. This differs from the standard form of a linear program (31). In order to convert our problem with all variables $x$ unrestricted in sign to this form, we make the change of variables

$$x = x_0 - 1 \nu.$$  

We note that $\nu = \max \{ 0, -x_0 \}$, although this last relationship need not be explicitly enforced [34]. The $1$ values are called the positive part of their respective variable. We then solve a related problem for $x = \nu(1 \nu) \geq 0 \in R^{n-1}$, and reconstruct the solution to $x$ from (41). Details can be found in [34], however we note here that this change of variables corresponds to embedding the original con-
where \( \mathbf{0}_B \) is the \( m \times 1 \) zero vector indexed by \( B \), and all previously introduced index sets, as appropriate, are embedded in \( \mathcal{E} \). The values in \( \mathcal{E} \) are commonly referred to as the ‘reduced cost coefficients’. From this we can find a solution to the dual problem as

\[
y^* = \mathbf{c}_B.
\]

(47)

If the big-M method is used, then at the solution any remaining factors of \( M \) are subtracted off of \( P \) to find the dual solution of the original problem. If \( \mathbf{x}_B > \mathbf{0} \) for all components, then the optimal dual vector \( y^* \) given in (47) is unique, i.e., the primal solution is non-degenerate. However, if any component of \( \mathbf{x}_B \) is zero, then it is possible to alter the index set \( B \), and thus the alter the dual solution vector, without changing the primal solution vector or the optimal value of \( F(\mathbf{A}, \mathbf{b}, \mathbf{c}) \). Let us denote the index set \( T \subset B \) as the rows for which \( \mathbf{x}_T = 0 \). We subsequently use the notation \( \langle B \rangle \) to denote the set of unique index sets that may represent the solution to (40). \( y^* \) is of the form [35,29]

\[
y^* = \left\{ \mathbf{y}^* \mathbf{y}_T = \mathbf{c}_B + t \mathbf{(B}^{-1} \mathbf{B} \mathbf{V} \mathbf{T})_{\mathcal{T}, \mathcal{J}} \mathbf{c}_B T_{\mathcal{J}, \mathcal{S}} + t \mathbf{(B}^{-1} \mathbf{B} \mathbf{V} \mathbf{T})_{\mathcal{T}, \mathcal{B}, \mathcal{S}} \geq 0, \mathbf{t} \in \mathbb{R}^{|T|} \right\},
\]

(48)

where the notation \( \langle B \rangle \) refers to the submatrix with rows in the index set \( I \) and columns in the index set \( J \), and \( t \in \mathbb{R}^{|T|} \) is a vector of parameters. Obviously, if \( T = \emptyset \), then \( y^* = \{ y^* \} \).

Finally, the unique value of the maximum is given by

\[
F(\mathbf{A}, \mathbf{b}, \mathbf{c}) = \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_B^T \mathbf{y}_B^* = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} + \mathbf{(c}_B^T \mathbf{B}^{-1} \mathbf{V} - \mathbf{c}_B^T) \mathbf{x}_B^*.
\]

(49)

5. Nonsmooth analysis

Various authors, primarily in the fields of economics and operations research, have explored what is known in the literature as the ‘sensitivity’ of a linear program to its data \((\mathbf{A}, \mathbf{b}, \mathbf{c})\) beginning in the 1950’s and the Dantzig’s inception of the topic. That is, if \( F \) is a smooth function of \( \mathbf{A}, \mathbf{b}, \mathbf{c} \), the values of interest in sensitivity analysis are \( \frac{\partial F}{\partial \mathbf{A}}, \frac{\partial F}{\partial \mathbf{b}}, \text{ and } \frac{\partial F}{\partial \mathbf{c}} \). The former two are well understood, in particular for a program in standard form. However, the vast majority of economic analyses take the entries in \( \mathbf{A} \) to be fixed. Those analyses of the sensitivity of a linear program to the coefficients in the constraint matrix that do exist make use of several regularity conditions (e.g., the problems treated are in standard form, with no linearly dependent columns or rows) that in general are not present for the SSH LP [25,24,36].

For the SSH linear program, we are interested how \( F(\mathbf{A}, \mathbf{b}, \mathbf{c}) \) changes with respect to the entries in the original matrix \( \mathbf{A} \). An example of this dependency for an SSH linear program (30) that has been put into extended form and undergoes the change of variables in Section 4.2.2 is shown in Fig. 10. In this example, the \( z \)-coordinate of one tetrahedron is varied and the vector \( \mathbf{b} \) is held constant.

In general, the maximum function is neither a smooth nor a convex function of entries in \( \mathbf{A} \). However, we have shown that the SSH linear program is always bounded and feasible for an appropriate choice of \( \mathbf{b} \), and it follows directly from the structure of an LP solution described in Section 4.3.3 that for problems of the form (30) \( F(\mathbf{A}, \mathbf{b}, \mathbf{c}) \) is Lipschitz over \( \mathbf{A}_0 \in \mathcal{B} \). With this in mind, we take a general non-smooth analysis approach to understand the change in the solution with respect to the entries in \( \mathbf{A} \) in the original matrix for problems of the type (30) with the information at hand from a problem solved in a higher dimension using \( \mathbf{A} \).

5.1. Generalized differential

To begin, we introduce the generalized directional derivative from Clarke [16]. First, we let \( f \) be a Lipschitz mapping from \( f : X \rightarrow \mathcal{B}, \) where \( X \) has an associated dual space, \( Y \). We also define the duality pairing \( \langle \cdot, \cdot \rangle \), between \( X \) and \( Y \). Clarke’s generalized directional derivative is then given by

\[
f^\mathcal{B}(\mathbf{x}, \lambda) = \lim \sup_{t \to 0^+} \frac{f(\mathbf{x} + t \lambda) - f(\mathbf{x})}{t},
\]

(50)

where \( \mathbf{x}, \lambda \in \mathcal{X} \) and \( t \) is a positive scalar. Clarke’s generalized subdifferential (or generalized gradient, which for simplicity we alternately refer to as the subgradient) of a Lipschitz function \( f \) at \( \mathbf{x} \) is the subset

\[
\partial f = \{ \mathbf{y} \in Y | f(\mathbf{y} + \lambda) \geq f(\mathbf{y}), \forall \lambda \in \mathcal{X} \},
\]

(51)

for \( \mathbf{y} \in \mathcal{Y} \). The generalized directional derivative can be recovered from the generalized gradient as

\[
f^\mathcal{B}(\mathbf{x}, \lambda) = \max \{ \langle \mathbf{y}, \lambda \rangle | \mathbf{y} \in \partial f \}.
\]

(52)

In the following sections, we develop an expression for the generalized directional derivative and use this result to recover and expression for the generalized gradient.

5.2. Generalized directional derivative of \( F(\mathbf{A}) \)

We first develop an expression for \( F^\mathcal{B}(\mathbf{A} : \mathbf{A}, \mathbf{b}, \mathbf{c}) \), which we shorten to \( F^\mathcal{B}(\mathbf{A}, \lambda) \), in the case where both (31) and (33) may be degenerate. Following Freund [25,24], Williams [37] and Mills [36], we adopt a perturbation matrix \( \lambda \in \mathbb{R}^{|\mathcal{X}||\mathcal{B}|+1} \), with a one in the entry of interest, and zeros in all other entries.

\[
\mathbf{A} = \left[ \begin{array}{c|c|c} \mathbf{e}_1 & \mathbf{0} \end{array} \right] \in \mathbb{R}^{|\mathcal{X}|(1+|\mathcal{B}|)}.
\]

(53)

In (53), \( \mathbf{e}_1 \in \mathbb{R}^{|\mathcal{X}|} \) is a row vector with zeros in all entries, and \( \mathbf{e}_1 \in \mathbb{R}^{|\mathcal{B}|} \) is a row vector with a one in the entry of interest, and zeros in all other entries. Let us call \( \mathbf{A}_1 = \mathbf{A} + t \lambda \). The solution to \( F(\mathbf{A} + t \lambda) = F(\mathbf{A}) \) is given by

\[
F(\mathbf{A}_1) = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} - \mathbf{(c}_B^T \mathbf{B}^{-1} \mathbf{V} - \mathbf{c}_B^T) \mathbf{x}_B^*.
\]

(54)

We assume that the perturbation leaves \( \mathbf{B} \) invertible, which is certainly reasonable as \( \mathbf{t} \to 0^+ \). In this expression, the index sets \( \mathcal{B} \) and \( \mathcal{V} \) now subdivide the perturbed matrix \( \mathbf{A}_1 \) into \( \mathbf{A} \) and \( \mathbf{V} \), and \( \mathbf{A} \) into \( \mathbf{A}_0 \) and \( \mathbf{A}_1 \). We have the relationship

\[
\mathbf{B}^{-1} = (\mathbf{B} + t \lambda_1)(\mathbf{B} + t \lambda_2)^{-1} = \mathbf{I},
\]

(55)

which can be premultiplied by \( \mathbf{B}^{-1} \) and rearranged such that

\[
\mathbf{B}^{-1} = \mathbf{B}^{-1} - t(\mathbf{B}^{-1} \lambda_1 \mathbf{B}^{-1}),
\]

(56)

from which we can derive an expression for \( \mathbf{B}^{-1} \) as an infinite series

\[
\mathbf{B}^{-1} = \sum_{t=0}^\infty t(\mathbf{B}^{-1} \lambda_1) \mathbf{B}^{-1}.
\]

(57)

This series converges for all \( |t| < \epsilon = (k \| -\mathbf{B}^{-1} \lambda_1 \|)^{-1} \) for some \( k > 0 \). Using \( \mathbf{B} = \mathbf{H} + t \lambda_1 \), this allows us to develop an explicit expression for the generalized directional derivative of \( F(\mathbf{A}) \) as
polyhedrons so that the problem which lie at vertices of the feasible primal and dual index sets

We can find a complementary dual solution which satisfies:

\[ y^T B = c_t. \]  (59)

This allows us to write (58) as:

\[ F'(A; \Lambda) = \sup_{y^T \Lambda B^*} - y^T A y, \]  \hspace{1cm} (60)

where \( Y^t \subset Y^* \) is the set of basic feasible dual solutions. However, because of the affine nature of \( X' \) and \( Y' \), (60) is equivalent to:

\[ F'(A; \Lambda) = \sup_{y^T \Lambda B^*} - y^T A y. \]  \hspace{1cm} (61)

Note that in (61), we have not assumed anything about the sign of \( x \) or whether (or not) the problem is primal degenerate, dual degenerate, or both.

5.3. \( F'(A; \Lambda) \) for unrestricted variables

We now use the expression in Eq. (61) to evaluate the generalized directional derivative in the case of the specific change in variables given in Section 4.2.2. For this problem, the structure of \( X' \) for the augmented problem \( F'(A; B, c) \) is such that we fully expect to find multiple solutions \( \lambda' \in X' \) (c.f. [34]), which suggests that it is possible for the generalized directional derivative \( F'(A; \Lambda) \) to take on infinite values. Luckily, we are not interested in \( F'(A; \Lambda) \) at face value, but rather in its projection onto a lower dimensional space. This may be accomplished by introducing a linearly dependent column into the perturbation matrix, which reflects the relationship \( \lambda = \lambda - 1 v \).

In order to find the generalized directional derivative with respect to the data of the original problem, \( \Lambda \) takes on the form \( \Lambda \)

\[ \Lambda = \begin{bmatrix} 0^t & 0 & 0^m \\ \vdots & \vdots & \vdots \\ 0^t & 0 & 0^m \\ \end{bmatrix} \in \mathbb{R}^{n \times (n+m+1)}. \]  \hspace{1cm} (62)
The generalized directional derivative with respect to the original entries in $A$ is then
\[
F^* (\tilde{A} \cdot \tilde{\lambda}) = \sup_{y \in \gamma \cdot \mathbb{R}^n} - y^T \tilde{\lambda} x.
\]
However, we know that $Y = \gamma \cdot \mathbb{R}^n$ and that
\[
\tilde{\lambda} x = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \xi_j - y \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \xi_j \end{pmatrix} = \xi_j A x.
\]
From these relationships, we have that:
\[
F^* (\tilde{A} \cdot \tilde{\lambda}) = F^* (A \cdot \lambda) = \sup_{y \in \gamma \cdot \mathbb{R}^n} - y^T \lambda x.
\]
From Eq. (65), we can infer a simple expression for the generalized gradient as:
\[
\partial F^* (A \cdot \lambda) = \{ -y^T \lambda x | y \in \gamma \cdot \mathbb{R}^n, x \in X^* \}.
\]
Finally, we note that if the problem is not degenerate, then the generalized gradient is the normal gradient, with the value
\[
\nabla F (A) = -y^T \lambda x.
\]

**Remark 7.** The SSH LP is structured so that any solution vector, which we call $x$ in (30), is restricted to lie in the normal cone or its reflection at some point on the boundary of each set. Furthermore, the KKT conditions state that the dual variable corresponding to the $i$th constraint is zero if that constraint is not active at the solution, and non-zero if the constraint is active. In the SSH problem, a non-zero dual value has a one-to-one correspondence to the associated vertex being a support point for one of the optimal set of hyperplanes from the SSH test, and therefore being associated with a feature involved in a collision. Furthermore (for a non-degenerate case) the expression for the gradient reads
\[
\nabla x g = y_i x, \quad x_i \in \text{ext} K_1
\]
\[
\nabla x g = -y_i x, \quad x_i \in \text{ext} K_2,
\]
where $y_i$ is the dual variable associated with the $i$th constraint. For degenerate problems, the generalized gradient has a similar structure, in that it is always in the direction $x$ or $-x$, modulo a constant non-negative factor ($y_i$). Therefore, not only is the subgradient non-zero for only those vertices associated with features involved in the collision, but, because we have solved for $x$ via optimization, the (generalized) gradient satisfies $\partial g \in -N(x)$, where $x$ is a feature involved in the collision.

**Remark 8.** The convention for an interpenetration constraint function is that $g(x) < 0$ implies an admissible configuration, and $g(x) > 0$ implies interpenetration. This can clearly be achieved by the program in Eq. (69).
\[
g(K_1, K_2) = - \max_{x \in \mathbb{R}^n} a_1 - a_2. \quad \text{Subject to:} \]
\[
A_1 x = a_1 \geq 0, \quad x \in \text{ext } K_1
\]
\[
A_2 x = a_2 \leq 0, \quad y \in \text{ext } K_2
\]
\[
\langle b, x \rangle - 1 = 0
\]
For this program, the generalized gradient is the same as in (68), multiplied by a factor of $-1$. In this case, $\partial g \in N(x)$ for features in each body involved in the collision, as illustrated for several different collisions between two tetrahedra in three dimensions in Fig. 12. Note that a force system $f \in -\partial g$ will be in the direction of normal forces on the system due to contact.
6. Implementation
6.1. Algorithm overview

In the previous sections, we have focused developing the SSH LP for convex polytopes from the SSH QCLP for general convex bodies, and shown that the SSH LP’s subdervative may be evaluated analytically in closed form once a solution to the SSH LP is known. The algorithm is meant as a general fine scale collision detection algorithm for polytopes, however the following pseudocode effectively summarizes the overall algorithm. The results of the algorithm are whether or not a collision has occurred, and the optimal primal and dual variables, or their range in the case of degenerate problems.

1: Run coarse scale collision detection 
2: for all Pairs of possibly intersecting polytopes, \((K_1, K_2)\) do 
3: Calculate \(\tilde{b} \), assemble equality constraint vector \(A_2\) 
4: Assemble inequality constraint matrix \(A_1\) from vertex sets of bodies 
5: Evaluate SSH LP (Eq. (30)) using the Simplex method 
6: if Subderivative information is needed then 
7: if Degeneracy detected then 
8: Follow Akgul [29] to determine \(\mathbf{x}', \mathbf{y}'\) range 
9: end if 
10: Return \(g(K_1, K_2), \mathbf{x}', \mathbf{y}'\) (range) 
11: else 
12: Return \(g(K_1, K_2)\) 
13: end if 
14: end for

6.2. Computational complexity

The computational complexity for the fine scale collision detection test is equivalent to that of evaluating the SSH LP, which is limited by the number of operations needed to invert a matrix. For each pair of bodies, this amounts to \(C_{\text{LP}}(\text{ext}K_1 + \text{ext}K_2 + 1)^4\) floating point operations (flops), where \(C_{\text{LP}} \approx 2\) after initialization so that the SSH LP can be warm started from the last known basis. In order to compare to the OOS test introduced in Section 1, it is important to recall that the OOS test requires at least the surface of the sets \(K_1\) and \(K_2\) to be triangulated. Using the notation \(E_{n-1}(T)\) to denote the set of simplices of dimension \(n-1\) the triangulation of the \(j^n\) set, the OOS test requires \(C_{\text{OS}}(E_{n-1}(T_1)||E_{n-1}(T_2))\) flops per pair of bodies, and \(C_{\text{OS}} = 30\) for \(n = 3\) and \(C_{\text{OS}} = 7\) for \(n = 2\), for the test in [17]. So, if one is only interested in whether or not the bodies are penetrating, a test based on overlapping oriented simplices is the most efficient choice so long as the triangulation of the bodies does not render \(|E_{n-1}(T_1)||E_{n-1}(T_2)| \approx (\text{ext}K_1 + \text{ext}K_2 + 1)^3\). However, we would suggest that for many applications the fact that the bodies in consideration need not be triangulated as well as the additional information provided by the SSH LP evaluation merits its use. Furthermore, if one is not judicious in the triangulation of the bodies, it is possible for \(|E_{n-1}(T_1)||E_{n-1}(T_2)| \approx (\text{ext}K_1 + \text{ext}K_2 + 1)^3\), in which case the SSH LP approaches the efficiency of the OOS test.

6.3. Linear programming solver

Because of the small dimension of the proposed linear program \((n = 6, m = 9\) for two tetrahedra) and the known shortcomings of commercial software in adequately describing the full solution set in the case of primal degeneracy, a basic two-phase version of the revised primal simplex method was implemented following [38,31,30]. Note that we did not implement the Big-M method, as in practice it can lead to significant conditioning problems (see [38,31,30] for details). Using this approach, a dual solution is immediately available at the end of the optimization. For degenerate cases, which occur quite infrequently in practice, the method of Akgul [29] is used to resolve the component-wise range of the dual and primal solution sets. Finally, after initialization, we use a ‘warm start’ procedure by starting the optimization at the last known optimal basis (index set \(B\)).

6.4. Choice of \(\beta\)

All examples were run with the vector \(\beta\) set to point from the arithmetic mean of \(\text{ext}K_1\) to the arithmetic mean of \(\text{ext}K_1\),

\[
\beta = \frac{1}{|\text{ext}K_1|} \sum_{k \in \text{ext}K_1} \mathbf{x} - \frac{1}{|\text{ext}K_2|} \sum_{j \in \text{ext}K_2} \mathbf{y}
\]

for finite element simulations, or a vector from the center of mass of \(K_2\) to the center of mass of \(K_1\) in rigid body simulations. Inasmuch, this choice of \(\beta\) works well so long as the arithmetic means are not the same point (causing \(\beta = 0\)), which we expect to be the case for most applications. This choice of \(\beta\) renders the SSH LP continuous over the entire domain of the simulation, because \(\beta\) is made a smooth function of the data. It is worth noting that \((70)\) is not the only acceptable choice for \(\beta\). For instance, if the method of Chung and Wang [9] is used as a coarse search for collisions, then the separating vector from this approach may be used, or the optimal direction \(\mathbf{a}\) from the previous solution to the problem could also be used.

The downside of our choice for \(\beta\) is that it does induce a dependence on \(\text{ext}K_1\) and \(\text{ext}K_1\) in the equality constraint of the problem (30) and therefore the chain rule must be used in the evaluation of the subgradient. If (30) is being used directly as a contact potential and its subgradient to determine contact forces, this choice of \(\beta\) causes forces on vertices that do not bound features involved in the collision. However, the value of the dual variable associated with the equality constraint, and therefore the magnitude of these non-local forces, is exactly the value of the optimum because, at the solution, \(\mathbf{b}'\mathbf{y}' = \mathbf{c}'\mathbf{x}\) for all \(\mathbf{x} \in X^*\) and \(\mathbf{y}' \in Y^*\), and there is only one non-zero entry in \(\mathbf{b}\). Again, if a given simulation is set up so that any overlap of colliding bodies is small, then the relative magnitude of the non-local forces will also be small. Alternately, these forces may simply be ignored in order to approximate \(N_{\text{CLP}}(\mathbf{x})\) in the closest admissible configuration, where \(g(\mathbf{x}) = 0\).
To begin, let us index the rows of \( A \) (columns of \( A^T \)) by the index set \( J_d \). We may identify the set of all active constraints as \( A_{d} \subset J_d \) as the equality constraint plus the inequality constraints associated with dual variables \( y_i \) for which \( \max_{y_i \in Y^*} y_i = 0 \). Because all constraints in the dual problem to the SSH LP are equality constraints we have identified a reduced underdetermined system for the active dual variables

\[
(A_{d} y)_{y} = c, \tag{71}
\]

where \( A_{d} \) refers to the submatrix of \( A \) whose rows correspond to weakly redundant constraints at the solution, plus the equality constraint. The Moore–Penrose pseudo-inverse can be used to find a unique solution to this system that gives all of the non-zero entries in \( y^* \in Y^* \). This method corresponds geometrically to the orthogonal projection of the origin of the dual space onto the affine optimal solution set \( Y^* \), and is precisely the correct operation to select \( y^* \in Y^* \) because the structure of the dual constraint set is such that symmetries in the contact configuration correspond to symmetries of the optimal set with respect to the origin of the dual space. For example, in a perfectly aligned face-face collision, the orthogonal projection of the origin of the dual space onto \( Y^* \) is equidistant from the vertices of \( Y^* \).

The case of multiple primal solutions (dual degeneracy) for the SSH LP is easier to visualize, and corresponds in two dimensions to vertex-vertex contact between bodies, as is shown in Fig. 11b. In this case, the set of dual constraints active for all primal solutions can be readily found. We can write an underdetermined system for an optimal primal solution vector as

\[
A_{d} x = b_{d}, \tag{72}
\]

where for the SSH LP, \( x = (\tau_1, \omega) \). Again, this solution is the orthogonal projection of 0 in the primal space onto \( X^* \). Perhaps
a more useful physical interpretation of the minimum norm solution \( x \in X \) to an underdetermined SSH primal problem is exactly that it is the solution with the smallest Euclidean norm, which corresponds to the direction in which the smallest contact force or impulse should be applied to prevent the interpenetration of matter, subject, of course, to any bias imposed by the choice of \( b \).

Fig. 12 shows several collisions of two tetrahedra in three dimensions. The subgradient with respect to each configuration variable—the vertex positions in this case—is plotted using \( \beta \) in Eq. (70), and the minimum norm solution to primal and/or dual variables in several degenerate cases. In these figures, the scale of the vectors reflects the relative magnitude of the dual variable associated with each constraint, and the direction reflects the primal solution, \( x \).

7. Examples

7.1. Collision type and closest features

The dual solution to the SSH linear program can be used directly to determine the type of collision that has taken place. As we noted in the remarks at the end of the previous section, any non-zero dual variables correspond to active constraints at the solution. The number of active constraints associated with each body can be used to determine the type of contact that has taken place. This can be seen in Fig. 12, where vertices associated with non-zero dual variables are highlighted. For example, if both bodies are tetrahedra, if there is one non-zero dual variable in the first set of constraints, and three non-zero dual variables in the second set, then a vertex of the first body is in contact with a face of the second body.
If a collision has not occurred, then the closest features can be determined by the same method. Furthermore, the closest vertex in $K_1$ to $K_2$ is the vertex associated with the largest dual variable. This can be seen from the generalized derivative developed in Section 5; i.e., if we examine the expression in (67), we see that if the dual variable for a given vertex constraint is zero, then small changes in that constraint do not affect the value of (30) and that constraint is redundant in the given configuration. Likewise, small changes in the position of the vertex associated with the largest dual variable cause the largest changes in the value of the SSH linear program.

### 7.2. Exact intersection point

Once the closest features have been determined from the dual solution to (30), this information can be used to predict the inter-
section point or to calculate it after contact has occurred through a simple projection operation. The exact operation depends on the type of (predicted) contact that has occurred.

7.3. Optimal impulse for dynamics

The following examples use a simple explicit Newmark time-integration scheme with a mid-point rule for the velocity calculation and a contact potential, as discussed in the introduction. The subgradient of the modified SSH linear program (69), multiplied by a constant factor, is used directly to determine contact forces in both rigid body and finite element simulations. These examples emphasize that the use of the SSH linear program directly as an approximation to the indicator function allows us to successfully model complicated collisions, non-convex geometries, and extremely stiff systems using only the most basic explicit time-integration scheme with a relatively large time-step.

![Fig. 19. Geometry and mesh of bodies used in the kinematic latching simulation.](image1)

![Fig. 20. Transparent top-down view of pins being guided into conical holes at \(t = 0.75\) s.](image2)

(a) Translational momentum  
(b) Angular momentum  
(c) Energy  

![Fig. 21. Momentum and energy histories for Neo-Hookean 'satellite' latching simulation.](image3)
7.3.1. Clumping rigid hexagons

Each of the following two examples uses a group of 16 identical hexagonal rigid bodies. Collisions are modeled as frictionless and inelastic, with a coefficient of restitution \( e = 0.8 \), and the time step is \( 10e^{-3} \) seconds. For both examples, the initial positions and velocities are the same, as shown in Fig. 13, where the relative magnitude and direction of the translational velocity is shown, and each body also has a random initial angular velocity.

In the first example, there are no potentials aside from the contact potential given by the SSH LP. After several collisions, occasionally with multiple collisions in one time step, the bodies disperse, see Fig. 15.

In the second example, each identical hexagonal body is equipped with alternating positive and negative Coulombic charges centered on each face and off-set slightly from the surface to avoid a singularity in the calculation of the Coulombic potential. Due to the strong attractive potential, the bodies lock together into a regular grid, see Fig. 16. Note that the second system, in particular, becomes extremely stiff as the bodies lock together, and the SSH LP contact potential successfully allows the bodies to oscillate around a minimum Coulombic potential energy, as illustrated in Fig. 14.

7.3.2. Cluster of cubes

A cluster of 16 cubes made of a neo-Hookean material \( (\lambda = 10e3 \text{ Pa, } \mu = \lambda/2) \), and modeled using lumped mass tetrahedral conforming solid elements with first order interpolation is considered next. All cubes start with an inward-radial velocity towards the cube in the center of the cluster proportional to the distance between each body’s center of mass and the origin, and no angular velocity. This is done specifically to induce collision configurations with primal and dual degeneracies. The timelapse images in Fig. 17 illustrate qualitatively the preservation of symmetry in the pre- and post-collision trajectories of the cubes, again using explicit Newmark integration with a timestep of \( 10e^{-4} \) seconds. Fig. 18 shows the energy and momentum behavior of the entire system of bodies in the simulation. The total energy, and the translational and angular momenta are preserved for the lumped-mass model.

Fig. 22. Timelapse images of kinematic ‘satellite’ latching simulation.
7.3.3. Kinematic ‘satellite’ docking

The following example also makes use of lumped-mass tetrahedral conforming linear solid elements which model a relatively stiff ($\sigma = 10^4$ Pa, $\mu = \lambda/2$) neo-Hookean material using a 10e -- 5 s. In this coarse model, two identical bodies are used, each with a set of pins, and a complementary set of holes. One such body is shown in Fig. 19. As can be seen in Figs. 19 and 20, the diameter of the bulge in the forward portion of the pins and of the cuff at the base of the pins is larger than the smallest diameter of the holes, so that both pins and holes must deform in order for latching to take place. The distance between the cuff and the base of the pin is slightly larger than the thickness of the shell. Fig. 22 shows several configurations of the bodies as time advances, and Fig. 21 shows the energy and momentum behavior of the two body system over the course of the simulation. Both of these figures show the sustained contact between the bodies once latching has taken place. Furthermore, note that the body on the right, which has a velocity of magnitude 3.5 m/s at $t = 0$ s in the negative x-direction, is initially off-set in the y-direction from the body on the left, which is initially at rest, so that the conical holes must guide the pins into alignment. A closeup of this alignment is shown in Fig. 20.

8. Conclusion and future directions

This paper has developed the supporting hyperplane (SSH) test for interpenetration, which can be formulated and solved as a linear program for polyhedral convex bodies as the SSH LP. We further show that the subdifferential of the SSH LP exists, is local, and lies in the normal cone of a contact configuration. Several examples illustrate the usefulness of the primal and dual solutions to (30) in determining closest features, projected collision points, and the force system at the time of contact.

We note in closing that the SSH test for interpenetration also offers advantages in friction formulations, collision integrators, and control algorithms. Its utility in formulating models of friction stems from the fact that the friction surface, like the optimal force system, can be determined directly from the primal and dual solutions (cf., e.g., [18,20]). The SSH algorithm is also well-suited to developing collision integrators for Lagrangian mechanics, e.g., within the framework of constrained variational integrators (cf., e.g., [17,18]). Finally, we see useful potential applications of the SSH test, e.g., within the framework of discrete mechanics and optimal control [39], to the control of systems of multiple non-smooth or non-convex bodies that tend to–or need to–cluster, e.g., to execute docking or self-assembly maneuvers.

Acknowledgements

The support of the W.M. Keck Foundation through Caltech’s Keck Institute for Space Studies is gratefully acknowledged.

References