American Journal of Computational and Applied Mathematics 2014, 4(3): 97-110
DOI: 10.5923/j.ajcam.20140403.05

Solving System of Linear Inequalities or Equalities on the Surface of the Unit Shell (LIS-III)

Paul T. R. Wang

WangPaul_Research

Abstract The author presents an innovative concept of a generic algorithm that solves system of homogeneous linear inequalities for finite number of unknowns and constraints utilizing normalized unit vectors of length 1 on the surface of an n-dimensional hyper sphere with a radius of 1 coined as the unit shell and the concept of equal distanced points as sysmmetic points along the arc connecting two equal distanced points to a selected set of points on the unit shell with increasing ranks to locate the desired solution point or points of the given system of linear inequalities if such solution point or points exist. A direct application of this innovative technique applied to a linear program formulated as a system of self-dual homogenerous linear inequalities is illustrated to establish its validity. Furthermore such technique is also illustrated to extend its applicability to solve Differential Variation Inequalities (DVI) over the Unit-Shell. Such a new technique is shown to be extremely efficient, numerical stable, and quite suitable for very large system of linear inequalities with number of variables and/or constraints over millions. In addition, such a new approach does provide insight to solve linear inequalities that is compatible to Gaussian elimination solving linear equalities.

Keywords Homogeneous Linear Inequalities, Unit-Shell, Liner Programs, Linear Projection and Distance to subspace, n-dimensional normed linear space, and differential variation inequalities

1. Introduction

Many years back, the author was working on the hemispherical cover for unit vectors on n-dimensional unit shell as the surface of an unit sphere. If we normalize all row vectors of a linear inequalities system (LIS) in its feasibility form ($f = Lx \geq 0$), they are points on the unit shell of this n-dimensional hyper sphere. Note that $Lx \geq 0$ has a solution implies that all the normalized row vectors of L and the normalized solution, x , has a dot product ≥ 0 . Hence the hemispherical surface of the unit sphere has a center at $x_u = x / \| x \|$ with all points $u_i = L_i / \| L_i \|$ reachable within an arc distance of $\pi / 2$ or less. In other words, we have a hemispherical cover $C(x_u, \pi / 2)$ covering all the points of $\{u_i\}$. Such a solution of the LIS is also a point on this unit shell as the geometrical center (a point with its maximum arc distance to all the constraint points minimized) of this hemispherical cover. Note that LIS is generic for all linear systems equalities and inequalities are both included. In particular linear programs (LPs) is only a special case of LIS [ref. 17].

* Corresponding author:
wangpaul19@gmail.com (Paul T. R. Wang)
Published online at http://journal.sapub.org/ajcam

The pursuit to locate a unit vector x_u for $f = Lx \geq 0$ did not succeed then as the geometrical center of a fixed number of points (especially when we are dealing with millions of points!) on a unit shell is very troublesome to locate in n-dimensional hyper space. However, now an effective approach has been identified. Two new insights make such a pursuit feasible and verifiable. First, the concept of "equidistant" for a specific point as an equal distanced point (EDP) to a group of points on the unit shell can be maintained using orthogonal subspace projection. Second, equidistance can be maintained by moving an EDP on the n-dimensional great circle arc connecting two equidistant points with respect to a given set of points. By repeatedly applying this EDP tracking strategy to a set of linearly independent unit vectors with increasing rank as to increase the number of points that are equidistant to the center point as its geometrical center, the solution for the original LIS can easily be located with a minimum number of steps no more than the maximum possible rank of the defining matrix L of the LIS. Such a technique is greedy, non-iterative, non-linear, and non-Gaussian, and use only the dot-product operator which is O(n) in operation. No basis vectors swapping is required to minimize the maximum arc distance from a given point to the geometrical center or centers of a set of points on the unit shell. Furthermore, most matrix operations carried out on the unit shell are over unit vectors; hence it is numerical stable avoiding ill-conditioned operations. Consequently, the unit shell approach can be used to resolve problems posted by any liner equalities or inequalities. As result of the perfectly symmetric nature of the unit shell, it can be demonstrated that it requires a minimum number of steps to reach a solution while maintaining numerical stability when it is compared to existing best known algorithms offered by the Simplex or interior point methods. The author illustrates this innovative new technique to solve a simple linear program (LP) system with 7 variables and 14 constraints. Excel is used to carry out such new approach step by step to illustrate and verify its correctness and efficiency. A twist of this approach is that it is capable of solving all the linear systems (equalities, Eigen systems, and inequalities alike) over the unit shell non-linearly! The normalization of vectors on the unit shell also offers numerical stability for ill-conditioned linear systems that are very troublesome to solve with existing techniques in dealing with very large linear systems with millions of variables or constraints. Such an approach may have other applications beyond linear systems such as physics as an alternative to the string theory to resolve the problem of renormalization and operations research NPC mystery (?).

The author presents such a geometrical center search strategy for the maximum possible rank with the same number of linearly independent rows in L as a solution to $f = Lx \geq 0$. Such analysis shows that it is possible to solve LIS (linear inequalities system) with non-iterative and non-Gaussian alike algorithms. The author is putting this draft together to highlight this unit shell approach. This approach is coined as LIS-III or Unit Shell for Linear Systems or (US_LS). Note that LIS-II is a generalized Gaussian elimination (GGE) procedure solving system of linear inequalities. LIS-II is capable of determining the feasible interval of individual variable as a linear inequalities feasibility analyzer (LIFA). LIS-I is a set of three key algorithms that recursively reduce maximum infeasibility, sum of all infeasibility, and the number of constraints with the worst case infeasibility for any given linear system in feasibility form. US_LS has many advantages over existing interior point methods or the ellipsoid method as US_LS is symmetric and numerical stable for all the vectors are normalized to be unit vectors on the surface of a unit shell. It is a global, symmetric, stable, and one pass approach without the need of iterative application of some key algorithms.

2. Highlights

1. Any linear systems in feasibility form (LSF) [ref. 17]:

$$f = Lw = \begin{vmatrix} L_1 \\ L_2 \\ : \\ L_{m-1} \\ L_m \end{vmatrix} w \geq 0 \quad \text{where } L_1, L_2, ---, L_m \quad \text{are the row vectors}$$

2. Linear Systems on the Unit Shell after normalization:

$$\begin{vmatrix} 1/\sqrt{L_1 \bullet L_1} & 0 & 0 & 0 & 0 \\ 0 & 1/\sqrt{L_2 \bullet L_2} & 0 & 0 & 0 \\ : & 0 & : & 0 & : \\ 0 & 0 & 0 & 1/\sqrt{L_{m-1} \bullet L_{m-1}} & 0 \\ 0 & 0 & 0 & 0 & 1/\sqrt{L_m \bullet L_m} \end{vmatrix} Lw = Uw = \begin{vmatrix} u_1 \\ u_2 \\ : \\ u_{m-1} \\ u_m \end{vmatrix} w \geq 0$$

Note that each row, u_i in *U* is a point on the surface of an n-dimensional unit sphere (with a radius of 1.0), namely the unit shell. The solution normalized solution, $w / \sqrt{w \bullet w}$ is also a point on the same unit shell. Moreover, the maximum arc distance from this point to all the points in *U* is minimized by the definition of a minimum cover on the unit shell for all points in *U*.

3. A great circle arc connecting two points on a unit shell

Let *p* and *q* are two points on the surface of an n-dimensional unit shell, then $x_u = y / \sqrt{y \bullet y}$ with $y = tp + (1-t)q$ with $0 \leq t \leq 1$ is a path on the unit shell connecting points p and q as part of the great circle that passing both points p and q. In other words, we have the equation for all the points on the great circle arc connecting *p* and *q*: $x_u = (tp + (1-t)q) / \sqrt{1 - 2t(1-t)(1 - p \bullet q)}$

Note that $\sqrt{1-2t(1-t)(1-p\bullet q)}=\|tp+(1-t)q\|$ is simply the normalization factor to project the vector y onto the unit shell,

Consequently, the mid-point of $\overset{\frown}{pq}$ is $(p+q)/(2\sqrt{1-\frac{1}{2}(1-p\bullet q)})$ it is also the geometrical center of $\{p,q\}$ on the unit shell.

4. Equidistant Point (EDP) point of a set of points, $U=\{u_1,u_2,----,u_k\}$ on the surface of the unit shell is any point, *p,* on the unit shell such that $\overset{\frown}{pu_i}=d$ for all *i*.

5. A Cover and the Geometrical Center of points on the unit shell

The disc centered at *e* with a radius of *R* is a cover for all the points with $\overset{\frown}{eu}\le R$ i.e., $C(e,R)$ or simply $C=(e,R)$.

Given a set of points, $U=\{u_1,u_2,----,u_k\}$ on the surface of the unit shell, a point, *c*, is a geometric center, GC, of U if the maximum arc distance from *c* to all the points in U is minimized. In other words,

$$\max_{\forall i}\{\overset{\frown}{c,u_i}\}=\min_{\forall x}\{\max_{\forall i}\{\overset{\frown}{x,u_i}\}\}$$

Given a set of points, $U_k=\{u_1,u_2,----,u_k\}$ on the surface of the unit shell, let rank $U_k=k$

Let e_k be an EDP for $U_k=\{u_1,u_2,----,u_k\}$ on the surface of the unit shell, let $e_k=\sum_{i=1}^{k}c_iu_i$. We have $e_k\bullet u_i=\cos^{-1}(R)=r\ \forall\, u_i\in U_k$ by definition of EDP; we also have $u_i\bullet u_i=1$ for all *i*

Hence, we should have the solution for the coefficients $\{c_1,c_2,---,c_k\}$ from

$$\begin{vmatrix} 1 & u_1\bullet u_2 & u_1\bullet u_3 & : & u_1\bullet u_k \\ u_2\bullet u_1 & 1 & & & u_2\bullet u_k \\ u_3\bullet u_1 & & : & & \\ : & & & 1 & \\ u_k\bullet u_1 & u_3\bullet u_2 & & u_k\bullet u_{k-1} & 1 \end{vmatrix}\begin{vmatrix} c_1\\ c_2\\ :\\ c_k\end{vmatrix}=r\vec{1}\ \text{ where }\ \vec{1}=\begin{vmatrix}1\\1\\1\\:\\1\end{vmatrix}$$

If $u_j=\sum_{i=1}^{k}\alpha_iu_i$, from $\overset{\frown}{e_ku_j}=\sum_{i=1}^{k}\alpha_i\,\overset{\frown}{e_ku_i}=R\sum_{i=1}^{k}\alpha_i$

then $\sum_{i=1}^{k}\alpha_i<1$, =1, >1 if u_j is an interior, boundary, or exterior point of the cover, $c=(e_k,R)$.

Swapping u_j with u_i we have $u_i=\dfrac{1}{\alpha_i}(u_j-\sum_{m\ne i,m\ne j}^{k}\alpha_mu_m)$.

Note: let $U_2=\{u_1,u_2\}$ and $U_3=\{u_1,u_2,u_3\}$, if u_3 is outside the minimum disc cover of $\overset{\frown}{u_1u_2}$, the cover as a disc centered at an EDP, e_2 of U_2 (as the geometrical center of U_2, namely, $e_2=(u_1+u_2)/\sqrt{4-2(1-u_1\bullet u_2)}$ with a radius R_2 as $C=(e_2,R_2=\frac{1}{2}\overset{\frown}{u_1u_2})$ does not cover u_3, it may still be possible to find a disc cover $C=(e_3,R_3)$ centered at e_3 with a radius R_3 as an EDP for U_3.

If rank U_3=3, then the normalized orthogonal projection of u_3 onto U_2,

i.e.,

$x_2 = (I - U_2(U_2^T U_2)^{-1} U_2^T) u_3 / \| (I - U_2(U_2^T U_2)^{-1} U_2^T) u_3 \|)$ is an EDP ($\neq e_2$) of U_2 with radius $\pi / 2$.

The great circle that goes thru both e_2 and x_2 are all EDPs of U_2. Note that an EDP, e_3 for U_3 is the normalized $tx_2 + (1-t)e_2$ such that $(tx_2 + (1-t)e_2) \bullet u_3 = (tx_2 + (1-t)e_2) \bullet u_1 = (tx_2 + (1-t)e_2) \bullet u_2$.

By induction, the same argument shows that exterior points of a disc cover on the unit shell may be covered by another disc as a boundary or interior point of another EDP at a higher rank. The minimum cover is one centered at an EDP with least number of boundary points and hence, smallest radius with the smallest rank. In summary, by moving to a higher rank, one can increase the number of boundary points centered at distinct EDP or EDPs with a larger radius. How to find a second EDP with increased rank (outside the subspace on the unit shell)?

Two methods are investigated:

(a) Using Liner Subspace Projection Operation, $P_A x = A(A^T A)^{-1} A^T x$ as described above.

(b) Using natural basis and subspace ranking

(a): Assume that we have set of unit vectors $V = \{v_i\}_{i=1}^m \subseteq U = \{u_j\}_{j=1}^k$ with $m \le k$, and another unit vector, g_k, that is an EDP for V. The following steps are needed to locate another EDP ($\neq g_k$) for V.

1. Compute the maximum arc distance from g_k to any point in U as $\overset{\frown}{v_{m+1} g_k} = \max\{\overset{\frown}{u_j g_k}\}_{j=1}^k$ with $v_{m+1} \in U - V$.
2. Compute subspace projection of v_{m+1} onto V as $P_V(v_{m+1}) = V(V^T V)^{-1} V^T v_{m+1}$
3. Identify the orthogonal component of v_{m+1} to V as $V^{\perp} v_{m+1} = (I - V(V^T)^{-1} V^T) v_{m+1} = x \neq \vec{0}$
4. Normalize *x* to become an unit vector as $y = x / \| x \|$
5. Move point $h = (tg_k + (1-t)y) / \sqrt{1 - 2t(1-t)(1 - g_k \bullet y)}$ for any $1 \le i \le k$ along the great circle on the unit shell towards from *y* towards g_k such that $\overset{\frown}{hv_{m+1}} = \overset{\frown}{hv_i}$, i.e., we have

$$t = (g_k \bullet v_{m+1} - g_k \bullet v_i) / (g_k \bullet v_{m+1} - g_k \bullet v_i - y \bullet v_{m+1} + y \bullet v_i)$$
$$= g_k \bullet (v_{m+1} - v_i) / (v_{m+1} - v_i) \bullet (g_k - y)$$

for $1 \le i \le m$

Alternative, for $s = 1 - t$, $s = y \bullet (v_{m+1} - v_i) / (y - g_k) \bullet (v_{m+1} - v_i)$ for $1 \le i \le m$

(b): Hint:

Let $h_k = g_k + \sum_{i=1}^n a_i e_i$, where e_i is the unit vector with 1 at its *i*-th position and *0* otherwise.

Note that in the above equality, h_k, g_k and e_i are n-dimensional vectors while a_i is a scalar real number, in other words,

We have, $a_i e_i = \begin{vmatrix} 0 \\ \vdots \\ a_i \\ \vdots \\ 0 \end{vmatrix}$

Solve

$$h_k \bullet u_j = g_k \bullet u_j + \sum_{i=1}^n u_j(i) a_i = R + \sum_{i=1}^n u_{ji} a_i = R + \Delta R \text{ for } j = 1,2,---.k, i = 1,2,---,n.$$

dimensional vector with its *i*-th component, $u_j(i) = U_{ji}$

Namely, we treat u_j as a row vector $u_j = (U_{j1}, U_{j2}, ----, U_{jn})$.

Note that $\{e_i\}_{i=1}^n$ has full rank *n* ($k \le n$), if *k<n*, we can find at least one $\vec{a}$

$\vec{a} = \sum_{i=1}^{n} a_i e_i$ from $U_k^T = \{u_1, u_2, ----, u_k\}$, such that we have $\vec{a} \bullet u_j = \Delta R \forall 1 \le j \le k$

For h_k to be an EDP distinct from g_k for $U = U_k$, we must have

$\sum_{i=1}^{n} a_i u_{ji} = \Delta R$ for *j=1,2,.....,k*. Namely, $|u_{ji}|\vec{a} = \Delta R \vec{1}$,

i.e.,

$$\begin{vmatrix} U_{11} & U_{12} & : & : & U_{1n} \\ U_{21} & U_{22} & : & : & U_{2n} \\ : & : & : & : & : \\ : & : & : & : & : \\ U_{k1} & U_{k2} & : & : & U_{kn} \end{vmatrix} \begin{vmatrix} a_1 \\ a_2 \\ : \\ : \\ a_n \end{vmatrix} = \Delta R \begin{vmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{vmatrix}$$

In other words, any unit vector as normalized h_k with $a = \sum_{i=1}^{n} a_i e_i$ obtained from

$U_{kk} \overrightarrow{a_k} + U_{k,n-k} \overrightarrow{a_{n-k}} = \Delta R \vec{1}$ is an EDP ($\ne g_k$) for U_k

6. Four cases of relationship of u_{k+1} with U_k.

CASE I. Rank U_{k+1} = Rank $U_k = m \le k$ and minimum $C(g_k, r_k)$ covers U_{k+1}.

$\Rightarrow$ we have $C(g_k, r_k)$ also a minimum cover of U_{k+1}.

CASE II. Rank U_{k+1} = Rank $U_k = m \le k$ and minimum $C(g_k, r_k)$ does **not** cover u_{k+1}.

CASE III. Rank $U_k = m$ < Rank $U_{k+1} = m+1$ and minimum $C(g_k, r_k)$ covers U_{k+1}.

$\Rightarrow$ It is possible to locate another EDP, $g_{k+1} \ne g_k$, for U_{k+1} such that $C(g_{k+1}, r_{k+1})$ covers U_{k+1} with $r_k \le r_{k+1}$. g_{k+1} is obtained from the great circle arc connecting $h_{k+1} \overset{\frown}{} g_k$ where h_{k+1} is the normalized $(I - V_m (V_m^T)^{-1} V_m^T) u_{k+1}$ for any full ranked $V_m \subseteq U_k$.

CASE VI. Rank $U_k = m$ < Rank $U_{k+1} = m+1$ and minimum $C(g_k, r_k)$ does **not** cover U_{k+1}.

Analysis: For CASE II,

Let $u_{k+1} = \sum_{i=1}^{m} \beta_i u_i$, we have $u_{k+1} \overset{\frown}{} g_k = (\sum_{i=1}^{m} \beta_i) u_i \overset{\frown}{} g_k = r_k \sum_{i=1}^{m} \beta_i > r_k$; hence $\sum_{i=1}^{m} \beta_i > 1$.

Move a point p_1, from u_{k+1} towards g_k along the great circle arc $u_{k+1} \overset{\frown}{} g_k$, until we have

$p_1 \overset{\frown}{} u_{k+1} = p_1 \overset{\frown}{} u_i$ for at least one $u_j \in U_{k+1}$ - $\{u_{k+1}\}$. Let $v_1 = u_{k+1}$ and $v_2 = u_i$,

Let h_2 as the midpoint of $v_1 \overset{\frown}{} v_2$, select v_3 from $U_{k+1} - V_2$ such that $v_3 \overset{\frown}{} h_2 = \max_{\forall u_x \in U_{k+1} - V_2} \{u_x \overset{\frown}{} h_2\}$.

For *i*=2,3 ----, m we can select linearly independent subset

$$V_i = \{v_1, v_2, ----, v_i\} \subseteq U_{k+1}$$

Such that we can find at least one $u = \vec{a} / \sqrt{\vec{a} \bullet \vec{a}} \neq h_{i-1}$ such that $|v_{ji}|\vec{a} = 0$, i.e.,

$$\begin{vmatrix} v_{11} & v_{12} & : & : & v_{1n} \\ v_{21} & v_{22} & : & : & v_{2n} \\ : & : & : & : & : \\ : & : & : & : & : \\ v_{i1} & v_{i2} & : & : & v_{in} \end{vmatrix} \begin{vmatrix} a_1 \\ a_2 \\ : \\ : \\ a_n \end{vmatrix} = 0$$

Let V_{ii} be any full ranked *I* by *I* sub-matrix of $|v_{ji}|$, we have $V_{ii}\vec{a_i} + V_{-i,n-i}\overrightarrow{a_{n-i}} = 0$

Solve for $\vec{a_i} = V_{ii}^{-1} V_{-i,n-i} \overrightarrow{a_{n-i}}$ Normalized $\vec{a}$, $u = \vec{a} / \sqrt{\vec{a} \bullet \vec{a}} \neq h_{i-1}$ is an EDP of

$V_i = \{v_1, v_2, ----, v_i\}$ and the great circle arc $\overset{\frown}{g_{i-1} u}$ connecting g_i and u are all EDP for V_{i-1}.

v_{i+1} is selected from $U_{k+1} - V_i$ such that $\overset{\frown}{v_{i+1} h_i} = \max_{\forall u_x \in U_{k+1} - V_i} \{\overset{\frown}{u_x h_i}\}$. When $i = m$, we have a new cover

$C(h_m, \overset{\frown}{v_1 h_m})$ for U_{k+1}. For CASE IV, It is possible to locate another EDP $g_{k+1} \neq g_k$ for U_{k+1} such that a hemispherical cover for U_{k+1} is obtained. This is because $C(x_u, \pi/2)$ is a hemispherical cover of U_{k+1} where

$$x_u = (I - V_m (V_m^T V_m)^{-1} V_m^T) u_{k+1} / \| (I - V_m (V_m^T V_m)^{-1} V_m^T) u_{k+1} \|.$$

7. Arc connecting two EPDs on Unit Shell

If points *p* and *q* are distinct equidistant points (EDP) for $U = \{u_1, u_2, ----, u_k\}$ on the surface of the unit shell, then, all the points on the great circle connecting *p* and *q* are all EDPs for U.

Proof:

Let *p* and *q* be two distinct equidistant points for $U = \{u_1, u_2, ----, u_k\}$, we have

$R_p = \overset{\frown}{u_1 p} = \overset{\frown}{u_2 p} = ... = \overset{\frown}{u_k p}$ and $R_q = \overset{\frown}{u_1 q} = \overset{\frown}{u_2 q} = ... = \overset{\frown}{u_k q}$.

Let $r_t = (tp + (1-t)q) / \| tp + (1-t)q \| = (tp + (1-t)q) / \sqrt{1 - 2t(1-t)(1 - p \bullet q)}$ be a normalized unit vector on the great circle connecting *p* and *q*. Then, it is clear that we have

$\overset{\frown}{u_i r_t} = (tR_p + (1-t)R_q) / \sqrt{1 - 2t(1-t)(1 - p \bullet q)} = R_t / \sqrt{1 - 2t(1-t)(1 - p \bullet q)} = R_{r_t}$ $\forall t$ and $\forall i$.

Where $R_t = tR_p + (1-t)R_q$ and $R_{r_t} = R_t / \sqrt{1 - 2t(1-t)(1 - p \bullet q)}$.

8. The US-LS algorithm, a unit shell algorithm to minimize the maximum infeasibility of a linear system in feasibility form, Excel is used to carry out this proposed algorithm step by step to solve a simple LP in LSF form with 7 variables and 14 constraints:

Step 1. Problem formulation: Let the LSF be normalized such that every row vector is an unit vector with it end point on the unit shell, we have the normalized solution of the LP also an unit vector on the unit shell and the problem becomes given an matrix with k unit vectors as

$$U = \begin{vmatrix} u_1 \\ u_2 \\ : \\ u_{k-1} \\ u_k \end{vmatrix}$$, locate a unit vector, $\vec{x}$ such that we have $U\vec{x} \geq 0$.

Step 2. Select a pair of distinct points, $\{v_1, v_2\} = V_2$ from rows in U such that $\overset{\frown}{v_1 v_2} = \max\{\overset{\frown}{u_i u_j} \mid u_i \in U \,\&\, u_j \in U, \&\, \overset{\frown}{u_i u_j} < \pi\}$ compute the midpoint, g_2, of $\overset{\frown}{v_1 v_2}$ as it is described in section 3.0. Note that g_2 is both an EDP (equal distanced point) and the geometrical center of V_2.

Step 3. Compute the subspace projection of V_2 as $P_{V_2} = V_2(V_2^T V_2)^{-1} V_2^T$.

Step 4. Find the first unit vector, v_3 from the unit vectors of $U - V_2$ such that the arc distance $\overset{\frown}{v_3 g_2} = \max_{\forall u_i \in U - V_2}\{\overset{\frown}{u_i g_2}\}$ and that the normalized $P_{V_2} v_3$ is not equal to v_3. In other words, let $v_3 \in U - V_2$ such that $\overset{\frown}{v_3 g_2} = \max_{\forall u_i \in U - V_2}\{\overset{\frown}{u_i g_2}\}$ and $y = (I - V_2(V_2^T V_2)^{-1})v_3 \neq \vec{0}$. Note that we have rank (V_3) =3 > rank (V_2) = 2 where $V_3 = \{v_1, v_2, v_3\}$

Step 5. Normalize vector y to become an unit vector $y_u = y / \| y \|$, then y_u is also an unit vector with its end point on the surface of the unit shell. Note that both g_2 and y_u are EDPs for points in V_2. Consequently, the great circle $\overset{\frown}{g_2 y_u}$ are all EDP for V_2. On this great circle, one can locate a unique point, h_3 as the shortest EPD for V_3 by the following two equations:

a. $p_t = (tg_2 + (1-t)y_u) / \sqrt{(1 - 2t(1-t)(1 - g_2 \bullet y_u)} = (tg_2 + (1-t)y_u) / \sqrt{1 - 2t(1-t)}$ as $g_2 \bullet y_u = 0$

$p_t \bullet v_3 = p_t \bullet v_1$ (note that we will always have $p_t \bullet v_1 = p_t \bullet v_2$

b. as p_t is also an EDP for V_2.

Solve for *t,* from a. and b., we have, note that we have $\overset{\frown}{p_t v_3} = \overset{\frown}{p_t v_1}$, hence,

$t = g_2 \bullet (v_3 - v_1) / (g_2 - y_u) \bullet (v_3 - v_1)$.

Substitute t into equation a., we have the shortest EDP or the geometrical center, $g_3 = p_t$ for V_3.

Steps 2 to 5 can be repeated with increasing subscription for V_i and v_{i+1} for $i = 3,4----,m$ where *m*=rank (U).

Step 6. Once the highest rank for, *m*, for U has reached, we have a geometrical center, g_m of V_m such that g_m is an EDP for V_m. For all rows (points) of $U - V_m$, compute $\overset{\frown}{g_m u_j}$ for maximum arc distance $\max_{\forall u_j \in U - V_m} \overset{\frown}{g_k u_j} = d_m$. *If* $d_m \leq \overset{\frown}{v_1 g_m}$, then we have identified a minimum cover $C(g_m, \overset{\frown}{g_m v_1})$.

If $\overset{\frown}{v_1 g_m} < d_m$, we have $V'_{m-1} = V_m - \{v_i\}$ has rank *m-1*, the orthogonal projection $(I - V'_{m-1}((V'_{m-1})^T V'_{m-1})^{-1}(V'_{m-1})^T)v_i$ normalized is an EDP h'_{m-1} for V'_{m-1} and has its arc distance to all points in V'_{m-1} as $\overset{\frown}{h'_{m-1} v_j} = \pi/2 \ \forall v_j \in V'_{m-1}$. In other words, we have $h'_{m-1} \bullet v_j = 0 \ \forall v_j \in V'_{m-1}$.

If a hemispherical cover, $C(h'_{m-1}, \pi/2)$ for U exists, it must be one of these *m* possible h'_{m-1}.

9. An Example: Consider the following sample LP pairs as primal and its dual [Refs 1 to 7].

$$Max \ -3y_2 - 3y_4$$

$$\text{subject to} \quad \begin{vmatrix} 1 & -2 & 0 & -1 \\ 0 & -1 & 1 & -2 \end{vmatrix} * \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \leq \begin{bmatrix} -4 \\ -5 \end{bmatrix}, \quad \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \geq 0$$

$$Min \quad -4x_1 - 5x_2$$

$$\text{subject to} \quad \begin{vmatrix} 1 & 0 \\ -2 & -1 \\ 0 & 1 \\ -1 & -2 \end{vmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ -3 \\ 0 \\ -3 \end{bmatrix}, \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq 0$$

In self-dual formulation as pure linear inequalities, we have:

$$F = L * w = \begin{vmatrix} 0 & 0 & -1 & 2 & 0 & 1 & -4 \\ 0 & 0 & 0 & 1 & -1 & 2 & -5 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & 0 & 0 & 0 & 0 & 3 \\ 4 & 5 & 0 & -3 & 0 & -3 & 0 \\ -4 & -5 & 0 & 3 & 0 & 3 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ 1 \end{bmatrix} \geq 0 \text{ -------------------- (1)}$$

Normalize (1) to have all row vectors as points, $U = \{u_1, u_2, -----, u_{14}\}$ on the 7-dimensional unit shell, we have

u1= (	0	0	-0.213201	0.426401	0	0.213200716	-0.8528029)
u2= (	0	0	0	0.179605	-0.179605	0.359210604	-0.8980265)
u3= (	1	0	0	0	0	0	0)
u4= (	-0.534522484	-0.267261	0	0	0	0	0.80178373)
u5= (	0	1	0	0	0	0	0)
u6= (	-0.267261242	-0.534522	0	0	0	0	0.80178373)
u7= (	0.520755644	0.650945	0	-0.390567	0	-0.390566733	0)
u8= (	-0.520755644	-0.650945	0	0.390567	0	0.390566733	0)
u11=(	0	0	1	0	0	0	0)
u12=(	0	0	0	1	0	0	0)
u13=(	0	0	0	0	1	0	0)
u14=(	0	0	0	0	0	1	0)

Let $V_1 = \{u_1\}$, we have $g_1 = u_1$, $R_1 = 0$, from $U - V_1$, we determine v_2 such that $\overset{\frown}{g_1 v_2} = \max_{\forall u_j \in U - V_1} \{\overset{\frown}{g_1 u_j}\}$, we have $v_2 = u_6$, i.e.,

$V_2 = \{u_1, u_6\}$, the geometrical center of V_2 is the midpoint of $\overset{\frown}{u_1 u_6}$, we have

g2=(0.33605856 -0.672117115 -0.268082 0.536163977 0 0.268081988 0.06415)

Hence, $R_2 = \cos^{-1}(v_1 \bullet g_2) = 0.36982899\pi$.

v_3 is determined from $U - V_2$ such that $\overset{\frown}{g_2 v_3} = \max_{\forall u_j \in U - V_2}\{\overset{\frown}{g_2 u_j}\}$, we have $v_3 = u_7$, i.e.,

$V_3 = \{u_1, u_6, v_7\}$, Note that g_2 is both EDP and the geometrical center of V_2.

To locate the geometrical center which is also an EDP for V_3, we can compute the orthogonal projection h_3 of v_3 of the subspace V_2, i.e., h_3 is the normalized unit vector of $(I - P_{V_2})v_3$ where

$P_{V_2} = V_2(V_2^T V_2)^{-1} V_2^T$. Note that this normalized unit vector

$h_3 = (V_2(V_2^T V_2)^{-1} V_2^T v_3 / \| (V_2(V_2^T V_2)^{-1} V_2^T v_3\|$ is orthogonal to points in V_2 and has an arc distance of $\pi/2$ to all the points in V_2. In other words, we have identified another EDP for V_2 that is distinct from g_2.

The great circle arc $\overset{\frown}{g_2 h_3}$ connecting g_2 and h_3 are all EDP for V_2 and the geometrical center for V_3 is also a point, g_3, on this great circle arc as an EDP for V_3 with the minimum radius. There is only one such point as the normalized $tg_2 + (1-t)h_3$ with *t* computed as shown in Step 5.

In this example, we have:

h_3 = (0.539654327 -0.026983 -0.661077 0.215862 0 -0.445215 0.161896298)

Using Excel, we have *t*=0.210480823 and

g_3 = (0.434874728 0.199207 -0.707824 0.346691 0 -0.361133 0.139907279)

Hence, we have $R_3 = \cos^{-1}(v_1 \bullet g_3) = 0.4673379\pi$.

Continue this process for V_4, V_5, V_6, and V_7 until we reach the maximum possible rank of 7 as:

$V_4 = \{u_1, u_6, u_7, u_{11}\}$ with

h_4 = (0.410337502 -0.237564 0.691095 0.453531 0 -0.302354 -0.021596711)

Using Excel, we have *t* =0.460315637 and

g_4 = (0.594408948 -0.310022 0.066472 0.570046 0 -0.464397 0.074360407)

We have $R_4 = \cos^{-1}(v_1 \bullet g_4) = 0.478825698\pi$

$V_5 = \{u_1, u_6, u_7, u_{11}, u_{14}\}$ with

h_4 = (0.410337502 -0.237564 0.691095 0.453531 0 -0.302354 -0.021596711)

Using Excel, we have *t*=0.598264383 and

g_5 = (0.801980629 -0.213305 0.055185 0.51732 0 0.055185 0.193950554)

We have $R_5 = \cos^{-1}(v_1 \bullet g_5) = 0.482425277\pi$

$V_6 = \{u_1, u_6, u_7, u_{11}, u_{14}, u_4\}$ with

h_6 = (-0.316227766 0.632455532 0 0.632456 0 0 0.316227766)

Using Excel, we have *t*= 0.483043855 and

g_6 = (0.316483009 0.316483009 0.037676 0.815307 0 0.037676 0.363473764)

We have $R_6 = \cos^{-1}(v_1 \bullet g_6) = 0.488004383\pi$

From g_6, we identified the unit vector orthogonal to V_6 from u_2 as h_7

$h_7 = ($ 0 0 0 0 -1 0 0 $)$

Using Excel, we have t= 0.468058988 and $V_7 = \{u_1, u_6, u_7, u_{11}, u_{14}, u_4, u_2\}$

Figure 1 illustrates the arc distance of $v_i \overset{\frown}{p}(t)$ and the arc distance of $u_2 \overset{\frown}{p}(t)$ for $0 \le t \le 1$ where $p(t) = p_t = (tg_6 + (1-t)h_7)/\sqrt{1-2t(1-t)}$. As { p_t } for different value of t are all EDPs for V_6 , we have an EDP for $V_7 = V_6 + \{u_2\}$ at t= 0.468058988.

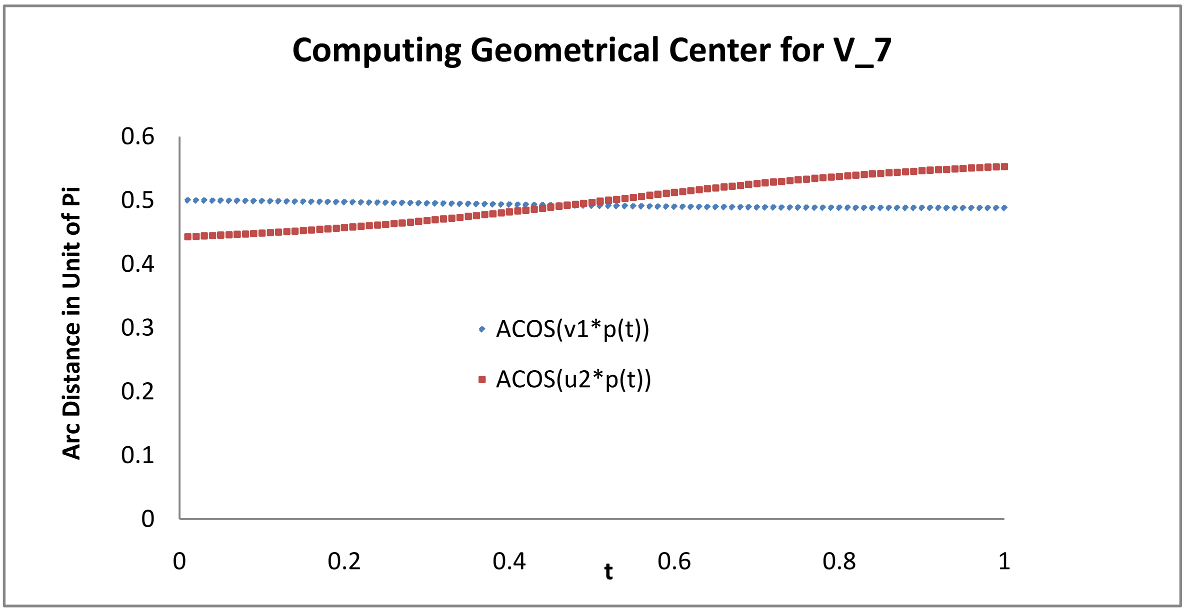


Figure 1. Computing EDP for V_7 as g_7

Consequently, we have

$g_7 = ($ 0.209065143 0.209065143 0.024889 0.538582 -0.750748 0.024889 0.240106712 $)$

With $R_7 = \cos^{-1}(v_1 \bullet g_7) = 0.492076887\pi$.

We have reached the geometrical center as an EDP for a full ranked V_7 as g_7. The only possible hemispherical cover for V_7 is centered at the normalized $(I - P_{V_7 - \{v_i\}})v_i$ and $Uw \ge 0$ has solution if and only if such as hemispherical cover for U exists.

It is trivial to check that the unit vector $x_u = (I - P_{V_7 - \{u_{14}\}})u_{14} / \|(I - P_{V_7 - \{u_{14}\}})u_{14}\|$ is the solution of $Uw \ge 0$ for the original LP. In other words, we have identified the hemispherical cover $C = (x_u, \pi/2)$ for all the 14 points in U with $x_u = w/\|w\|$.

With Excel, we have:

$x_u = ($ 0.353553391 0.353553391 0 0.353553391 0 0.707106781 0.353553391$)$

To verify that x_u is indeed the solution of the original LP in (1), we have $x_u = w/\|w\|$ such that $w = \|w\| x_u = 2.828427125 * x_u = (x_1, x_2, y_1, y_2, y_3, y_4, 1)$=(1,1,0,1,0,2,1)

Hence the solution to the LP in (1) is:

$$x_1 = 1, x_2 = 1, y_1 = 0, y_2 = 1, y_3 = 0, y_4 = 2$$

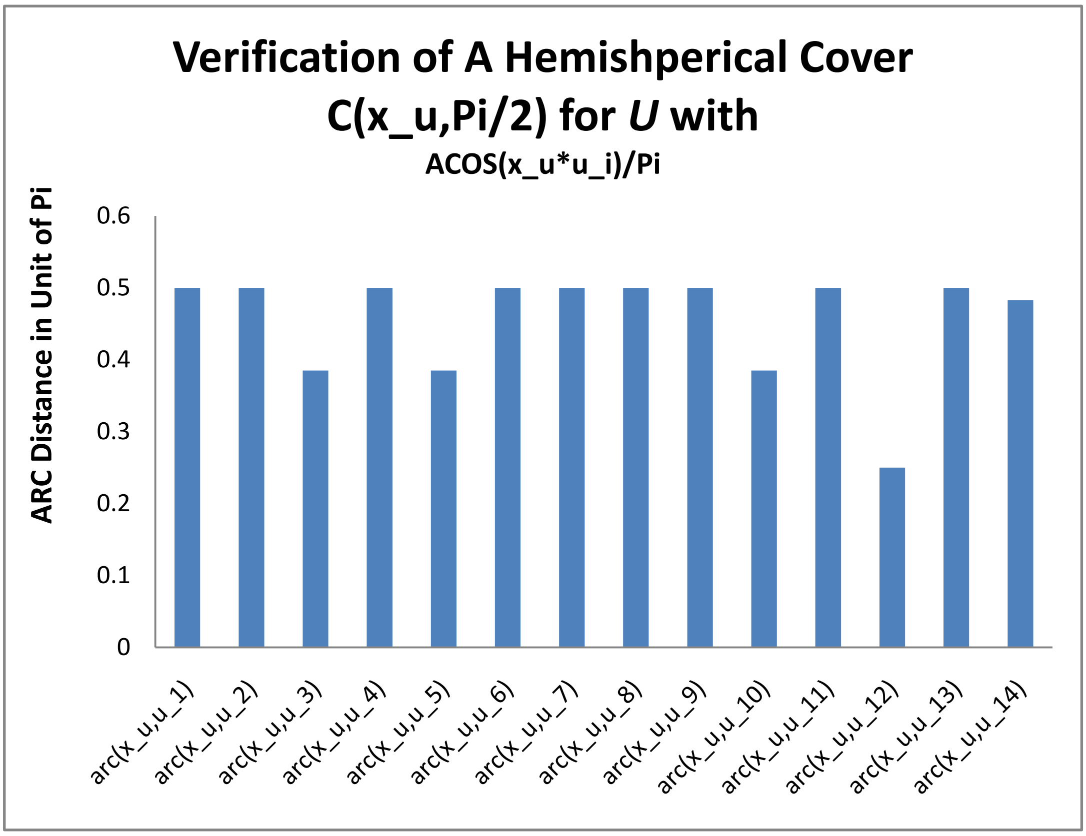


Figure 2. Solution of LP as the Center of a Hemispherical Cover, $C(x_u, \pi/2)$ for $f = Uw \geq 0$

Figure 2 illustrates the arc distance from all points on U to the solution point, x_u, such an EDP for V_7 as the geometrical center of $U = \{u_i\}_{i=1}^{14}$. Note that such a solution point is a point with a minimum arc distance to all the constraint points. In other words, it has the minimum arc distance to all the constraint points. It is the best solution to any given set of linear inequalities that can be humanly resolved.

The solution obtained from the unit shell is identical to that obtained by either the simplex method, interior point method, LIS-I, or the GGE (LIS-II) approaches. Over the past few years, the author has developed three different methods for solving any linear systems and they are as efficient as the text book simplex and interior point methods with considerable simplification and numerical stability.

This paper details the theory and steps of such a generic algorithm to solve both linear equalities and inequalities on the Unit Shell with a minimum number of steps involving only the unit vectors and linear subspace projection operation $I - (V(V^T V)^{-1})V^T$. A simple LP solved with this new technique is demonstrated step by step with Excel tabulation and matrix operations. The support is attached for verification and validation of this new approach.

10. Solving Differential Variation Inequalities (DVI) on the Unit-Shell

We generalize the unit-shell algorithm over linear space to a normed Banach space with an inner-product operator <u,v> with norm $\|u\| = \sqrt{<u, u>}$ for solving variational inequalities (VI) or differential variational inequalities (DVI) as follows [refs 14 and 15]:

Let $\phi = \{Bx \leq b\}$, where B is an $m \times n$ matrix as a nonempty convex compact polyhedron in R^n

Let F be a continuously differentiable function from ϕ into R^n with Jacobian F'.

The variational inequality problem (VIP) associated with F and ϕ is to locate a solution x^* in ϕ satisfying the variational inequality (VI): $(x^* - x)^T F(x^*) \leq 0 \forall\, x$ in ϕ. Note that in E^n, we have $\|x\| = \sqrt{<x, x>} = \sqrt{x^T x}$

Let the gap function associated with a VIP be defined for x in ϕ as:

$$g(x) = \max_{y \in \phi} (x - y)^T F(x)$$

While the dual gap function associated with a VIP is defined as $\bar{\bar{g}}(x) = \max_{y \in \phi} (x - y)^T F(y)$

Using Newton's first order Taylor linear approximation around a point x_k in ϕ, a linearized VIP as LVIP can be computed iteratively for $k = 0,1,2, \ldots,$ as:

$$(x_{k+1} - y)^T (F(x_k) + F'(x_k)(x - x_k)) \le 0 \forall y \in \phi.$$

Consider the following nonconvex, nonlinear constrained mathematical program:

$\min_{x,\lambda} h(x,\lambda) \overset{def}{=} \lambda^T(b - Bx) = x^T F(x) + b^T \lambda$ subject to $F(x) + B^T\lambda=0,\ Bx \le b, 0 \le \lambda.$

Note that optimality occurs at:

$$\min_{By \le b} y^T F(x) = \max_{\substack{B^T\mu = F(x) \\ \mu \le 0}} b^T \mu$$

subject to: $By \le b, B^T \mu = F(x), \& \ \mu \le 0$

Consequently, we have the following homogeneous linear feasible system of inequalities $f = Lw \ge 0$

$$\text{Where } L = \begin{bmatrix} B & 0 & b \\ 0 & B^T & -F(x) \\ 0 & -B^T & F(x) \\ 0 & -I & 0 \end{bmatrix} \text{ and } w = \begin{bmatrix} y \\ \mu \\ 1 \end{bmatrix}$$

Note that $f = Lw \ge 0$ can be solved effectively over the unit shell using only linear projection and the concept of equi-distanced points to selected set of normalized unit vectors derived from row vectors of L as described and demonstrated in this draft paper from Section 1.0 to Section 8.0.

11. Other Examples of Linear Programs Solved by the Unit Shell Approach

First, Any non-homogeneous linear system with equalities, inequalities, or mixed as constraints may be converted to homogeneous linear system by increased its number of columns as variables by 1 shifting the right hand side vector to the left hand side so that the right hand side is always a vector with only zero coefficients. The row vectors of such a homogeneous linear system can always be normalized into vector of unit length as unit vector on the surface of a n-dimensional unit sphere as the unit shell. The normalized solution of such a homogeneous linear system is also a unit vector on the same unit-shell. In this section, the author provides both theoretical analysis and examples for four possible cases of solving homogeneous linear systems on the unit shell, namely, with unique solution, no solution, infinite solutions, and solutions that are unbounded. Both unbounded and infeasible LPs are examined and analyzed to provide necessary and sufficient conditions for linear systems with null or infinite many solutions such that the solvability of any system of linear equalities or inequalities are resolved with a computational complexity that is compatible to that of the traditional Gaussian Elimination for system of linear equalities, namely, $O[kmn]$ where k is the rank of the linear system (equalities and inequalities included) with m constraints and n variables.

11.1. Necessary, Sufficient Conditions, and Examples for the Four Cases of Linear Systems with Unique Solution, No Solution, Infinite Many Solution, and Unbounded Linear Solution

Example in Section 8 clearly illustrated the fact that when the rank of the homogenous linear inequalities equals to the number of unknowns plus 1, and all the unit vectors associated with the homogenous linear inequalities are reachable (or covered) by the hemispherical cover of the unit shell centered at:

$$x_u = w / \| w \| \text{ such that } x = (1,1,0,1,0,2,1).$$

If all the unit vectors associated with a given homogenous linear system can not be covered by such a hemispherical cover, it does not have a solution and the conflicting inequalities can be easily identified. On the other hand, if the rank of hemispherical cover is less than the number of unknowns, we will have an infinite number of EDPs for all the constraint inequalities, hence, it will have infinite many solutions or as unbounded cases.

3. Conclusions and Future Work

In conclusion, the author presents an innovative approach that relates algebraic relation in vectors to geometrical relation as arc distance on the unit shell, i.e., the surface of a n-dimensional hyper sphere.

The solution or solutions to any given linear system (both equalities and inequalities included) is shown to be the geometrical center or centers that are points on the unit shell with its maximum arc distance to all the constraints point minimized. This technique is illustrated and verified with an example of solving a linear program (LP) with 7 variables and 14 constraints. Only the dot product and linear subspace projection operation are used to locate the geometrical center or centers with a minimum number of steps. As dot product of two unit vectors is a direct measurement of correlation between two sets of numbers, the geometrical center or centers do reveal maximum correlation with minimum discrepancy. When we can compute efficiently the geometrical centers of a set of constraint points, it is applicable to a very wide range of scientific and engineering applications in aviation, finance, economics, operations research (OR), data mining, signal processing, and pattern recognition … etc. Additional benefit of dealing with only unit vectors is its numerical stability free from ill-conditioned operations that other approaches encountered very frequently. A future plan is to implement the unit shell algorithm in java as a generic optimization tool for general purpose scientific and engineering use.

In addition, the author is working on recursive projection and distance functions for unit vectors onto its k-dimensional subspaces and its direct application in solving homogeneous linear inequalities. This approach does provide effective and numerical stable algorithms that can handle very large system of linear inequalities with a computational complexity compatible to the classic Gaussian for solving system of linear equalities. This approach is extremely handy in dealing with linear inequalities with millions of unknowns and/or constraints. A drafted paper is currently under peer review by both mathematician and engineers.

Figure 3 illustrates the possibility of using recursively projection and distance functions in trigonometry as ***Sin*** and ***Cos*** functions to replace the classic linear space projector, $P_A x = A(A^T A)^{-1} A^T x$, and its orthogonal operators: $O_A x = (I - A(A^T A)^{-1} A^T) x$.

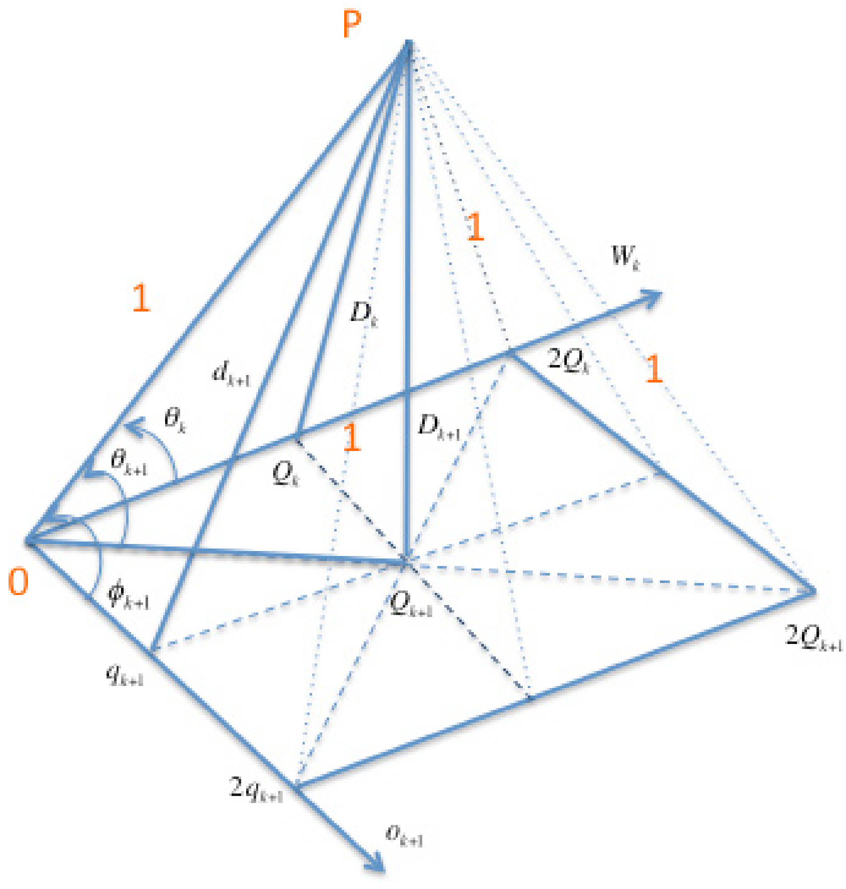
In Figure 3, W_k is a linear subspace with dimension and rank = k and an orthogonal basis.

Note that the computational complexity of locating such a geometry center for the set of normalized homogenous linear inequalities can be shown to be compatible to that of the classic Gaussian Elimination solving systems of linear equalities.

Support from NSF funding or private foundations source in innovative mathematical computation for further clarification and elaboration will be pursued.

ACKNOWLEDGEMENTS

The author acknowledges valuable suggestions, comments, and initial peer review of the concept of the Unit Shell by his colleagues, in particular, David Hammrick, Dr. Leone C. Monticone, and Dr. William P. Niedringhaus during past 10 years at the MITRE Corporation. Latest discussion and comments from Oliver C. Wang and Brian M. Jones have contributed to the clarity of the role of linear projection and orthogonal distance functions relevant to solving linear inequalities. Peer reviews and editorial enhancements from the referees and editors at Scientific & Academic Publishing (SAP), USA is also essential for the correctness and readability of this paper.



$$|0Q_k| = \cos\theta_k,\ |0Q_{k+1}| = \cos\theta_{k+1},\ |0q_{k+1}| = \cos\phi_{k+1}$$

$$|PQ_k| = D_k = \sin\theta_k,\ |PQ_{k+1}| = D_{k+1} = \sin\theta_{k+1},\ |Pq_{k+1}| = d_{k+1} = \sin\phi_{k+1}$$

New Trigonometry Identities for E^n

$$\cos^2\theta_k + \sin^2\theta_k = 1,$$

$$\cos^2\theta_{k+1} = \cos^2\theta_k + \cos^2\phi_{k+1},\quad \sin^2\theta_{k+1} = \sin^2\theta_k - \cos^2\phi_{k+1}$$

$$\cos^2\theta_n = \cos^2\theta_k + \cos^2\theta_{n-k},\quad \sin^2\theta_k - \cos^2\theta_{n-k} = \sin^2\theta_{n-k} - \cos^2\theta_k$$

Figure 3. Recursive Linear Subspace Projection and Distance Functions as Trigonometry Identities

REFERENCES

[1] Strang, Gilbert, "*Introduction to Applied Mathematics*", John Wiley & Sons Inc., New York, 1979.

[2] Strang, Gilbert, *"Karmarkar's algorithm and its place in applied mathematics",* The Msthematical Intelligencer 9(2): pp. 4-10, New York: Springer, 1987.

[3] Dantzig, G. G. *"Maximization of a llinear function of variables subject to linear inequalities"*, 1947, Published pp. 339-347, in T.C. Koopmans (ed.): Activity Analysis of Production and Allocation, Wiley & Chapman-Hall, New York-Lodon, 1951.

[4] Dantzig, G. B. "*Linear Programming and Extensions"*. Princeton, NJ: Princeton University Press, 1963.

[5] Fukuda, Komei and Terlaky, Tamas, "*Crisis-cross methods: A fresh view on pivot algorithms"*, Mathematical Programming: Series B, No. 79, Papers from 16th International Symposium on Mathematical Programming, Lausanne, 1997.

[6] Khachiyan, L. G., *"Polynomial algorithms in linear programming",* U.S.S.R., Computational Mathematical and Mathematical Physics 20 (1980) pp. 53-72.

[7] Karmarkar, N., "*A New Polynomial Time Algorithm for Linear Programming",* AT&T Bell Laboratories, Murray Hill, New Jersey, September, 1984.

[8] Gondzio, Jackek and Terlaky, Tamas, *"A computational view of interior point method"*, Advances in linear and integer programming, Oxford Lecture Series in Mathematics and its Applications, 4, New York, Osford University Press. pp. 103-144, MR1438311, 1996.

[9] Nocedal, Jorge and Wright, Stephen J,: " *Numerical Optimization*", Springer Science+Business Media, Inc., 1999.

[10] Michael. R. Garey and David. S. Johnson, *COMPUTERS AND INTRACTABILITY, A Guide to the Theory of NP-Completeness*, Bell Laboratories, Murray Hill, New Jersey, 1979.

[11] Nemirovsky, A. and Yudin, N. "*Interior-Point Polynomial Methods in Convex Programming",* Philadelphia, PA: SIAM, 1994.

[12] Alexander Schrijver, "*Theory of Linear and Integer Programming", Department of Econometrics"*, Tilburg University, A Wiley-Interscience Publication, New York, 1979

[13] Niedringhaus, W., "*Stream Option Manager (SOM): Automated Integration of Aircraft Separation, Merging, Stream Management, and Other Air Traffic Control Problems*", *IEEE Transactions Systems. Man & Cybernetics*, Vol. 25 No. 9, Sept. 1995.

[14] Niedringhaus, W., "*Maneuver Option Manager (MOM): Automated Simplification of Complex Air Traffic Control Problems*", *IEEE Transactions Systems. Man & Cybernetics*, May 1992.

[15] Marcotte, Patrice, **"***A New Algorithm for Solving Variational Inequalties with Application to the Traffic Assignment Problem"***,** Centre de Recherche sur les Transports, University de Montreal, Canada, Mathematical Programming 33 (1985) pp. 339-351, North-Holland

[16] Sun, Min, "*A New Alternating Direction Method for Co-corecive Variational* Inequality *Problems with Linear Equality and Inequality Constraints"*, pp. 161-176, Advanced Modeling and Optimization, Vol. 12, Number 2, 2010

[17] Wang, Paul T. R., *"Solving Linear Programming Problems in Self-dual Form with the Principle of Minmax*", MITRE MP-89W00023, The MITRE Corporation, July, 1989.

[18] Wang, Paul T. R., Niedringhaus, William P., and McMahon, Matthew T., *"A Generic Linear Inequalities Solver (LIS) with an Application for Automated Air Traffic Control". America Journal of Computational and Applied Mathematics,* pp 195-206, Volume 3, Number 4, August 2013.

[19] EE236-A, Lecture 15," *Self-dual formulations*", University of California, Department of Electrical Engineering, 2007-08.

[20] Self-Dual Form: http://www.ee.ucla.edu/ee236a/lectures/hsd.pdf

[21] ILOG, "*Introduction to ILOG CPLEX*", 2007, http://www.ilog.com/products/optimization/qa.cfm?presentation=3

[22] ILOG, "*CPLEX Barrier Optimizer*", 2008, http://www.ilog.com/products/cplex/product/barrier.cfm

[23] Steven Skiena, "*LP_SOLVE: Linear Programming Code*", Stony Brook University, Dept. of Computer Science", 2008 http://www.cs.sunysb.edu/~algorith/implement/lpsolve/implement.shtml