Fast Implementation of Lemke's Algorithm for Rigid Body Contact Simulation

John E. Lloyd

January 18, 2005

Abstract

We present a fast method for solving rigid body contact problems with friction, based on optimizations incorporated into Lemke's algorithm for solving linear complementarity problems. These optimizations reduce the expected solution complexity (in the number of contacts) from $O(n^3)$ to nearly $O(nm + m^3)$, where m is the number of bodies in the system. For a fixed m the expected complexity is then close to $O(n)$. By simplifying internal computations our method also improves numerical robustness, and removes the need to explicitly compute the large matrices associated with rigid body contact problems.

1 Introduction

Computing the reaction forces that arise between rigid bodies in contact is useful in a wide variety of applications, including robotics, computer graphics, animation, haptics, and mechanical simulation.

This calculation can be expensive whenever there are multiple contacts, which can happen even in simple situations involving extended contact between two rigid bodies (Fig. 1). For example, contact between overlapping convex polygonal faces (Fig. 2) can be modeled by placing contacts at the vertices of their intersection, but even for two squares this may still produce up to eight contacts. Nor is it generally easy to eliminate redundant contacts in advance, since this involves calculations equivalent to solving the contact problem itself.

Figure 1: Some extended contacts with numerous contacts points. Left to right: block-in-corner (12 contacts), face-on-face (6 contacts), peg-in-hole (arbitrary contacts).

Rigid body contact can be formulated as a linear complementarity problem (or LCP, described below), which can then be solved by either indirect (iterative) methods, or direct (pivoting) methods [RWC92, Mur88]. With iterative methods (including impulse-based techniques [MC95, GBF03]), the chief concerns are accuracy and convergence. Impulse-based techniques are useful for handling collisions and producing visually plausible behaviors for large numbers of objects, but may suffer from low accuracy or slow convergence [Lac03]. For direct methods, the only one which has been proven[ST96, AP02] to work in the presence of friction is Lemke's algorithm, which will produce an exact answer but has a nominal expected time complexity of $O(n^3)$ in the number of contacts. Lemke's algorithm is also numerically sensitive, particularly in the presence of redundant contacts, and requires working with large matrices: for example, the LCP matrix for a problem with 16 contacts and an 8-sided friction cone approximation has a size of 160.

In this paper, we present an implementation of Lemke's algorithm for contact problems that removes some of these difficulties. First, the method is fast: in tests described below, the average solution time for a 16 contact peg-in-hole problem was reduced from 30 msec to 2 msec. The expected complexity also appears to reduce from $O(n^3)$ to nearly $O(nm + m^3)$, where m is the number of rigid bodies; for fixed m this becomes $O(n)$. This increases the practicality of rigid body simulation for real-time interactive applications, such as haptics. Also, there is no need to explicitly calculate the (large) LCP matrix, and numerical robustness is improved because of internal reductions in problem size.

Sections 2, 3, and 4 give background on contact problems, the solution

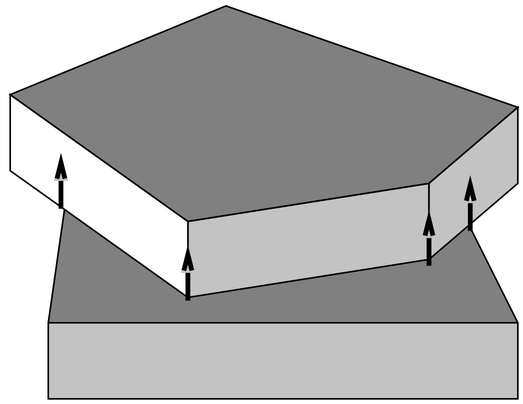

Figure 2: Contacts (arrows) modeling an extended face-on-face contact.

of LCPs by pivoting methods, and Lemke's algorithm. Section 5 will show how the problem's structure can used to reduce the complexity of the Lemke calculations, and Section 5.3 will show how to streamline things further by reducing the problem size. Experiments to test these methods are described in Section 5.4.

2 Problem Formulation

Rigid body contact has been described in many publications (e.g., [Bar94, AP97, SS98, Ste00, AP02]) and so will only be summarized here.

The dynamics equation for a rigid body system with contact constraints and m bodies can be expressed as

$$\mathcal{M}\dot{\mathbf{v}} - \tilde{\mathbf{N}}\boldsymbol{\theta} - \tilde{\mathbf{D}}\boldsymbol{\phi} = \mathbf{f}_x \tag{1}$$

where $\mathcal{M} \in \mathbb{R}^{6m \times 6m}$ is the block-diagonal system mass matrix containing the mass matrices $\mathcal{M}_k \in \mathbb{R}^{6\times 6}$ for each body, $\mathbf{v} = (\mathbf{v}_1^T \dots \mathbf{v}_m^T)^T$ and $\mathbf{f}_x = (\mathbf{f}_{x1}^T \dots \mathbf{f}_{xm}^T)^T$ are composite vectors in $\mathbb{R}^{6m}$ containing the spatial velocities and external forces (including coriolis forces) for each body, $\tilde{\mathbf{N}}$ and $\tilde{\mathbf{D}}$ are the constraint matrices for normal forces and friction, and $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ give the normal and friction force components acting along these constraints. Bilateral constraints can also be included in this formulation, but we will omit this for brevity.

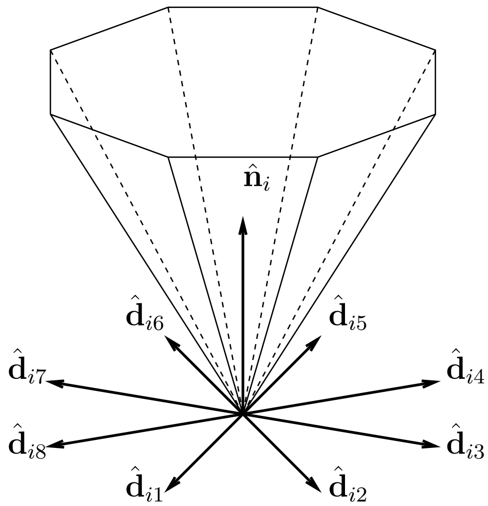

Figure 3: Friction directions $\hat{\mathbf{d}}_{ij}$ at contact i, and the associated polyhedral approximation to the friction cone.

Each contact i is associated with a point in space $\mathbf{p}_i$ and a normal vector $\hat{\mathbf{n}}_i$, as well as a set of d directions $\hat{\mathbf{d}}_{ij}, j \in \{1, \ldots, d\}$ which form a convex span of the tangent place perpendicular to $\hat{\mathbf{n}}_i$ and along which the frictional forces act (Figure 3)[1].

If there are n contacts, then $\tilde{\mathbf{N}} \in \mathbb{R}^{6m \times n}$ generates the system forces produced by the normal force components $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$. Each column of $\tilde{\mathbf{N}}$ corresponds to a contact i, and will contain, for each body k connected to this contact, 6 entries of the form

$$\pm \begin{pmatrix} \hat{\mathbf{n}}_i \\ {}^k\mathbf{p}_i \times \hat{\mathbf{n}}_i \end{pmatrix}$$

where the sign depends on whether the normal is facing into or away from the body, and ${}^k\mathbf{p}_i$ gives the contact point with respect to the body's coordinate frame. Other elements of $\tilde{\mathbf{N}}$ are 0. Since there are at most two bodies connected to each contact, each column of $\tilde{\mathbf{N}}$ will contain at most 12 non-zero elements.

[1] For simplicity of description, we assume that the number of friction directions d is the same at each contact, although this is not necessary.

Likewise, each of the d friction directions $\hat{\mathbf{d}}_{ij}$ at contact i will be associated with a friction force component ϕ_{ij}, and a column of $\tilde{\mathbf{D}} \in \mathbb{R}^{6m \times nd}$. The set of all friction components is given by $\boldsymbol{\phi} = (\phi_{11}, \phi_{12}, \ldots, \phi_{21}, \phi_{22}, \ldots \phi_{nd})$. As with $\tilde{\mathbf{N}}$, each column of $\tilde{\mathbf{D}}$ will contain, for each body k connected to the contact, 6 entries of the form

$$\pm \begin{pmatrix} \hat{\mathbf{d}}_{ij} \\ {}^k\mathbf{p}_i \times \hat{\mathbf{d}}_{ij} \end{pmatrix}$$

where opposite signs are selected for each of the (at most two) bodies connected to the contact. Again, each column of $\tilde{\mathbf{D}}$ will contain at most 12 non-zero elements.

When solving (1), it is common to integrate it over a time step h, in order to ensure the existence of solutions ([AP97, Ste00]). Using an explicit Euler step, with initial velocities $\mathbf{v}_0$, $\mathbf{v}$ can be obtained by solving

$$\mathcal{M}\mathbf{v} - \tilde{\mathbf{N}}\boldsymbol{\theta} - \tilde{\mathbf{D}}\boldsymbol{\phi} = \tilde{\mathbf{k}}_x \tag{2}$$

where $\tilde{\mathbf{k}}_x \equiv \mathbf{f}_x h + \mathcal{M}\mathbf{v}_0$ and $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are now impulse components. While we will use (2) in this paper, other formulations, based on $\dot{\mathbf{v}}$, or other integration schemes ([AP02]), or first order physics, all have the same structure as equation (8) below and so will benefit from the methods to be described.

The solution to (2) is also subject to constraints. First, we have the non-penetration constraint, which requires that the normal force components $\boldsymbol{\theta}$, as well as the normal velocity components (which we will denote by $\boldsymbol{\nu}$ and which can be computed by $\tilde{\mathbf{N}}^T\mathbf{v}$) must both be non-negative. Also, a non-zero normal force component implies a zero normal velocity component, and vice versa, so together we have[2]

$$\boldsymbol{\nu} = \tilde{\mathbf{N}}^T\mathbf{v} \geq 0, \quad \boldsymbol{\theta} \geq 0, \quad \boldsymbol{\nu}^T\boldsymbol{\theta} = 0. \tag{3}$$

Secondly, we want the friction forces to be as opposite as possible to the tangential velocity components. At each contact i, let $-\lambda_i$ be the minimum tangential velocity component among all the directions $\hat{\mathbf{d}}_{ij}$. Since the $\hat{\mathbf{d}}_{ij}$ form a convex span of the contact's tangent plane, $\lambda_i \geq 0$. Each tangential velocity component is $\tilde{\mathbf{d}}_{ij}^T\mathbf{v}$, where $\tilde{\mathbf{d}}_{ij}$ is the column of $\tilde{\mathbf{D}}$ generated by $\hat{\mathbf{d}}_{ij}$.

[2] A relational operator applied to a vector is assumed to imply satisfaction of that operator for each element.

Defining $\sigma_{ij} \equiv \tilde{\mathbf{d}}_{ij}^T \mathbf{v} + \lambda_i$, we see that $\sigma_{ij} \geq 0$. To ensure that friction acts along the direction(s) $\hat{\mathbf{d}}_{ij}$ most opposite to the tangential velocity, we also require that $\phi_{ij} > 0$ only if $\sigma_{ij} = 0$, and vice versa. Expressed for the whole system, these constraints take the form

$$\boldsymbol{\sigma} = \tilde{\mathbf{D}}^T \mathbf{v} + \mathbf{E}\boldsymbol{\lambda} \geq 0, \quad \boldsymbol{\lambda} \geq, \quad \boldsymbol{\sigma}^T \boldsymbol{\phi} = 0, \tag{4}$$

where $\boldsymbol{\sigma} = (\sigma_{11}, \sigma_{12}, \ldots, \sigma_{21}, \sigma_{22}, \ldots \sigma_{nd})$, $\boldsymbol{\lambda} \equiv (\lambda_1, \ldots, \lambda_n)$, and $\mathbf{E} \in \mathbb{R}^{nd \times n}$ with the form

$$\mathbf{E} = \begin{pmatrix} \mathbf{1} & & \\ & \ddots & \\ & & \mathbf{1} \end{pmatrix}, \ \mathbf{1} \in \mathbb{R}^d, \ \mathbf{1} = (1, \ldots, 1)^T.$$

Finally, the net frictional force at each contact i must lie inside the friction cone defined by the contact's coefficient of friction μ_i. Using a polyhedral approximation to the friction cone formed by the $\hat{\mathbf{d}}_{ij}$ (Figure 3), and defining $\gamma_i \equiv \mu_i \theta_i - \sum_j \phi_{ij}$, the friction cone constraint is enforced by requiring $\gamma_i \geq 0$. If $\gamma_i > 0$, the net force is inside the friction cone (stiction), implying there must be no tangential velocity and hence $\lambda_i = 0$. Conversely, if $\lambda_i > 0$, the frictional force must be on the friction cone and so $\gamma_i = 0$. Expressed for the entire system, these constraints take the form

$$\boldsymbol{\gamma} \equiv \boldsymbol{\mu}\boldsymbol{\theta} - \mathbf{E}^T \boldsymbol{\phi} \geq 0, \quad \boldsymbol{\lambda} \geq 0, \quad \boldsymbol{\gamma}^T \boldsymbol{\lambda} = 0, \tag{5}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{n \times n}$ is a diagonal matrix of friction coefficients.

The combination of (2) and constraints (3)-(5) can be expressed by one system of the form

$$\begin{pmatrix} 0 \\ \boldsymbol{\nu} \\ \boldsymbol{\sigma} \\ \boldsymbol{\gamma} \end{pmatrix} = \begin{pmatrix} \mathcal{M} & -\tilde{\mathbf{N}} & \tilde{\mathbf{D}} & 0 \\ \tilde{\mathbf{N}}^T & 0 & 0 & 0 \\ \tilde{\mathbf{D}}^T & 0 & 0 & \mathbf{E} \\ 0 & \boldsymbol{\mu} & -\mathbf{E}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\theta} \\ \boldsymbol{\phi} \\ \boldsymbol{\lambda} \end{pmatrix} + \begin{pmatrix} \tilde{\mathbf{k}}_x \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{6}$$
$$\boldsymbol{\nu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\lambda} \geq 0, \quad \boldsymbol{\nu}^T \boldsymbol{\theta} = \boldsymbol{\sigma}^T \boldsymbol{\phi} = \boldsymbol{\gamma}^T \boldsymbol{\lambda} = 0.$$

By solving for $\mathbf{v}$, the equality part of this system can be reduced to

$$\begin{pmatrix} \boldsymbol{\nu} \\ \boldsymbol{\sigma} \\ \boldsymbol{\gamma} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{N}}^T \mathcal{M}^{-1} \tilde{\mathbf{N}} & \tilde{\mathbf{N}}^T \mathcal{M}^{-1} \tilde{\mathbf{D}} & 0 \\ \tilde{\mathbf{D}}^T \mathcal{M}^{-1} \tilde{\mathbf{N}} & \tilde{\mathbf{D}}^T \mathcal{M}^{-1} \tilde{\mathbf{D}} & \mathbf{E} \\ \boldsymbol{\mu} & -\mathbf{E}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta} \\ \boldsymbol{\phi} \\ \boldsymbol{\lambda} \end{pmatrix} + \begin{pmatrix} \tilde{\mathbf{N}}^T \mathcal{M}^{-1} \tilde{\mathbf{k}}_x \\ \tilde{\mathbf{D}}^T \mathcal{M}^{-1} \tilde{\mathbf{k}}_x \\ 0 \end{pmatrix} .z \tag{7}$$

To simplify notation in the remainder of this paper, we assume that $\mathcal{M}$ is symmetric positive definite[3] and so can be factored as $\mathcal{M} = GG^T$. If we then define

$$\mathbf{N} \equiv G^{-1}\tilde{\mathbf{N}}, \quad \mathbf{D} \equiv G^{-1}\tilde{\mathbf{D}}, \quad \mathbf{k}_x \equiv G^{-1}\tilde{\mathbf{k}}_x,$$

we can further simplicfy (6)

$$\begin{pmatrix} \boldsymbol{\nu} \\ \boldsymbol{\sigma} \\ \boldsymbol{\gamma} \end{pmatrix} = \begin{pmatrix} \mathbf{N}^T\mathbf{N} & \mathbf{N}^T\mathbf{D} & 0 \\ \mathbf{D}^T\mathbf{N} & \mathbf{D}^T\mathbf{D} & \mathbf{E} \\ \boldsymbol{\mu} & -\mathbf{E}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta} \\ \boldsymbol{\phi} \\ \boldsymbol{\lambda} \end{pmatrix} + \begin{pmatrix} \mathbf{N}^T\mathbf{k}_x \\ \mathbf{D}^T\mathbf{k}_x \\ 0 \end{pmatrix}, \tag{8}$$
$$\boldsymbol{\nu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\lambda} \geq 0, \quad \boldsymbol{\nu}^T\boldsymbol{\theta} = \boldsymbol{\sigma}^T\boldsymbol{\phi} = \boldsymbol{\gamma}^T\boldsymbol{\lambda} = 0.$$

Solving this for $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ gives the reaction forces (or impulses) from which $\mathbf{v}$ can then be determined via (2).

System (8) is an example of a *linear complementarity problem* (LCP). It has been shown by various authors ([ST96, AP97, AP02]) that the contact LCP can always be solved by a pivoting technique known as Lemke's algorithm.

3 Solution of LCPs by Pivoting Methods

A linear complementarity problem can be stated as follows: solve

$$\mathbf{w} = \mathbf{M}\mathbf{z} + \mathbf{q} \tag{9}$$

for the variables $\mathbf{w}$ and $\mathbf{z}$, subject to the constraints

$$\mathbf{w} \geq 0, \ \mathbf{z} \geq 0, \ \mathbf{w}^T\mathbf{z} = 0$$

where $\mathbf{M}$ is a square matrix. If $\mathbf{q} \geq 0$ then we immediately have a solution given by $\mathbf{z} = 0$ and $\mathbf{w} = \mathbf{q}$. Otherwise, we can search for a solution by setting other combinations of $\mathbf{z}$ and $\mathbf{w}$ variables to zero. In particular, let $\mathbf{z}_\alpha$ and $\mathbf{w}_{\tilde{\alpha}}$ be equally sized subsets of the variables $\mathbf{z}$ and $\mathbf{w}$, formed from the index sets α and $\tilde{\alpha}$, and let the remaining variables by given by $\mathbf{z}_\beta$ and $\mathbf{w}_{\tilde{\beta}}$. With a suitable row/column rearrangement, (9) can then be partitioned as

$$\begin{pmatrix} \mathbf{w}_{\tilde{\alpha}} \\ \mathbf{w}_{\tilde{\beta}} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{\tilde{\alpha}\alpha} & \mathbf{M}_{\tilde{\alpha}\beta} \\ \mathbf{M}_{\tilde{\beta}\alpha} & \mathbf{M}_{\tilde{\beta}\beta} \end{pmatrix} \begin{pmatrix} \mathbf{z}_\alpha \\ \mathbf{z}_\beta \end{pmatrix} + \begin{pmatrix} \mathbf{q}_{\tilde{\alpha}} \\ \mathbf{q}_{\tilde{\beta}} \end{pmatrix}. \tag{10}$$

[3] If $\mathcal{M}$ is not SPD, the results to follow are still valid, albeit in a slightly more complicated form.

If $\mathbf{M}_{\tilde{\alpha}\alpha}$ is non-singular, we can exchange $\mathbf{w}_{\tilde{\alpha}}$ and $\mathbf{z}_{\alpha}$ to get a *pivoted* system

$$\begin{pmatrix} \mathbf{z}_{\alpha} \\ \mathbf{w}_{\tilde{\beta}} \end{pmatrix} = \mathbf{M}' \begin{pmatrix} \mathbf{w}_{\tilde{\alpha}} \\ \mathbf{z}_{\beta} \end{pmatrix} + \mathbf{q}', \tag{11}$$

where

$$\begin{aligned} \mathbf{M}' &= \begin{pmatrix} \mathbf{M}_{\tilde{\alpha}\alpha}^{-1} & -\mathbf{M}_{\tilde{\alpha}\alpha}^{-1}\mathbf{M}_{\tilde{\alpha}\beta} \\ \mathbf{M}_{\tilde{\beta}\alpha}\mathbf{M}_{\tilde{\alpha}\alpha}^{-1} & \mathbf{M}_{\tilde{\beta}\beta} - \mathbf{M}_{\tilde{\beta}\alpha}\mathbf{M}_{\tilde{\alpha}\alpha}^{-1}\mathbf{M}_{\tilde{\alpha}\beta} \end{pmatrix}, \\ \mathbf{q}' &= \begin{pmatrix} \mathbf{q}'_{\tilde{\alpha}} \\ \mathbf{q}'_{\tilde{\beta}} \end{pmatrix} = \begin{pmatrix} -\mathbf{M}_{\tilde{\alpha}\alpha}^{-1}\mathbf{q}_{\tilde{\alpha}} \\ \mathbf{q}_{\tilde{\beta}} - \mathbf{M}_{\tilde{\beta}\alpha}\mathbf{M}_{\tilde{\alpha}\alpha}^{-1}\mathbf{q}_{\tilde{\alpha}} \end{pmatrix}. \end{aligned} \tag{12}$$

The variable sets $\{\mathbf{z}_{\alpha}, \mathbf{w}_{\tilde{\beta}}\}$ and $\{\mathbf{w}_{\tilde{\alpha}}, \mathbf{z}_{\beta}\}$ are called the *basic* and *non-basic* variables, respectively. The basic variables are associated with a *basis matrix* $\mathbf{B}$, defined, with respect to the partition (10), by

$$\mathbf{B} = \begin{pmatrix} 0 & -\mathbf{M}_{\tilde{\alpha}\alpha} \\ \mathbf{I} & -\mathbf{M}_{\tilde{\beta}\alpha} \end{pmatrix} \tag{13}$$

and whose inverse is given by

$$\mathbf{B}^{-1} = \begin{pmatrix} -\mathbf{M}_{\tilde{\alpha}\alpha}^{-1} & 0 \\ -\mathbf{M}_{\tilde{\beta}\alpha}\mathbf{M}_{\tilde{\alpha}\alpha}^{-1} & \mathbf{I} \end{pmatrix}. \tag{14}$$

Form (12), we see that

$$\mathbf{q}' = \mathbf{B}^{-1} \begin{pmatrix} \mathbf{q}_{\tilde{\alpha}} \\ \mathbf{q}_{\tilde{\beta}} \end{pmatrix}. \tag{15}$$

$\mathbf{w}$ and $\mathbf{z}$ variables with matching indices are called *complementary*, with each being the *complement* of the other. If the index sets α and $\tilde{\alpha}$ are identical, then $\mathbf{z}_{\alpha}$ is complementary to $\mathbf{w}_{\alpha}$ and the basis is a *complementary basis*. If the basis for a pivoted system (11) is complementary and $\mathbf{q}' \geq 0$, then we have a solution to the LCP given by

$$\mathbf{z}_{\alpha} = \mathbf{q}'_{\tilde{\alpha}},\ \mathbf{w}_{\tilde{\beta}} = \mathbf{q}'_{\tilde{\beta}},\ \mathbf{w}_{\tilde{\alpha}} = \mathbf{z}_{\beta} = 0. \tag{16}$$

Generally speaking, a pivoting method searches for a solution to an LCP be incrementally exchanging, or *pivoting*, variables (usually one pair at a time) in order to find a complementary basis for which $\mathbf{q}' \geq 0$.

4 Lemke's Algorithm

Lemke's algorithm [RWC92, Mur88] is a pivoting method where the search is facilitated by augmenting the LCP (9) with an auxiliary variable z_0 and a *covering vector* $\mathbf{c} > 0$:

$$\mathbf{w} = \bar{\mathbf{M}} \begin{pmatrix} \mathbf{z} \\ z_0 \end{pmatrix} + \mathbf{q}, \quad \bar{\mathbf{M}} \equiv \begin{pmatrix} \mathbf{M} & \mathbf{c} \end{pmatrix} \tag{17}$$

This augmented system can be partitioned and pivoted exactly as shown in (10) and (11), with $\mathbf{c}$ now included in the appropriate partitions $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}$, $\bar{\mathbf{M}}_{\tilde{\beta}\alpha}$, etc.

The algorithm works as follows:

- **Step 0.** If $\mathbf{q} \geq 0$, stop; $\mathbf{z} = 0$ solves the system. Otherwise, choose variable w_r in $\mathbf{w}$ for which $r = \arg\min\{q_i/c_i\}$, and pivot z_0 with w_r. Set the *driving variable* y_r to z_r.

- **Step 1.** Let $\mathbf{m}'$ be the column of the pivoted matrix $\bar{\mathbf{M}}'$ corresponding to y_r. If $\mathbf{m}' \geq 0$, stop: the LCP has no solution or is unsolvable by Lemke's algorithm. Otherwise, let y_s be the (basic) variable indexed by

$$s = \arg\min\{-q'_i/m'_i : m'_i < 0\} \tag{18}$$

- **Step 2.** If $y_s = z_0$, pivot z_0 with y_r and stop: the resulting $\mathbf{q}'$ solves the LCP. Otherwise, pivot y_s with y_r, set the new driving variable y_r to the complement of y_s, and return to step 1.

If test (18) results in a tie, the LCP is *degenerate*, and one or more columns from the inverse basis matrix $\mathbf{B}^{-1}$ (equation 14) may be needed to resolve the tie; see section 4.9 in [RWC92] or 2.2.7 of [Mur88]. Since $\mathbf{B}^{-1}$ is formed from a (negated) submatrix of $\bar{\mathbf{M}}'$ and a submatrix of $\mathbf{I}$, determining one of its columns is at worst equivalent to computing a column of $\bar{\mathbf{M}}'$, as described below.

It should be noted that after the initial pivot in Step 0, we always have $\mathbf{q}' \geq 0$; this is possible because $\mathbf{c} \geq 0$. However, in this case, $\mathbf{q}' \geq 0$ does *not* imply a solution of the form (16) because the basis is not complementary. Instead, the basis is *almost complementary*: the variable sets $\mathbf{z}_\alpha$ and $\mathbf{w}_{\tilde{\alpha}}$ are complementary *except* that $\mathbf{z}_\alpha$ also contains z_0 and $\mathbf{w}_{\tilde{\alpha}}$ contains a corresponding non-complementary variable w_r. Stated with respect to the index sets, α and $\tilde{\alpha}$ are identical except that α also contains 0 and $\tilde{\alpha}$ also contains r. As it happens, w_r is either the driving variable y_r, or its complement, depending on whether y_r is a $\mathbf{w}$ or $\mathbf{z}$ variable. A successful termination of the algorithm in Step 2 involves pivoting z_0 with y_r, thus removing z_0 and w_r from $\mathbf{z}_\alpha$ and $\mathbf{w}_{\tilde{\alpha}}$, so that the resulting (final) basis is complementary and hence $\mathbf{q}' \geq 0$ implies a solution of the form (16).

Generally, each pivot step requires computing $\mathbf{q}'$ and a single column $\mathbf{m}'$ from $\bar{\mathbf{M}}'$, using the formulae (12). $\mathbf{q}'$ takes the recursive form

$$\mathbf{q}' = \begin{pmatrix} \mathbf{q}'_{\tilde{\alpha}} \\ \mathbf{q}'_{\tilde{\beta}} \end{pmatrix} = \begin{pmatrix} -\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}^{-1} \mathbf{q}_{\tilde{\alpha}} \\ \mathbf{q}_{\tilde{\beta}} + \bar{\mathbf{M}}_{\tilde{\beta}\alpha} \mathbf{q}'_{\tilde{\alpha}} \end{pmatrix}. \tag{19}$$

For $\mathbf{m}'$, if the driving variable y_r is a $\mathbf{z}$ variable, then

$$\mathbf{m}' = \begin{pmatrix} \mathbf{m}'_{\tilde{\alpha}} \\ \mathbf{m}'_{\tilde{\beta}} \end{pmatrix} = \begin{pmatrix} -\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}^{-1} \mathbf{m}_{\tilde{\alpha}} \\ \mathbf{m}_{\tilde{\beta}} + \bar{\mathbf{M}}_{\tilde{\beta}\alpha} \mathbf{m}'_{\tilde{\alpha}} \end{pmatrix} \tag{20}$$

where $\mathbf{m}_{\tilde{\alpha}}$ and $\mathbf{m}_{\tilde{\beta}}$ are the $\tilde{\alpha}$ and $\tilde{\beta}$ partitions of the column of $\mathbf{M}$ corresponding to y_r. Otherwise, if y_r is a $\mathbf{w}$ variable,

$$\mathbf{m}' = \begin{pmatrix} \mathbf{m}'_{\tilde{\alpha}} \\ \mathbf{m}'_{\tilde{\beta}} \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{M}}_{\tilde{\alpha}\alpha}^{-1} \mathbf{e}_r \\ \bar{\mathbf{M}}_{\tilde{\beta}\alpha} \mathbf{m}'_{\tilde{\alpha}} \end{pmatrix} \tag{21}$$

where $\mathbf{e}_r$ is the $\tilde{\alpha}$ partition of the r-th column of the identity matrix $\mathbf{I}$.

Calculations (19)-(21) each entail solving a system

$$\bar{\mathbf{M}}_{\tilde{\alpha}\alpha} \mathbf{x} = \mathbf{b} \tag{22}$$

where $\mathbf{b}$ is $\mathbf{q}_{\tilde{\alpha}}$, $\mathbf{m}_{\tilde{\alpha}}$, or $\mathbf{e}_r$. This, plus the other calculations in (19)-(21), can be done in $O(n^2)$ time if $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}$ is available in factored form. After each pivot, this factorization can also be updated in $O(n^2)$ time, since $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}$ will have changed by at most one row and/or column [RWC92], [Sar78] (although this update must be done carefully to limit numerical errors [Tom78, GMSW87]). Then, since Lemke's algorithm typically requires $O(n)$ pivots [RWC92], its overall expected complexity is $O(n^3)$. It should be noted, however, that problems do exist for which Lemke's algorithm has a worst-case complexity of $O(2^n)$ ([Mur88], Chapter 6).

5 Simplification using Matrix Structure

We now show how the structure of (8) allows us to compute $\mathbf{q}'$ and $\mathbf{m}'$ efficiently and without having to explicitly form the matrix $\bar{\mathbf{M}}$.

We will simplify our presentation by ignoring z_0 and $\mathbf{c}$, so that $\bar{\mathbf{M}} = \mathbf{M}$, and by assuming that $\mathbf{M}_{\tilde{\alpha}\alpha}$ corresponds to a complementary basis (i.e., the index sets $\tilde{\alpha}$ and α are identical). In fact, the bases produced by Lemke's algorithm are almost complementary, as described in Section 4. This means that $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}$ is complementary, *except* for an additional column (corresponding to z_0) formed from the covering vector $\mathbf{c}$, and an additional row (corresponding to w_r). The modifications required to deal with this are presented in appendix A.

5.1 Simplification for Complementary Bases

We begin by showing how to partition (8) in accordance with (10). Examining (8), it can be seen that $\mathbf{M}$ and $\mathbf{q}$ have a size of $n(2+d)$, with $\mathbf{w}$ composed of the variables $\boldsymbol{\nu}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\gamma}$, and $\mathbf{z}$ composed of $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and $\boldsymbol{\lambda}$.

The basic $\mathbf{z}$ variables $\mathbf{z}_\alpha$ will be denoted by $\boldsymbol{\theta}_\alpha$, $\boldsymbol{\phi}_\alpha$, and $\boldsymbol{\lambda}_\alpha$. We will let $\mathbf{N}_\alpha$ and $\mathbf{D}_\alpha$ be the submatrices of $\mathbf{N}$ and $\mathbf{D}$ associated with $\boldsymbol{\theta}_\alpha$ and $\boldsymbol{\phi}_\alpha$, and let $\mathbf{N}_\beta$ and $\mathbf{D}_\beta$ be the submatrices of $\mathbf{N}$ and $\mathbf{D}$ associated with the non-basic $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ variables (which are indexed by β). Because we are assuming a complementary basis, $\alpha = \tilde{\alpha}$ and $\beta = \tilde{\beta}$, and the partition components can be arranged into the following form:

$$\mathbf{M}_{\tilde{\alpha}\alpha} = \begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{N}_\alpha & \mathbf{N}_\alpha^T\mathbf{D}_\alpha & 0 \\ \mathbf{D}_\alpha^T\mathbf{N}_\alpha & \mathbf{D}_\alpha^T\mathbf{D}_\alpha & \mathbf{E}_{\alpha\alpha} \\ \boldsymbol{\mu}_{\alpha\alpha} & -\mathbf{E}_{\alpha\alpha}^T & 0 \end{pmatrix}, \quad \mathbf{M}_{\tilde{\alpha}\beta} = \begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{N}_\beta & \mathbf{N}_\alpha^T\mathbf{D}_\beta & 0 \\ \mathbf{D}_\alpha^T\mathbf{N}_\beta & \mathbf{D}_\alpha^T\mathbf{D}_\beta & \mathbf{E}_{\alpha\beta} \\ \boldsymbol{\mu}_{\alpha\beta} & -\mathbf{E}_{\beta\alpha}^T & 0 \end{pmatrix},$$

$$\mathbf{M}_{\tilde{\beta}\alpha} = \begin{pmatrix} \mathbf{N}_\beta^T\mathbf{N}_\alpha & \mathbf{N}_\beta^T\mathbf{D}_\alpha & 0 \\ \mathbf{D}_\beta^T\mathbf{N}_\alpha & \mathbf{D}_\beta^T\mathbf{D}_\alpha & \mathbf{E}_{\beta\alpha} \\ \boldsymbol{\mu}_{\beta\alpha} & -\mathbf{E}_{\alpha\beta}^T & 0 \end{pmatrix}, \quad \mathbf{M}_{\tilde{\beta}\beta} = \begin{pmatrix} \mathbf{N}_\beta^T\mathbf{N}_\beta & \mathbf{N}_\beta^T\mathbf{D}_\beta & 0 \\ \mathbf{D}_\beta^T\mathbf{N}_\beta & \mathbf{D}_\beta^T\mathbf{D}_\beta & \mathbf{E}_{\beta\beta} \\ \boldsymbol{\mu}_{\beta\beta} & -\mathbf{E}_{\beta\beta}^T & 0 \end{pmatrix},$$

$$\mathbf{q}_{\tilde{\alpha}} = \begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{k}_x \\ \mathbf{D}_\alpha^T\mathbf{k}_x \\ 0 \end{pmatrix}, \quad \mathbf{q}_{\tilde{\beta}} = \begin{pmatrix} \mathbf{N}_\beta^T\mathbf{k}_x \\ \mathbf{D}_\beta^T\mathbf{k}_x \\ 0 \end{pmatrix} \tag{23}$$

where $\mathbf{E}_{ab}$ and $\boldsymbol{\mu}_{ab}$ denote submatrices of $\mathbf{E}$ and $\boldsymbol{\mu}$ formed from rows and columns indexed (with respect to $\mathbf{M}$) by the index sets a and b.

The main calculation we are concerned with is solving $\mathbf{M}_{\tilde{\alpha}\alpha}\mathbf{x} = \mathbf{b}$ where $\mathbf{b}$ is either $\mathbf{q}_{\tilde{\alpha}}$, $\mathbf{m}_{\tilde{\alpha}}$, or $\mathbf{e}_r$, and $\mathbf{m}_{\tilde{\alpha}}$ is one of the columns of $\mathbf{M}_{\tilde{\alpha}\beta}$. Generalizing our use of $\boldsymbol{\theta}_\alpha$, $\boldsymbol{\phi}_\alpha$, and $\boldsymbol{\lambda}_\alpha$ to denote the components of $\mathbf{x}$, the expanded form of this equation is

$$\begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{N}_\alpha & \mathbf{N}_\alpha^T\mathbf{D}_\alpha & 0 \\ \mathbf{D}_\alpha^T\mathbf{N}_\alpha & \mathbf{D}_\alpha^T\mathbf{D}_\alpha & \mathbf{E}_{\alpha\alpha} \\ \boldsymbol{\mu}_{\alpha\alpha} & -\mathbf{E}_{\alpha\alpha}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\alpha \\ \boldsymbol{\lambda}_\alpha \end{pmatrix} = \begin{pmatrix} \mathbf{b}_\nu \\ \mathbf{b}_\sigma \\ \mathbf{b}_\gamma \end{pmatrix} \tag{24}$$

where $\mathbf{b}_\nu$, $\mathbf{b}_\sigma$, and $\mathbf{b}_\gamma$ are appropriate partitions of $\mathbf{b}$.

Since each column of $\mathbf{E}_{\alpha\alpha}$ has at least one non-zero unit entry (proven in Appendix C), it can, with a suitable rearrangement, be partitioned into

$$\begin{pmatrix} \mathbf{E}_{\kappa\kappa} \\ \mathbf{I} \end{pmatrix}. \tag{25}$$

Noting that each row of $\mathbf{E}_{\alpha\alpha}$ is associated with an element of $\boldsymbol{\phi}_\alpha$, let $\boldsymbol{\phi}_\kappa$ and $\boldsymbol{\phi}_x$ be those elements of $\boldsymbol{\phi}_\alpha$ associated with $\mathbf{E}_{\kappa\kappa}$ and $\mathbf{I}$, respectively, and let $\mathbf{D}_\kappa$ and $\mathbf{D}_x$ be the corresponding submatrices of $\mathbf{D}_\alpha$. This allows (24) to be further partitioned into

$$\begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{N}_\alpha & \mathbf{N}_\alpha^T\mathbf{D}_\kappa & \mathbf{N}_\alpha^T\mathbf{D}_x & 0 \\ \mathbf{D}_\kappa^T\mathbf{N}_\alpha & \mathbf{D}_\kappa^T\mathbf{D}_\kappa & \mathbf{D}_\kappa^T\mathbf{D}_x & \mathbf{E}_{\kappa\kappa} \\ \mathbf{D}_x^T\mathbf{N}_\alpha & \mathbf{D}_x^T\mathbf{D}_\kappa & \mathbf{D}_x^T\mathbf{D}_x & \mathbf{I} \\ \boldsymbol{\mu}_{\alpha\alpha} & -\mathbf{E}_{\kappa\kappa}^T & -\mathbf{I} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\kappa \\ \boldsymbol{\phi}_x \\ \boldsymbol{\lambda}_\alpha \end{pmatrix} = \begin{pmatrix} \mathbf{b}_\nu \\ \mathbf{b}_\kappa \\ \mathbf{b}_x \\ \mathbf{b}_\gamma \end{pmatrix}. \tag{26}$$

Now we can solve for $\boldsymbol{\phi}_x$:

$$\boldsymbol{\phi}_x = \boldsymbol{\mu}_{\alpha\alpha}\boldsymbol{\theta}_\alpha - \mathbf{E}_{\kappa\kappa}^T\boldsymbol{\phi}_\kappa - \mathbf{b}_\gamma. \tag{27}$$

Then, using a block elementary column operation of the form

$$\begin{pmatrix} A & B \\ C & -\mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & 0 \\ C & \mathbf{I} \end{pmatrix} = \begin{pmatrix} A + BC & B \\ 0 & -\mathbf{I} \end{pmatrix}$$

and defining

$$\mathbf{N}_* \equiv \mathbf{N}_\alpha + \mathbf{D}_x\boldsymbol{\mu}_{\alpha\alpha},\ \mathbf{D}_* \equiv \mathbf{D}_\kappa - \mathbf{D}_x\mathbf{E}_{\kappa\kappa}^T,\ \mathbf{b}_* \equiv \mathbf{D}_x\mathbf{b}_\gamma \tag{28}$$

we can eliminate $\boldsymbol{\phi}_x$ and reduce the system to

$$\begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{N}_* & \mathbf{N}_\alpha^T\mathbf{D}_* & 0 \\ \mathbf{D}_\kappa^T\mathbf{N}_* & \mathbf{D}_\kappa^T\mathbf{D}_* & \mathbf{E}_{\kappa\kappa} \\ \mathbf{D}_x^T\mathbf{N}_* & \mathbf{D}_x^T\mathbf{D}_* & \mathbf{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\kappa \\ \boldsymbol{\lambda}_\alpha \end{pmatrix} = \begin{pmatrix} \mathbf{b}_\nu \\ \mathbf{b}_\kappa \\ \mathbf{b}_x \end{pmatrix} + \begin{pmatrix} \mathbf{N}_\alpha^T \\ \mathbf{D}_\kappa^T \\ \mathbf{D}_x^T \end{pmatrix} \mathbf{b}_*.$$

Next, we can define

$$\mathbf{v}_c \equiv \mathbf{N}_*\boldsymbol{\theta}_\alpha + \mathbf{D}_*\boldsymbol{\phi}_\kappa - \mathbf{b}_*, \tag{29}$$

solve for $\boldsymbol{\lambda}_\alpha$:

$$\boldsymbol{\lambda}_\alpha = -\mathbf{D}_x^T\mathbf{v}_c + \mathbf{b}_x, \tag{30}$$

and then use a block elementary row operation of the form

$$\begin{pmatrix} \mathbf{I} & -C \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} A & C \\ B & \mathbf{I} \end{pmatrix} = \begin{pmatrix} A - CB & 0 \\ B & \mathbf{I} \end{pmatrix}$$

to eliminate $\boldsymbol{\lambda}_\alpha$ and further reduce the system to

$$\begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{N}_* & \mathbf{N}_\alpha^T\mathbf{D}_* \\ \mathbf{D}_*^T\mathbf{N}_* & \mathbf{D}_*^T\mathbf{D}_* \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\kappa \end{pmatrix} = \begin{pmatrix} \mathbf{b}_\nu \\ \mathbf{b}'_\kappa \end{pmatrix} + \begin{pmatrix} \mathbf{N}_\alpha^T \\ \mathbf{D}_*^T \end{pmatrix} \mathbf{b}_* \tag{31}$$

where $\mathbf{b}'_\kappa \equiv \mathbf{b}_\kappa - \mathbf{E}_{\kappa\kappa}\mathbf{b}_x$. This smaller system is solved for $\boldsymbol{\theta}_\alpha$ and $\boldsymbol{\phi}_\kappa$, after which we back-solve for $\boldsymbol{\phi}_x$ and $\boldsymbol{\lambda}_\alpha$ using (27) and (30).

Computations (19)-(21) also involve calculating $\mathbf{M}_{\tilde{\beta}\alpha}\mathbf{x}$. From (23), we see that this takes the expanded form

$$\mathbf{M}_{\tilde{\beta}\alpha}\mathbf{x} = \begin{pmatrix} \mathbf{N}_\beta^T\mathbf{N}_\alpha & \mathbf{N}_\beta^T\mathbf{D}_\alpha & 0 \\ \mathbf{D}_\beta^T\mathbf{N}_\alpha & \mathbf{D}_\beta^T\mathbf{D}_\alpha & \mathbf{E}_{\beta\alpha} \\ \boldsymbol{\mu}_{\beta\alpha} & -\mathbf{E}_{\alpha\beta}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\alpha \\ \boldsymbol{\lambda}_\alpha \end{pmatrix}. \tag{32}$$

Now, combining the definition of $\mathbf{v}_c$ in (29) with (28) and (27), we get

$$\begin{aligned} \mathbf{v}_c &= (\mathbf{N}_\alpha + \mathbf{D}_x\boldsymbol{\mu}_{\alpha\alpha})\boldsymbol{\theta}_\alpha + (\mathbf{D}_\kappa - \mathbf{D}_x\mathbf{E}_{\kappa\kappa}^T)\boldsymbol{\phi}_\kappa - \mathbf{D}_x\mathbf{b}_\gamma \\ &= \mathbf{N}_\alpha\boldsymbol{\theta}_\alpha + \mathbf{D}_\kappa\boldsymbol{\phi}_\kappa + \mathbf{D}_x\boldsymbol{\phi}_x \\ &= \mathbf{N}_\alpha\boldsymbol{\theta}_\alpha + \mathbf{D}_\alpha\boldsymbol{\phi}_\alpha \end{aligned}$$

so that (32) becomes

$$\mathbf{M}_{\tilde{\beta}\alpha}\mathbf{x} = \begin{pmatrix} \mathbf{N}_\beta^T \\ \mathbf{D}_\beta^T \\ 0 \end{pmatrix} \mathbf{v}_c + \begin{pmatrix} 0 \\ \mathbf{E}_{\beta\alpha}\boldsymbol{\lambda}_\alpha \\ \boldsymbol{\mu}_{\beta\alpha}\boldsymbol{\theta}_\alpha - \mathbf{E}_{\alpha\beta}^T\boldsymbol{\phi}_\alpha \end{pmatrix}. \tag{33}$$

5.2 Expected Complexity

Matrix notation aside, the calculations in the previous section are actually quite easy. This is mainly because

1. Each column of $\mathbf{E}_{\alpha\alpha}$ has at most 2 non-zero unit entries (proven in Appendix B), and so each column of $\mathbf{E}_{\kappa\kappa}$ has at most 1 non-zero unit entry;

2. Each row or column of $\boldsymbol{\mu}_{\alpha\alpha}$ has at most one non-zero entry (since it is a submatrix of a diagonal matrix);

3. Each column of $\mathbf{N}$ and $\mathbf{D}$ (and hence all their submatrices) has at most 12 non-zero entries (as discussed in Section 2);

To actually analyze the complexity of these calculations, it will be useful to let r be the size of the reduced system matrix in (31). The the combined number of elements in $\boldsymbol{\theta}_\alpha$ and $\boldsymbol{\phi}_\kappa$ will be r, and therefore $\mathbf{N}_\alpha$, $\mathbf{N}_*$, $\mathbf{D}_\kappa$ and $\mathbf{D}_*$ will each have a size of $6m \times O(r)$.

We start by considering the formation of $\mathbf{b}$, which as mentioned earlier will be either $\mathbf{q}_{\tilde{\alpha}}$, $\mathbf{m}_{\tilde{\alpha}}$, or e_r, where $\mathbf{m}_{\tilde{\alpha}}$ is one of the columns of $\mathbf{M}_{\tilde{\alpha}\beta}$ in (23). Since each column of $\mathbf{N}$ and $\mathbf{D}$ has at most 12 non-zero entries (remark 3, above), it is easy to see that in all cases we can form $\mathbf{b}$ with complexity $O(n)$.

Next, examining (26), we see that $\boldsymbol{\phi}_x$ and $\boldsymbol{\lambda}_\alpha$ have the same number of elements. Since the size of $\boldsymbol{\lambda}_\alpha$ is $O(n)$, the size of $\boldsymbol{\phi}_x$ must also be $O(n)$, and this combined with remarks 1 and 2 means that computation of $\boldsymbol{\phi}_x$ in (27) has $O(n)$ complexity. Similarly, since $\mathbf{N}_\alpha$ and $\mathbf{D}_\kappa$ each have $O(r)$ columns, the computation of $\mathbf{N}_*$ and $\mathbf{D}_*$ in (28) has $O(r)$ complexity. Since $\mathbf{b}_*$ and $\mathbf{v}_c$ each have $6m$ elements, the columns of $\mathbf{D}_x$, $\mathbf{N}_*$, and $\mathbf{D}_*$ each have at most 12 elements (remark 3), and $\mathbf{N}_*$ and $\mathbf{D}_*$ have $O(r)$ columns, the complexity of computing $\mathbf{b}_*$ and $\mathbf{v}_c$ in (28) and (30) is $O(m)$ and $O(r+m)$, respectively. Also from remark 3, the computation of $\boldsymbol{\lambda}_\alpha$ in (30) is $O(n)$, and the formation of the reduced matrix and right-hand vector in (31) is $O(r^2)$ and $O(r)$.

The solution of the system (31) will nominally require $O(r^3)$ operations, although if r is large we can keep the matrix in a factored form that is updated after each pivot, and so reduce the solution complexity to $O(r^2)$. The remaining calculations to determine $\mathbf{q}'$ or $\mathbf{m}'$ will involve formula (33), which by analysis similar to the above has a complexity of $O(n)$. Merging these results, we find that the complexity of computing $\mathbf{q}'$ or $\mathbf{m}'$, and hence the complexity for each pivot step of Lemke's algorithm, is $O(m + n + r^2)$.

Now, the matrix in (31) is equal to the product

$$\begin{pmatrix} \mathbf{N}_\alpha^T \\ \mathbf{D}_*^T \end{pmatrix} \begin{pmatrix} \mathbf{N}_* & \mathbf{D}_* \end{pmatrix}.$$

The non-singularity of $\mathbf{M}_{\tilde{\alpha}\alpha}$ ensures that this matrix has full rank, and so its size r must be bounded by the rank of each factor. These factors each have one dimension equal to $6m$, which in turn bounds their rank, and so we have that $r \leq 6m$. This means the per-pivot complexity of $O(m + n + r^2)$ is bounded by $O(n + m^2)$. Since the expected number of pivots in Lemke's algorithm is of the same order as the size of $\mathbf{M}$, which is $n(2 + d)$, we obtain an overall expected complexity of $O(n^2 + nm^2)$.

Generally speaking, we can expect $m \leq n$, since otherwise there will be more bodies than contacts and the problem will likely decompose into subproblems. If m is noticeably smaller than n, we should get a large improvement over the standard Lemke algorithm, for which the expected complexity is $O(n^3)$. If m is fixed, the complexity improves still further to $O(n^2)$, which is consistent with the tests labeled *structural* in section 5.4.

In summary, what we have done is reduce the calculations to a form that is partly constrained by the number of degrees of freedom in the system, rather than the number of contacts.

5.3 Reducing the Problem Size

In this section, we show how to reduce the expected complexity still further by reducing the expected number of pivots.

The number of pivots is bounded from below by the number of basic $\mathbf{z}$ variables in the final solution. A $\boldsymbol{\lambda}$ variable will be non-zero (and hence basic) for any contact exhibiting tangential motion (Section 2). Since this is possible at all contacts, we can expect $O(n)$ $\boldsymbol{\lambda}$ variables to be basic, implying that the number of pivots will be at least $O(n)$.

To reduce this, we observe that we don't actually need all the values of $\boldsymbol{\lambda}$ in order to solve the contact problem. In particular, we need not consider values of $\boldsymbol{\lambda}$ or $\boldsymbol{\phi}$ for any contact i which is inactive (i.e., for which θ_i non-basic). To see this, observe that if θ_i is non-basic, then $\theta_i = 0$ and, by the friction cone constraints (5), all the $\boldsymbol{\phi}$ variables associated with that contact must be zero as well. An inactive contact therefore makes no contribution to $\mathbf{v}$.

Put another way, we need not be concerned about the frictional aspect of a contact until that contact becomes active. We can therefore begin the solution of a contact problem using a simple, frictionless, LCP of the form

$$\boldsymbol{\nu} \ = \ \mathbf{N}^T \boldsymbol{\theta} + \mathbf{N}^T \mathbf{k}_x.$$

Then whenever a contact becomes active for the first time, we expand the system (into the form of (8)) by adding the necessary $\boldsymbol{\phi}$, $\boldsymbol{\lambda}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\gamma}$ variables (and corresponding matrix rows and columns) associated with that contact, and then continue with Lemke's algorithm using the existing set of basic variables. This process of expanding an LCP online while it is being solved is described in Section 4.6 of [RWC92]. Computationally, all that is required is expanding the covering vector $\mathbf{c}$ and assigning it appropriate values, as described in Appendix D.

Ideally, most of the contacts for which the problem is expanded will remain active in the final solution, so that few pivots will be expended for other, inactive, contacts. Since the number of active contacts (i.e., basic $\boldsymbol{\theta}$ variables) is bounded by the size r of the matrix in (31), and r is $O(m)$, we can, in the best case, expect the number of pivots to be reduced from $O(n)$ to $O(m)$.

Combined with the per-pivot complexity of $O(n+m^2)$ described in Section 5.2, this would give an overall expected complexity of $O(nm + m^3)$, or $O(n)$ when m is fixed. We do in fact observe behavior close to $O(n)$ in the tests named *reduced* in section 5.4.

5.4 Experimental Results

We now present some comparative tests to show the utility of the methods described in this paper. All tests were implemented in Java (version 1.4.2, with HotSpot) and executed on a 1 GHz Pentium III-M. Each computation was done with $d = 8$ and the friction coefficients μ_i in the range 0.2 to 0.3.

Each test measured expected solution times for a particular set of contact problems using three methods:

- *Standard:* A standard, efficient implementation of Lemke's algorithm as described in [Sar78].
- *Structural:* An implementation using the structural simplifications described in Section 5.

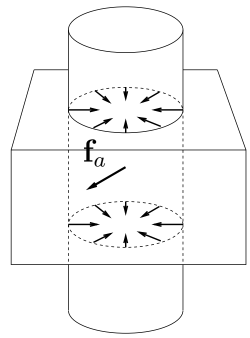

Figure 4: Peg-in-hole, with 16 contacts and an applied wrench $\mathbf{f}_a$.

- *Reduced:* An implementation using Section 5 plus the problem reduction described in Section 5.3.

Expected solution times were measured by solving each problem 20 times with randomly generated external forces, and averaging the computation times.

The first test was for a single body ($m = 1$) and involved computing the reaction forces on a peg passing through a hole in a fixed block (Fig. 4), in response to a random wrench applied at the center. Contacts were arranged around the hole's top and bottom, and expected computation times were measured for different numbers of contacts ranging from 8 to 32. The results are shown in 5, with the top, middle, and bottom lines corresponding to the *standard*, *structural*, and *reduced* methods, respectively. Our *structural* and *reduced* methods (shown close-up in Fig. 6) can be seen to be significantly faster. As the number of contacts n varied by a factor of 4 (from 8 to 32), the average compute times (in msec) varied from 6.18 to 400.75 (*standard*), 2.55 to 26.63 (*structural*), and 1.08 to 4.58 (*reduced*). These variations are roughly compatible with the expected complexities of $O(n^3)$, $O(n^2)$, and $O(n)$, respectively.

The next set of tests was performed on a variety of ten different single body contact situations (again $m = 1$), similar to those shown in Fig. 1: block in corner, ball resting in the rim of a hole, block in groove, face on surface, etc., with the number of contacts in the problem varying from 1 to 16. Results are shown in Fig. 7, with Fig. 8 showing a close-up scatter plot of the results for the *reduced* method. The complexity of the *reduced* method

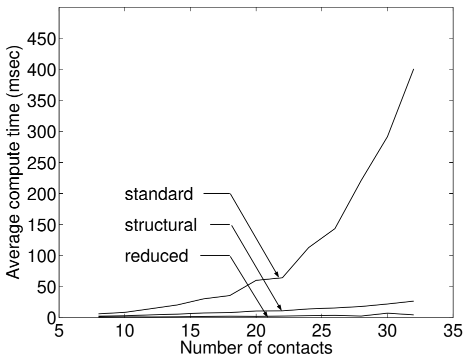

Figure 5: Average compute times for peg-in-hole by three different methods.

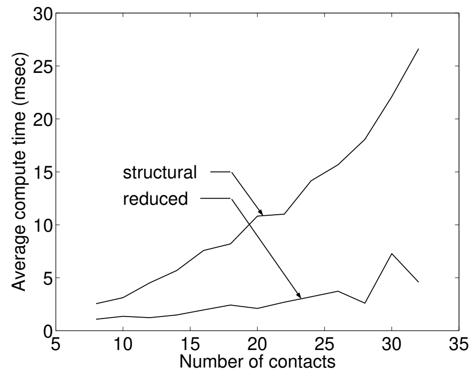

Figure 6: Close-up of our *structural* and *reduced* results shown in Fig. 5.

appears to be just a little greater than $O(n)$. As n varied from 8 to 16, the computation time varied from 0.57 to 1.34. Over the same range, the other methods varied from 5.2 to 50.9 (*standard*) and 2.1 to 12.5 (*structural*), again loosely compatible with $O(n^3)$ and $O(n^2)$.

The last set of tests were for multiple bodies, and computed the reaction forces for a stack of blocks, subject to gravity, with a random wrench applied to the top block (Fig. 9). Ten configurations were tested, with the number of blocks m ranging from 1 to 5, and the number of contacts n from 4 to 32, depending on the positioning of the blocks with respect to each other. The compute times, shown in Fig. 10, varied from 2.1 to 2136 (*standard*), 1.45 to 224 (*structural*), and 1.32 to 96 (*reduced*). These ranges are roughly compatible, modulo a constant multiplier, with the expected complexities of $O(n^3)$, $O(n^2 + nm^2)$, and $O(nm + m^3)$.

5.5 Conclusions

We have developed a fast method for solving rigid body contact problems, with friction, based on Lemke's algorithm for solving LCPs. In essence, the method exploits the sparsity of the factors of $\mathbf{M}$ to reduce the problem in size to one that is largely bounded by the number of degrees of freedom in the system. This amounts to a reduction in expected complexity from $O(n^3)$ down to nearly $O(nm + m^3)$, where n and m are the number of contacts and

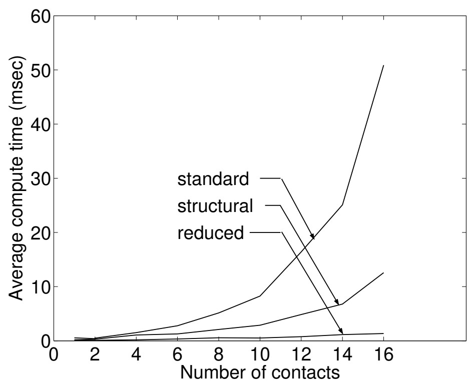

Figure 7: Average compute times for 10 different single body problems with different numbers of contacts.

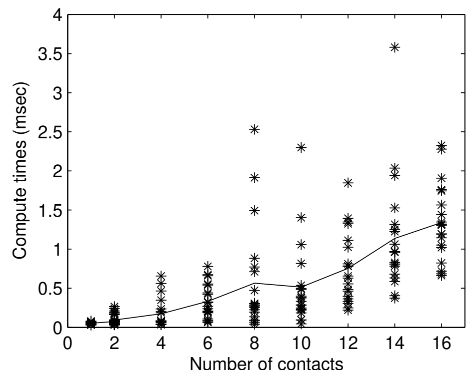

Figure 8: Close up scatter plot of the *reduced* results shown in Fig. 7.

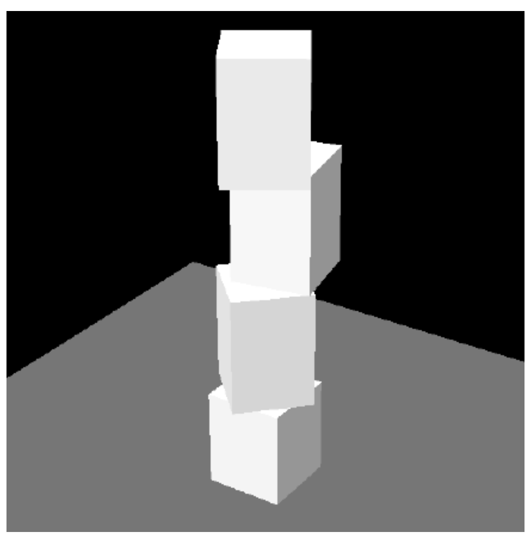

Figure 9: Stacked blocks used for the multi-body test.

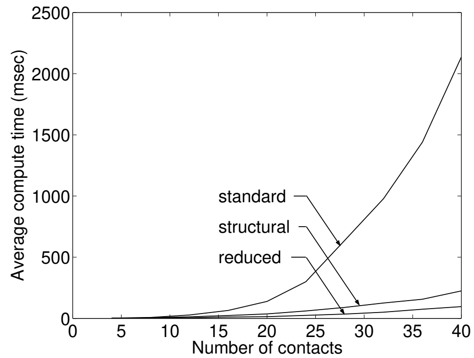

Figure 10: Average compute times for 10 different stacked block problems with different numbers of contacts.

bodies in the system. When m is fixed, the expected complexity is then close to $O(n)$. These results should improve the utility of rigid body simulation in real-time and interactive applications.

Our method also improves numerical robustness by decreasing the size of the primary linear system that must be solved during each pivot step. This system, (31), is bounded in size by $6m$, and so if m is small enough (e.g., 3 or 4) it may be solved directly in real-time using numerically robust methods, without having to do factorization updates in which round-off error can accumulate. Our method also eliminates the need to calculate the system matrix $\mathbf{M}$, whose size can be quite large even for simple problems.

Acknowledgment

This work was made possible with funding from IRIS, the Canadian Institute for Robotics and Intelligent Systems.

References

[AP97] Mihai Anitescu and Florian A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14(3):231–247, November 1997.

[AP02] Mihai Anitescu and Florian A. Potra. A time-stepping method for stiff multibody dynamics with contact and friction. *International Journal for Numerical Methods in Engineering*, 55(7):753–784, November 2002.

[Bar94] David Baraff. Fast contact force computation for nonpentrating rigid bodies. In *Proceedings of SIGGRAPH 94*, pages 23–34, July 1994.

[GBF03] Eran Guendelman, Robert Bridson, and Ronald Fedkiw. Nonconvex rigid bodies with stacking. In *Proceedings of SIGGRAPH 2003*, pages 871–878, July 2003.

[GMSW87] Philip E. Gill, Walter Murray, Michael A. Saunders, and Margaret H. Wright. Maintaining lu factors of a general sparse matrix. *Linear Algebra and its Applications*, 88/89:239–270, 1987.

[Lac03] Claude Lacoursiere. Splitting methods for dry frictional contact problems in rigid multibody systems: Preliminary performance results. In *SIGRAD2003, The Annual SIGRAD Conference*, Umea, Sweden, November 2003.

[MC95] Brian Mirticj and John Canny. Impulse-based simulation of rigid bodies. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 181–188, Monterey, California, April 1995.

[Mur88] Katta G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Helderman-Verlag, 1988.

[RWC92] Richard E. Stone Richard W. Cottle, Jong-Shi Pang. *The Linear Complementarity Problem*. Academic Press, 1992.

[Sar78] Roger W. H. Sargent. An efficient implementation of the lemke algorithm and its extension to deal with upper and lower bounds. *Mathematical Programming Study*, 7:36–54, 1978.

[SS98] Joerg Sauer and Elmar Schoemer. A constraint-based approach to rigid body dynamics for virtual reality applications. In *Proceedings of ACM Symposium on VR Software and Technology 98*, pages 153–161, Taipai, Taiwan, November 1998.

[ST96] David E. Stewart and Jeffrey C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal for Numerical Methods in Engineering*, 39(15):2673–2691, August 1996.

[Ste00] David E. Stewart. Rigid body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, March 2000.

[Tom78] J. A. Tomlin. Robust implementation of lemke's method for the linear complementarity problem. *Mathematical Programming Study*, 7:55–60, 1978.

A Simplifications for Almost Complementary Bases

In this appendix we show how to compute $\mathbf{m}'$ and $\mathbf{q}'$ for the almost complementary bases produced by Lemke's algorithm.

We begin by partitioning the augmented system (17) into the form

$$\begin{pmatrix}\mathbf{w}_{\tilde{\alpha}}\\ \mathbf{w}_{\tilde{\beta}}\end{pmatrix} = \begin{pmatrix}\bar{\mathbf{M}}_{\tilde{\alpha}\alpha} & \bar{\mathbf{M}}_{\tilde{\alpha}\beta}\\ \bar{\mathbf{M}}_{\tilde{\beta}\alpha} & \bar{\mathbf{M}}_{\tilde{\beta}\beta}\end{pmatrix}\begin{pmatrix}\mathbf{z}_{\alpha}\\ \mathbf{z}_{\beta}\end{pmatrix} + \begin{pmatrix}\mathbf{q}_{\tilde{\alpha}}\\ \mathbf{q}_{\tilde{\beta}}\end{pmatrix}. \tag{34}$$

As mentioned in Section 4, the index sets α and $\tilde{\alpha}$ for an almost complementary basis are identical, except that α also contains 0 (for the variable z_0), and $\tilde{\alpha}$ contains an extra index r (for the non-complementary variable w_r). Similarly, the index sets β and $\tilde{\beta}$ are identical, except that $\tilde{\beta}$ has one *less* index r.

This implies that $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}$, $\bar{\mathbf{M}}_{\tilde{\beta}\alpha}$, etc. have almost the same structure as the complementary components described in (23), except that $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}$ and $\bar{\mathbf{M}}_{\tilde{\beta}\alpha}$ also contain $\mathbf{c}$, and a row corresponding to w_r is shifted from $\bar{\mathbf{M}}_{\tilde{\beta}\alpha}$ and $\bar{\mathbf{M}}_{\tilde{\beta}\beta}$ to $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}$ and $\bar{\mathbf{M}}_{\tilde{\alpha}\beta}$. Arranging the $\mathbf{c}$ column on the left and the w_r column on the bottom, the components have the following form:

$$\bar{\mathbf{M}}_{\tilde{\alpha}\alpha} = \begin{pmatrix}\mathbf{N}_{\alpha}^T\mathbf{N}_{\alpha} & \mathbf{N}_{\alpha}^T\mathbf{D}_{\alpha} & 0 & \\ \mathbf{D}_{\alpha}^T\mathbf{N}_{\alpha} & \mathbf{D}_{\alpha}^T\mathbf{D}_{\alpha} & \mathbf{E}_{\alpha\alpha} & \begin{pmatrix}\mathbf{c}_{\alpha}\end{pmatrix}\\ \boldsymbol{\mu}_{\alpha\alpha} & -\mathbf{E}_{\alpha\alpha}^T & 0 & \\ \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & c_r\end{pmatrix}, \quad \bar{\mathbf{M}}_{\tilde{\alpha}\beta} = \begin{pmatrix}\mathbf{N}_{\alpha}^T\mathbf{N}_{\beta} & \mathbf{N}_{\alpha}^T\mathbf{D}_{\beta} & 0\\ \mathbf{D}_{\alpha}^T\mathbf{N}_{\beta} & \mathbf{D}_{\alpha}^T\mathbf{D}_{\beta} & \mathbf{E}_{\alpha\beta}\\ \boldsymbol{\mu}_{\alpha\beta} & -\mathbf{E}_{\beta\alpha}^T & 0\\ \mathbf{r}_4 & \mathbf{r}_5 & \mathbf{r}_6\end{pmatrix},$$

$$\bar{\mathbf{M}}_{\tilde{\beta}\alpha} = \begin{pmatrix}\mathbf{N}_{\tilde{\beta}}^T\mathbf{N}_{\alpha} & \mathbf{N}_{\tilde{\beta}}^T\mathbf{D}_{\alpha} & 0 & \\ \mathbf{D}_{\tilde{\beta}}^T\mathbf{N}_{\alpha} & \mathbf{D}_{\tilde{\beta}}^T\mathbf{D}_{\alpha} & \mathbf{E}_{\tilde{\beta}\alpha} & \begin{pmatrix}\mathbf{c}_{\beta}\end{pmatrix}\\ \boldsymbol{\mu}_{\tilde{\beta}\alpha} & -\mathbf{E}_{\alpha\tilde{\beta}}^T & 0 & \end{pmatrix}, \quad \bar{\mathbf{M}}_{\tilde{\beta}\beta} = \begin{pmatrix}\mathbf{N}_{\tilde{\beta}}^T\mathbf{N}_{\beta} & \mathbf{N}_{\tilde{\beta}}^T\mathbf{D}_{\beta} & 0\\ \mathbf{D}_{\tilde{\beta}}^T\mathbf{N}_{\beta} & \mathbf{D}_{\tilde{\beta}}^T\mathbf{D}_{\beta} & \mathbf{E}_{\tilde{\beta}\beta}\\ \boldsymbol{\mu}_{\tilde{\beta}\beta} & -\mathbf{E}_{\beta\tilde{\beta}}^T & 0\end{pmatrix},$$

$$\mathbf{q}_{\tilde{\alpha}} = \begin{pmatrix}\mathbf{N}_{\alpha}^T\mathbf{k}_x\\ \mathbf{D}_{\alpha}^T\mathbf{k}_x\\ 0\\ q_r\end{pmatrix}, \quad \mathbf{q}_{\tilde{\beta}} = \begin{pmatrix}\mathbf{N}_{\tilde{\beta}}^T\mathbf{k}_x\\ \mathbf{D}_{\tilde{\beta}}^T\mathbf{k}_x\\ 0\end{pmatrix} \tag{35}$$

$\mathbf{N}_{\alpha}$, $\mathbf{D}_{\alpha}$, $\mathbf{N}_{\beta}$, and $\mathbf{D}_{\beta}$ are the same submatrices of $\mathbf{N}$ and $\mathbf{D}$ described in Section 5.1, formed from columns corresponding to the basic and non-basic $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ variables. $\mathbf{N}_{\tilde{\beta}}$ and $\mathbf{D}_{\tilde{\beta}}$ are submatrices whose columns correspond to the basic variables of $\boldsymbol{\nu}$ and $\boldsymbol{\sigma}$ (and so will be identical to $\mathbf{N}_{\beta}$ and $\mathbf{D}_{\beta}$ except

w_r	$\mathbf{r}_1$	$\mathbf{r}_2$	$\mathbf{r}_3$	$\mathbf{r}_4$	$\mathbf{r}_5$	$\mathbf{r}_6$	q_r
ν	$\mathbf{n}_r^T\mathbf{N}_\alpha$	$\mathbf{n}_r^T\mathbf{D}_\alpha$	0	$\mathbf{n}_r^T\mathbf{N}_\beta$	$\mathbf{n}_r^T\mathbf{D}_\beta$	0	$\mathbf{n}_r^T\mathbf{k}_x$
σ	$\mathbf{d}_r^T\mathbf{D}_\alpha$	$\mathbf{d}_r^T\mathbf{D}_\alpha$	$\mathbf{e}_{r\alpha}$	$\mathbf{d}_r^T\mathbf{N}_\beta$	$\mathbf{d}_r^T\mathbf{D}_\beta$	$\mathbf{e}_{r\beta}$	$\mathbf{d}_r^T\mathbf{k}_x$
γ	$\boldsymbol{\mu}_{r\alpha}$	$-\mathbf{e}_{\alpha r}^T$	0	$\boldsymbol{\mu}_{r\beta}$	$-\mathbf{e}_{\beta r}^T$	0	0

Table 1: Values of $\mathbf{r}_1$-$\mathbf{r}_6$ and q_r depending on whether w_r is a $\boldsymbol{\nu}$, $\boldsymbol{\sigma}$, or $\boldsymbol{\gamma}$ variable.

for the possible omission of a column associated with w_r). $\mathbf{E}_{ab}$ and $\boldsymbol{\mu}_{ab}$ again denote submatrices of $\mathbf{E}$ and $\boldsymbol{\mu}$ formed from rows and columns indexed (with respect to $\mathbf{M}$) by the index sets a and b.

The subrows $\mathbf{r}_1$-$\mathbf{r}_6$ are formed from the row of $\mathbf{M}$ corresponding to w_r, and c_r and q_r are the corresponding elements of $\mathbf{c}$ and $\mathbf{q}$. The values of $\mathbf{r}_1$-$\mathbf{r}_6$ and q_r depend on whether w_r is a $\boldsymbol{\nu}$, $\boldsymbol{\sigma}$, or $\boldsymbol{\gamma}$ variable, as depicted in Table 1. There, $\mathbf{n}_r$ and $\mathbf{d}_r$ are columns of $\mathbf{N}$ or $\mathbf{D}$ corresponding to w_r; $\boldsymbol{\mu}_{r\alpha}$, $\boldsymbol{\mu}_{r\beta}$, $\mathbf{e}_{r\alpha}$, and $\mathbf{e}_{r\beta}$ are rows of $\boldsymbol{\mu}_{\beta\alpha}$, $\boldsymbol{\mu}_{\beta\beta}$, $\mathbf{E}_{\beta\alpha}$, and $\mathbf{E}_{\beta\beta}$ corresponding to w_r; and $\mathbf{e}_{\alpha r}$ and $\mathbf{e}_{\beta r}$ are columns of $\mathbf{E}_{\beta\alpha}$ and $\mathbf{E}_{\beta\beta}$ corresponding to w_r.

As with complementary bases, the computation of $\mathbf{q}'$ and $\mathbf{m}'$ is described by (19)-(21) and involves solving a system of the form $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}\mathbf{x} = \mathbf{b}$, and then computing $\bar{\mathbf{M}}_{\tilde{\beta}\alpha}\mathbf{x}$. The details of solving $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}\mathbf{x} = \mathbf{b}$ depend on whether w_r is a $\boldsymbol{\nu}$, $\boldsymbol{\sigma}$, or $\boldsymbol{\gamma}$ variable, and these three cases are analyzed in the subsections below.

The details of solving $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}\mathbf{x} = \mathbf{b}$ depend on whether w_r is a $\boldsymbol{\nu}$, $\boldsymbol{\sigma}$, or $\boldsymbol{\gamma}$ variable, and these three cases are analyzed in the subsections below.

For $\bar{\mathbf{M}}_{\tilde{\beta}\alpha}\mathbf{x}$, the calculation is essentially the same as (33), except for an additional term due to the $\tilde{\beta}$ component of the covering vector, and the fact that $\mathbf{v}_c$ is now defined by (40), described below, rather than (29):

$$\bar{\mathbf{M}}_{\tilde{\beta}\alpha}\mathbf{x} = \begin{pmatrix} \mathbf{N}_{\tilde{\beta}}^T \\ \mathbf{D}_{\tilde{\beta}}^T \\ 0 \end{pmatrix} \mathbf{v}_c + \begin{pmatrix} 0 \\ \mathbf{E}_{\tilde{\beta}\alpha}\boldsymbol{\lambda}_\alpha \\ \boldsymbol{\mu}_{\tilde{\beta}\alpha}\boldsymbol{\theta}_\alpha - \mathbf{E}_{\alpha\tilde{\beta}}^T\boldsymbol{\phi}_\alpha \end{pmatrix} + \mathbf{c}_{\tilde{\beta}} z_0. \tag{36}$$

All these computations can be done with the same complexity as described in Section 5.1.

A.1 Solving $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}\mathbf{x} = \mathbf{b}$ when w_r is a $\boldsymbol{\nu}$ variable

We begin with the structure of $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}$ in (35). Since each column of $\mathbf{E}_{\alpha\alpha}$ must still contain at least one non-zero entry (Appendix C), it can again be partitioned as in (25). Hence, as in Section 5.1, we can again partition $\boldsymbol{\phi}_\alpha$ and $\mathbf{D}_\alpha$ into $\boldsymbol{\phi}_\kappa$, $\boldsymbol{\phi}_x$, $\mathbf{D}_\kappa$ and $\mathbf{D}_x$. Given the values of $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ when w_r is a $\boldsymbol{\nu}$ variable (Table 1), and generalizing our use of $\boldsymbol{\theta}_\alpha$, $\boldsymbol{\phi}_\alpha$, $\boldsymbol{\lambda}_\alpha$, and z_0 to denote the components of $\mathbf{x}$, we can then expand $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}\mathbf{x} = \mathbf{b}$ into a form analagous to (26):

$$\begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{N}_\alpha & \mathbf{N}_\alpha^T\mathbf{D}_\kappa & \mathbf{N}_\alpha^T\mathbf{D}_x & 0 & \mathbf{c}_\nu \\ \mathbf{D}_\kappa^T\mathbf{N}_\alpha & \mathbf{D}_\kappa^T\mathbf{D}_\kappa & \mathbf{D}_\kappa^T\mathbf{D}_x & \mathbf{E}_{\kappa\kappa} & \mathbf{c}_\kappa \\ \mathbf{D}_x^T\mathbf{N}_\alpha & \mathbf{D}_x^T\mathbf{D}_\kappa & \mathbf{D}_x^T\mathbf{D}_x & \mathbf{I} & \mathbf{c}_x \\ \boldsymbol{\mu}_{\alpha\alpha} & -\mathbf{E}_{\kappa\kappa}^T & -\mathbf{I} & 0 & \mathbf{c}_\gamma \\ \mathbf{n}_r^T\mathbf{N}_\alpha & \mathbf{n}_r^T\mathbf{D}_\kappa & \mathbf{n}_r^T\mathbf{D}_x & 0 & c_r \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\kappa \\ \boldsymbol{\phi}_x \\ \boldsymbol{\lambda}_\alpha \\ z_0 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_\nu \\ \mathbf{b}_\kappa \\ \mathbf{b}_x \\ \mathbf{b}_\gamma \\ b_r \end{pmatrix}. \tag{37}$$

As in Section 5.1, we can solve for and eliminate $\boldsymbol{\phi}_x$:

$$\boldsymbol{\phi}_x = \boldsymbol{\mu}_{\alpha\alpha}\boldsymbol{\theta}_\alpha - \mathbf{E}_{\kappa\kappa}^T\boldsymbol{\phi}_\kappa + \mathbf{c}_\gamma z_0 - \mathbf{b}_\gamma, \tag{38}$$

Then defining

$$\mathbf{c}_* \equiv \mathbf{D}_x\mathbf{c}_\gamma \tag{39}$$

and redefining $\mathbf{v}_c$ as

$$\mathbf{v}_c \equiv \mathbf{N}_*\boldsymbol{\theta}_\alpha + \mathbf{D}_*\boldsymbol{\phi}_\kappa - \mathbf{b}_* + \mathbf{c}_* z_0, \tag{40}$$

we can solve for and eliminate $\boldsymbol{\lambda}_\alpha$:

$$\boldsymbol{\lambda}_\alpha = -\mathbf{D}_x^T\mathbf{v}_c + \mathbf{b}_x - \mathbf{c}_x z_0, \tag{41}$$

leaving us with the reduced system

$$\begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{N}_* & \mathbf{N}_\alpha^T\mathbf{D}_* & \mathbf{c}_\nu + \mathbf{N}_\alpha^T\mathbf{c}_* \\ \mathbf{D}_*^T\mathbf{N}_* & \mathbf{D}_*^T\mathbf{D}_* & \mathbf{c}'_\kappa + \mathbf{D}_*^T\mathbf{c}_* \\ \mathbf{n}_r^T\mathbf{N}_* & \mathbf{n}_r^T\mathbf{D}_* & c_r + \mathbf{n}_r^T\mathbf{c}_* \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\kappa \\ z_0 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_\nu \\ \mathbf{b}'_\kappa \\ b_r \end{pmatrix} + \begin{pmatrix} \mathbf{N}_\alpha^T \\ \mathbf{D}_*^T \\ \mathbf{n}_r^T \end{pmatrix} \mathbf{b}_* \tag{42}$$

where $\mathbf{c}'_\kappa \equiv \mathbf{c}_\kappa - \mathbf{E}_{\kappa\kappa}\mathbf{c}_x$. This system is then solved directly for $\boldsymbol{\theta}_\alpha$, $\boldsymbol{\phi}_\kappa$, and z_0, after which we back-solve for $\boldsymbol{\phi}_x$ and $\boldsymbol{\lambda}_\alpha$.

A.2 Solving $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}\mathbf{x} = \mathbf{b}$ when w_r is a $\boldsymbol{\sigma}$ variable

Given the values of $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ when w_r is a $\boldsymbol{\sigma}$ variable (Table 1), we can expand $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}\mathbf{x} = \mathbf{b}$ into a form similar to (37):

$$\begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{N}_\alpha & \mathbf{N}_\alpha^T\mathbf{D}_\kappa & \mathbf{N}_\alpha^T\mathbf{D}_x & 0 & \mathbf{c}_\nu \\ \mathbf{D}_\kappa^T\mathbf{N}_\alpha & \mathbf{D}_\kappa^T\mathbf{D}_\kappa & \mathbf{D}_\kappa^T\mathbf{D}_x & \mathbf{E}_{\kappa\kappa} & \mathbf{c}_\kappa \\ \mathbf{D}_x^T\mathbf{N}_\alpha & \mathbf{D}_x^T\mathbf{D}_\kappa & \mathbf{D}_x^T\mathbf{D}_x & \mathbf{I} & \mathbf{c}_x \\ \boldsymbol{\mu}_{\alpha\alpha} & -\mathbf{E}_{\kappa\kappa}^T & -\mathbf{I} & 0 & \mathbf{c}_\gamma \\ \mathbf{d}_r^T\mathbf{N}_\alpha & \mathbf{d}_r^T\mathbf{D}_\kappa & \mathbf{d}_r^T\mathbf{D}_x & \mathbf{e}_{r\alpha} & c_r \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\kappa \\ \boldsymbol{\phi}_x \\ \boldsymbol{\lambda}_\alpha \\ z_0 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_\nu \\ \mathbf{b}_\kappa \\ \mathbf{b}_x \\ \mathbf{b}_\gamma \\ b_r \end{pmatrix}. \tag{43}$$

As is Section A.1, we can solve for and eliminate $\boldsymbol{\phi}_x$ and $\boldsymbol{\lambda}$ using (38) amd (41), which leads to the following reduced system,

$$\begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{N}_* & \mathbf{N}_\alpha^T\mathbf{D}_* & \mathbf{c}_\nu + \mathbf{N}_\alpha^T\mathbf{c}_* \\ \mathbf{D}_*^T\mathbf{N}_* & \mathbf{D}_*^T\mathbf{D}_* & \mathbf{c}'_\kappa + \mathbf{D}_*^T\mathbf{c}_* \\ \mathbf{d}_*^T\mathbf{N}_* & \mathbf{d}_*^T\mathbf{D}_* & c'_r + \mathbf{d}_*^T\mathbf{c}_* \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\kappa \\ z_0 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_\nu \\ \mathbf{b}'_\kappa \\ b'_r \end{pmatrix} + \begin{pmatrix} \mathbf{N}_\alpha^T \\ \mathbf{D}_*^T \\ \mathbf{d}_*^T \end{pmatrix} \mathbf{b}_* \tag{44}$$

where

$$\mathbf{d}_* \equiv \mathbf{d}_r - \mathbf{D}_x\mathbf{e}_{r\alpha}^T, \quad c'_r \equiv c_r - \mathbf{e}_{r\alpha}\mathbf{c}_x, \quad b'_r \equiv b_r - \mathbf{e}_{r\alpha}\mathbf{b}_x \tag{45}$$

This system is then solved directly for $\boldsymbol{\theta}_\alpha$, $\boldsymbol{\phi}_\kappa$, and z_0, after which we back-solve for $\boldsymbol{\phi}_x$ and $\boldsymbol{\lambda}_\alpha$.

A.3 Solving $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}\mathbf{x} = \mathbf{b}$ when w_r is a $\boldsymbol{\gamma}$ variable

Given the values of $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ when w_r is a $\boldsymbol{\gamma}$ variable (Table 1), $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}\mathbf{x} = \mathbf{b}$ takes the form

$$\begin{pmatrix} \mathbf{N}_\alpha^T\mathbf{N}_\alpha & \mathbf{N}_\alpha^T\mathbf{D}_\kappa & \mathbf{N}_\alpha^T\mathbf{D}_x & 0 & \mathbf{c}_\nu \\ \mathbf{D}_\kappa^T\mathbf{N}_\alpha & \mathbf{D}_\kappa^T\mathbf{D}_\kappa & \mathbf{D}_\kappa^T\mathbf{D}_x & \mathbf{E}_{\kappa\kappa} & \mathbf{c}_\kappa \\ \mathbf{D}_x^T\mathbf{N}_\alpha & \mathbf{D}_x^T\mathbf{D}_\kappa & \mathbf{D}_x^T\mathbf{D}_x & \mathbf{I} & \mathbf{c}_x \\ \boldsymbol{\mu}_{\alpha\alpha} & -\mathbf{E}_{\kappa\kappa}^T & -\mathbf{I} & 0 & \mathbf{c}_\gamma \\ \boldsymbol{\mu}_r & -\mathbf{e}_{\kappa r}^T & 0 & 0 & c_r \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\kappa \\ \boldsymbol{\phi}_x \\ \boldsymbol{\lambda}_\alpha \\ z_0 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_\nu \\ \mathbf{b}_\kappa \\ \mathbf{b}_x \\ \mathbf{b}_\gamma \\ b_r \end{pmatrix} \tag{46}$$

where $\mathbf{e}_{\kappa r}$ is a subvector of $\mathbf{e}_{\alpha r}$ whose elements are associated with $\boldsymbol{\phi}_\kappa$ (the subvector of $\mathbf{e}_{\alpha r}$ corresponding to $\boldsymbol{\phi}_x$ must be 0 since it is beneath $\mathbf{I}$ and any column of $\mathbf{E}^T$ can contain at most one non-zero entry).

Again, we can solve for and eliminate $\boldsymbol{\phi}_x$ and $\boldsymbol{\lambda}$ using (38) and (41), and obtain the following reduced system,

$$\begin{pmatrix} \mathbf{N}_\alpha^T \mathbf{N}_* & \mathbf{N}_\alpha^T \mathbf{D}_* & \mathbf{c}_\nu + \mathbf{N}_\alpha^T \mathbf{c}_* \\ \mathbf{D}_*^T \mathbf{N}_* & \mathbf{D}_*^T \mathbf{D}_* & \mathbf{c}'_\kappa + \mathbf{D}_*^T \mathbf{c}_* \\ \boldsymbol{\mu}_r & -\mathbf{e}_{\kappa r}^T & c_r \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\kappa \\ z_0 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_\nu \\ \mathbf{b}'_\kappa \\ b_r \end{pmatrix} + \begin{pmatrix} \mathbf{N}_\alpha^T \\ \mathbf{D}_*^T \\ 0 \end{pmatrix} \mathbf{b}_* \tag{47}$$

which is then solved directly for $\boldsymbol{\theta}_\alpha$, $\boldsymbol{\phi}_\kappa$, and z_0.

B Columns of $\mathbf{E}_{\alpha\alpha}$ have at most two non-zero entries

Each of the friction directions $\hat{\mathbf{d}}_{ij}$ associated with a particular contact i lie in the tangent plane of that contact and hence can be expressed as a combination of two unit basis vectors $\hat{\mathbf{d}}_x$ and $\hat{\mathbf{d}}_y$,

$$\hat{\mathbf{d}}_{ij} = \cos(a_j)\hat{\mathbf{d}}_x + \sin(a_j)\hat{\mathbf{d}}_y \tag{48}$$

where a_j is an angle giving the direction of $\hat{\mathbf{d}}_{ij}$. Now let $\mathbf{d}_{ij}$ denote the columns of $\mathbf{D}$ associated with contact i. Since the $\mathbf{d}_{ij}$ are formed from the $\hat{\mathbf{d}}_{ij}$ by a linear transformation, (48) carries through to $\mathbf{d}_{ij}$, so that

$$\mathbf{d}_{ij} = \cos(a_j)\mathbf{d}_x + \sin(a_j)\mathbf{d}_y \tag{49}$$

where $\mathbf{d}_x$ and $\mathbf{d}_y$ are basis columns produced from $\hat{\mathbf{d}}_x$ and $\hat{\mathbf{d}}_y$.

Now, for any given basis, we will have $\mathbf{M}_{\tilde{\alpha}\alpha}\mathbf{z}_\alpha = -\mathbf{q}_{\tilde{\alpha}}$. For a complementary basis, this can be expressed by (24) with $\mathbf{b} = \mathbf{q}_{\tilde{\alpha}}$:

$$\begin{pmatrix} \mathbf{N}_\alpha^T \mathbf{N}_\alpha & \mathbf{N}_\alpha^T \mathbf{D}_\alpha & 0 \\ \mathbf{D}_\alpha^T \mathbf{N}_\alpha & \mathbf{D}_\alpha^T \mathbf{D}_\alpha & \mathbf{E}_{\alpha\alpha} \\ \boldsymbol{\mu}_{\alpha\alpha} & -\mathbf{E}_{\alpha\alpha}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta}_\alpha \\ \boldsymbol{\phi}_\alpha \\ \boldsymbol{\lambda}_\alpha \end{pmatrix} = - \begin{pmatrix} \mathbf{N}_\alpha^T \mathbf{k}_x \\ \mathbf{D}_\alpha^T \mathbf{k}_x \\ 0 \end{pmatrix}. \tag{50}$$

The second block row of this system implies that

$$\mathbf{D}_\alpha^T(\mathbf{N}_\alpha\boldsymbol{\theta}_\alpha + \mathbf{D}_\alpha\boldsymbol{\phi}_\alpha + \mathbf{k}_x) + \mathbf{E}_{\alpha\alpha}\boldsymbol{\lambda}_\alpha = 0, \tag{51}$$

which reduces to

$$\mathbf{D}_\alpha^T\mathbf{v}_t + \mathbf{E}_{\alpha\alpha}\boldsymbol{\lambda}_\alpha$$

where $\mathbf{v}_t \equiv \mathbf{N}_\alpha \boldsymbol{\theta}_\alpha + \mathbf{D}_\alpha \boldsymbol{\phi}_\alpha + \mathbf{k}_x$. This means that for every non-basic variable σ_{ij} associated with a particular contact i, we have

$$\mathbf{d}_{ij}^T \mathbf{v}_t + \lambda_i = 0, \tag{52}$$

where $\mathbf{d}_{ij}$ is the column of $\mathbf{D}$ corresponding to σ_{ij}. For an almost complementary basis, we obtain the analogous result

$$\mathbf{d}_{ij}^T \mathbf{v}_t + \lambda_i + c_{ij} z_0 = 0, \tag{53}$$

where c_{ij} is the element of the covering vector $\mathbf{c}$ corresponding to σ_{ij}. If we assume that for a given i, c_{ij} is the same for all j, then both (52) and (53) can be expressed in the form

$$\mathbf{d}_{ij}^T \mathbf{v}_t = -p_i \tag{54}$$

for some fixed value p_i. Combining this with (49) then gives

$$q_x \cos(a_j) + q_y \sin(a_j) = -p_i \tag{55}$$

where $q_x = \mathbf{d}_x^T \mathbf{v}_t$ and $q_y = \mathbf{d}_y^T \mathbf{v}_t$ are again fixed for a given i. However, (55) can have at most two real-valued solutions for a_j, as can be verified using the tangent half-angle substitution

$$\cos(a_j) = \frac{1 - z^2}{1 + z^2}, \quad \sin(a_j) = \frac{2z}{1 + z^2} \tag{56}$$

and observing that the resulting equation is a quadratic in z. Therefore, for a particular contact i, equation (52) or (53) can be satisfied for at most two different $\mathbf{d}_{ij}$, which means that at most two σ_{ij} variables can be non-basic, and hence each column of $\mathbf{E}_{\alpha\alpha}$ can have at most two non-zero entries.

C Columns of $\mathbf{E}_{\alpha\alpha}$ have at least one non-zero entry

Here we show that $\mathbf{E}_{\alpha\alpha}$ will never contain a zero column.

For a complementary basis, $\mathbf{M}_{\tilde{\alpha}\alpha}$ has the form shown in (23), and so a zero-column in $\mathbf{E}_{\alpha\alpha}$ would imply a zero column in $\mathbf{M}_{\tilde{\alpha}\alpha}$, which cannot happen since Lemke's algorithm (and pivoting methods in general) guarantees that $\mathbf{M}_{\tilde{\alpha}\alpha}$ is never singular.

For an almost complementary basis, suppose $\mathbf{E}_{\alpha\alpha}$ contains a zero column, denoted by $\mathbf{e}_{\alpha r}$. That column must then correspond to a λ_i which is basic. At the same time, no ϕ_{ij} for that contact can be basic, because for any basic ϕ_{ij} there would be a corresponding non-basic σ_{ij} and hence a non-zero element in $\mathbf{e}_{\alpha r}$. Since λ_i is basic, γ_i must be non-basic, and so the augmented system contains the equation (taken from the fourth block row of (37), (43), or (46))

$$\gamma_i = \mu_i \theta_i - \sum_{j=1}^{d} \phi_{ij} + c_i z_0 = 0$$

where c_i is the element of the covering vector $\mathbf{c}$ corresponding to γ_i. Since all ϕ_{ij} are non-basic and hence 0, this reduces to

$$\mu_i \theta_i + c_i z_0 = 0.$$

But this is impossible since $\mu_i \geq 0$, $\theta_i \geq 0$, $c_i > 0$ (by construction), and $z_0 > 0$ (since otherwise Lemke's algorithm would have terminated with a solution). Hence $\mathbf{E}_{\alpha\alpha}$ cannot contain a zero column.

D Extending the covering vector

Here we describe how to extend the covering vector $\mathbf{c}$ when the contact LCP is expanded as described in Section 5.3. When extending $\mathbf{c}$, the new elements $\mathbf{c}_\delta$ must be assigned values such that for the current basis and the expanded value of $\mathbf{q}$, denoted by $\mathbf{q}^x$, we still have $\mathbf{q}^{x\prime} \geq 0$. We do this by first setting the $\mathbf{c}_\delta$ to some arbitrary positive value (typically 1), and then increase them if it turns out that $\mathbf{q}' \not\geq 0$.

To compute $\mathbf{q}^{x\prime}$, we need the expanded versions of $\bar{\mathbf{M}}_{\tilde{\alpha}\alpha}$ and $\bar{\mathbf{M}}_{\tilde{\beta}\alpha}$, which we will denote by $\bar{\mathbf{M}}^x_{\tilde{\alpha}\alpha}$ and $\bar{\mathbf{M}}^x_{\tilde{\beta}\alpha}$. Notationally, new rows will be placed at the bottom. Because the basic $\mathbf{z}$ variables are identical for the original and expanded systems, we have

$$\bar{\mathbf{M}}^x_{\tilde{\alpha}\alpha} = \bar{\mathbf{M}}_{\tilde{\alpha}\alpha} \quad \text{and} \quad \bar{\mathbf{M}}^x_{\tilde{\beta}\alpha} = \begin{pmatrix} \bar{\mathbf{M}}_{\tilde{\beta}\alpha} \\ \bar{\mathbf{M}}_{\delta\beta} \end{pmatrix}. \tag{57}$$

where $\bar{\mathbf{M}}_{\delta\beta}$ gives the new rows of $\bar{\mathbf{M}}^x_{\tilde{\beta}\alpha}$. Likewise, for $\mathbf{q}^x$, we have ${\mathbf{q}^x}_{\tilde{\alpha}} = \mathbf{q}_{\tilde{\alpha}}$ and

$${\mathbf{q}^x}_{\tilde{\beta}} = \begin{pmatrix} \mathbf{q}_{\tilde{\beta}} \\ \mathbf{q}_\delta \end{pmatrix} \tag{58}$$

where $\mathbf{q}_\delta$ are the new values. Then from (19) we obtain

$$\mathbf{q}^{x\prime} = \begin{pmatrix} \mathbf{q}' \\ \mathbf{q}_\delta + \bar{\mathbf{M}}_{\delta\beta}\mathbf{q}'_{\tilde{\alpha}} \end{pmatrix}.$$

Now, since Lemke's algorithm guarantees $\mathbf{q}' \geq 0$ at each step, in checking whether $\mathbf{q}^{x\prime} \geq 0$, we only need to be concerned about $\mathbf{q}_\delta + \bar{\mathbf{M}}_{\delta\beta}\mathbf{q}'_{\tilde{\alpha}}$, which we will designate as $\mathbf{q}_\delta{}'$. If $\mathbf{q}_\delta{}' \geq 0$, all is well and the new elements of the covering vector can be left unchanged. Otherwise, we utilize the fact that (for an almost complementary basis) one of the columns of $\bar{\mathbf{M}}_{\delta\beta}$ is $\mathbf{c}_\delta$, and its corresponding element in $\mathbf{q}'_{\tilde{\alpha}}$ is z_0. If we then increase each element of $\mathbf{c}_\delta$ by some Δc such that

$$\Delta c > -\min\left(\mathbf{q}_\delta{}'/z_0\right) \tag{59}$$

the resulting new value for $\mathbf{q}_\delta{}'$ will satisfy $\mathbf{q}_\delta{}' > 0$.