Computational Geometry: Theory and Applications 7:25-35, 1997.

Rational Orthogonal Approximations to Orthogonal Matrices

Victor J. Milenkovic*
University of Miami
Department of Mathematics and Computer Science
P.O. Box 249085, Coral Gables, FL 33124
vjm@cs.miami.edu

Veljko Milenkovic
Engineering Consultant†
1222 Highway 42
Sturgeon Bay, WI 54235

Abstract

Several algorithms are presented for approximating an orthogonal rotation matrix M in three dimensions by an orthogonal matrix with rational entries. The first algorithm generates an approximation $M_2(M, \epsilon)$ with accuracy ϵ and $2b+2$-bit numerators and a common $2b+2$-bit denominator (bit-size $2b+2$), where $b = \lceil -\lg \epsilon \rceil$ ($\epsilon \approx 2^{-b}$). The second algorithm uses basis reduction to generate an approximation $M_\nu(M, \epsilon)$ with accuracy $\epsilon^{\nu/1.5}$ and bit-size νb for some $1.5 \leq \nu \leq 6$ (but ν cannot be controlled except by trial and error). A third algorithm, based on integer programming, generates optimal $M_{\text{opt}}(M, \epsilon)$ with accuracy ϵ and bit-size proven to be no more than $1.5b$. In practice, the second algorithm generates an approximation with $\nu \approx 1.5$ and is much faster than the third algorithm. The best bit-sizes which one could obtain using previously known results in two dimensions [1] are more than $3b$ bits for numerator and denominator. Applications are described for the approximation functions in the area of solid modeling.

1 Introduction

Certain numerical issues must be resolved in order to implement an algorithm of computational geometry as a computer program. The implementation can use exact or rounded arithmetic. If rounded arithmetic is used, it is necessary to deal with topological inconsistencies and numerical error. Exact arithmetic does not present these problems, but it has, in general, much higher cost than rounded arithmetic.

Let us suppose we choose to use exact arithmetic. It is desirable that all operations be within the field of rational numbers. This is difficult to accomplish for constructions or computations involving rotations, such as solids modeling or robotic path planning. Orthogonal rotation matrices[1], often defined in terms of Euler angles ψ, ϕ, θ,

$$M(\psi,\phi,\theta) = \begin{bmatrix} \cos\psi\cos\phi - \sin\psi\cos\theta\sin\phi & -\cos\psi\sin\phi - \sin\psi\cos\theta\cos\phi & \sin\psi\sin\theta \\ \sin\psi\cos\phi + \cos\psi\cos\theta\sin\phi & -\sin\psi\sin\phi + \cos\psi\cos\theta\cos\phi & -\cos\psi\sin\theta \\ \sin\theta\sin\phi & \sin\theta\cos\phi & \cos\theta \end{bmatrix}, \quad (1)$$

*Supported by NSF grants NSF-CCR-91-157993 and NSF-CCR-90-09272

†Formerly affiliated with Ford Motor Company.

[1]Matrix M is orthogonal if $M^t = M^{-1}$. These are sometimes called *orthonormal*.

have entries which are irrational and non-algebraic except for very special choices of input angles. What is desired is the following: a function $M_{\text{approx}}(M, \epsilon)$ which takes an arbitrary orthogonal matrix M and accuracy $\epsilon > 0$ as input and returns an orthogonal matrix that has rational entries and which approximates M to within ϵ:

$$\max_{|v|=1} |(M - M_{\text{approx}})v| \leq \epsilon. \tag{2}$$

At the ACM geometry conference last year, Canny, Donald, and Ressler [1] presented a technique for generating rational two dimensional rotation matrices. For any two dimensional rotation matrix M, their technique can generate a rational matrix with accuracy 2^{-b} and rational entries with b-bit numerators and a common b-bit denominator (bit-size b). This is the worst-case behavior of their technique. Given a desired rotation angle θ, they use the standard rational parameterization of the unit circle, $\sin\theta = \frac{2t}{1+t^2}$ and $\cos\theta = \frac{1-t^2}{1+t^2}$, where $t = \tan(\theta/2)$. They then apply a variation of known techniques for finding the solution to the *best approximation problem*[2] (see [10][8]) to find the best rational approximation for t with accuracy ϵ. This leads to the best rational approximation to M. Since their technique generates angles with rational sines and cosines, it can be used to create rational three-dimensional orthogonal matrices through the use of Equation 1 above. However, this results in an increase in bit-size to roughly $3b$ numerator and denominator, which is far from the best possible.

In this paper, we present a technique based on quaternion arithmetic for directly generating three dimensional rational orthogonal matrices. This technique reduces the matrix approximation problem to that of finding simultaneous rational approximations, p_1/p_0, p_2/p_0, p_3/p_0 to real numbers $\alpha_1, \alpha_2, \alpha_3$. We prove that the best simultaneous approximation leads to the best rational matrix approximation. A naive approximation, setting $p_0 = 2^b$ and finding the best p_1, p_2, p_3, yields $M_2(M, \epsilon)$ with bit-size $2b + 2$ and accuracy $\epsilon = 2^{-b}$.

Using the *basis reduction* algorithm of Lenstra, Lenstra, and Lovasz [10] [9], we describe a method of approximating α_i by p_i/p_0, $i = 1, 2, 3$ with a "smaller sized" p_0 for a given accuracy. However, it does not give as good a control on the accuracy attained. For any given $\eta > 0$, it finds an integer $p_0 \leq 8\eta^{-3}$ such that there is an approximation to M with accuracy η/p_0 and bit-size $2\lceil \lg p_0 \rceil$. If we set $\eta = \epsilon^{1/4}$ and if p_0 is near its upper bound, then the accuracy will be ϵ and the bit-size close to $1.5b$ for $b = -\lg \epsilon$. If p_0 is small, we do not attain the desired accuracy. In the worst case, we have to set $\eta = \epsilon$ and then we obtain accuracy ϵ^4 (much closer than we wanted) and bit-size $6b$. Thus the approximation is $M_\nu(M, \epsilon)$ with accuracy $\epsilon^{\nu/1.5}$ and bit-size νb for some $1.5 \leq \nu \leq 6$. This method generates good approximations in practice, because it takes only a few tries to find p_0 near its upper bound. This technique yields accuracies of $\epsilon = 10^{-6}$ in a few seconds on a 30 MIPS workstation running Maple. Using specially coded routines for basis reduction in C, one might perform the necessary basis reduction one or two orders of magnitude faster. The basis reduction algorithm runs in strongly polynomial time.

Finally, we prove that the optimal approximation $M_{\text{opt}}(M, \epsilon)$ has bit-size at most $1.5b$, and we show how to apply integer programming to find an approximation $M_{\text{opt}}(M, \epsilon)$ within one bit of optimal. This can be solved in polynomial time using an algorithm of Lovasz and Scarf [11] which has been implemented by Cook, *et al* [3]. We plan to run tests of the running time, but it is probably most practical to use the older basis reduction method.

We report on our experiment with one simple basis reduction heuristic. Creating an approximation to 10^{-6} accuracy requires about 0.01 second on a DEC Alpha 3000/700. The heuristic performs close to the theoretical optimum.

[2]Given a number $\alpha \in \mathbf{Q}$ and an $\epsilon \in \mathbf{Q}$, $\epsilon > 0$, find a rational number p/q such that $q > 0$, $|\alpha - p/q| < \epsilon$ and q is as small as possible.

1.1 Applications

Currently, there is no reasonable way to implement a computer system that can model polyhedral objects. Such a system would have half-spaces (such as $\{(x, y, z) \mid x \geq 0\}$) as primitives and would allow at least the following operations:

- regularized set operations: union, intersection, complement, difference;
- Euclidean transformations: translation, rotation, scaling;
- convex hull.

There are no known robust algorithms for implementing such a system in rounded floating point arithmetic. Actually, one can describe combinatorially consistent algorithms for these operations, but there are imaginable cases for these algorithms which have unbounded numerical error.

If one implements such a system using exact rational arithmetic in a standard fashion, then there are sequences of operations which have exponential growth in bit-complexity. For instance, intersecting three polyhedra might generate a vertex which is the intersection of three faces. Taking the convex hull will generate a face containing three vertices. Generating a point from three planes and generating a plane from three points in each case trebles the number of bits in the representation. This example is clearly not a proof, but we expect that for any exact algorithm, the bit-complexity will grow exponentially in the worst case.

We see the design of a practical solid modeler as consisting of three parts: fast integer arithmetic for geometric constructions, rational Euclidean transformations, and geometric rounding.

It is commonly understood that geometric constructions depend on the *signs* of arithmetic expression on the input coordinates, not the actual values. This fact is what makes the use of rounded arithmetic so perilous: the sign function has infinite relative error for inputs near zero. On the other hand, there are ways to compute the sign of integer expressions which are practically and/or theoretically more efficient than determining the actual value. Karasick, Lieber and Nackman [7], Clarkson [2], and Fortune and Van Wyk [5] have given results in this area, of which the latter is perhaps the most practical.

This paper describes how to obtain rational orthogonal matrices in three dimensions. From these, arbitrary rational Euclidean transformations can be constructed.

Even if multiple-precision operations can be made relatively cheap, some means must be developed for overcoming the exponential growth in the bit-complexity of the polyhedra. The first author and Nackman [12] have shown that finding the minimum-perturbation rounding of the coordinates to lower precision, without changing the combinatorial structure, is an NP-complete problem. However, this author has proposed techniques [13] [14] that round to lower precision and change the combinatorial structure in a reasonable way. The first author and Nackman are also working on heuristics for rounding without changing the combinatorial structure.

The type of combinatorial structure that the system must preserve depends on the application. Some applications may rely greatly on the preservation of combinatorial structure. Suppose an application needs the system to model a sequence of cutting, drilling, assembly, and other manufacturing operations. Certain rotations arise in the application. As a result of using approximations to these rotations, the output of the system will not exactly represent reality. However, if the approximate matrices are strictly orthogonal, the output is guaranteed to be physically realizable. This consideration justifies the present research into techniques to generate such matrices.

As has been shown, all applications must eventually apply rounding to the geometric structure. However, the designer is free to postpone rounding until the consistency of the result has been verified. Thus it is hoped that by combining different techniques, one could offer a solution to the solid modeling problem, which the field of computational geometry has so far failed to provide.

2 Algorithms for Constructing Rational Orthogonal Approximations

Our goal is to construct a rational rotation matrix M_{approx} which approximates an arbitrary rotation matrix M. This section shows how to reduce this problem to that of finding a good simultaneous rational approximation $p_1/p_0, p_2/p_0, p_3/p_0$ to three real values $\alpha_1, \alpha_2, \alpha_3$. It is also shown that the best approximation to the α's yields the best approximation matrix M_{approx}.

2.1 Quaternion Arithmetic

This section reviews quaternion arithmetic and discusses how it can be used to represent orthogonal transformations.

It is commonly known that a rotation in $\mathbf{R}^2$ can be represented as multiplication by a complex number of unit length. If we represent a point $v = (v_x, v_y)$ as a complex number $v_x + v_y i$, then $(\cos\theta + \sin\theta i)(v_x + v_y i)$ is the complex representation of the result of rotating v about the origin by angle θ.

Hamilton [6] introduced the quaternion algebra as a generalization of complex numbers that can be used to represent rotations in $\mathbf{R}^3$. The quaternion group $\mathcal{Q} = \{\pm 1, \pm i, \pm j, \pm k\}$ is a non-commutative group of order eight whose elements satisfy,

$$i^2 = j^2 = k^2 = ijk = -1,$$

$$ij = k = -ji, \qquad jk = i = -kj, \qquad ki = j = -ik.$$

A quaternion is a linear combination of the elements of $\mathcal{Q}$ over $\mathbf{R}$: $Q = a + bi + cj + dk$. The conjugate of a quaternion is defined by $\overline{Q} = a - bi - cj - dk$ and satisfies an identity analogous to the ordinary complex conjugate:

$$Q\overline{Q} = \overline{Q}Q = a^2 + b^2 + c^2 + d^2.$$

We can represent a point $v = (v_x, v_y, v_z) \in \mathbf{R}^3$ as a "pure imaginary" quaternion,

$$\hat{v} = v_x i + v_y j + v_z k.$$

For a given angle θ and a given unit vector $u \in \mathbf{R}^3$, we define,

$$q(\theta, u) = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}\hat{u} = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}u_x i + \sin\frac{\theta}{2}u_y j + \sin\frac{\theta}{2}u_z k. \tag{3}$$

The product $q(\theta, u)\hat{v}\overline{q(\theta, u)}$ is the quaternion representation $\hat{v'}$ of a point $v' \in \mathbf{R}^3$ which is the result of rotating v counterclockwise by angle θ about the axis defined by u (see [15]). Note that $q(2\pi - \theta, -u) = -q(\theta, u)$ represents the same rotation as $q(\theta, u)$. Thus the mapping from unit quaternions to $SO(3)$ (orthogonal transformations with determinant 1) is two-to-one.

Unlike the case with complex numbers, the quaternion representation of $SO(3)$ leads directly to a technique for generating orthogonal matrices with rational entries. Let $Q = Q_0 + Q_1 i + Q_2 j + Q_3 k$ be an arbitrary non-zero quaternion with integer components Q_0, Q_1, Q_2, Q_3. The unit quaternion $q = |Q|^{-1}Q = q_0 + q_1 i + q_2 j + q_3 k$ is not rational, but the matrix of the transformation which q represents is rational. A theorem by Rodriguez [4] shows that this matrix is

$$(Q_0^2+Q_1^2+Q_2^2+Q_3^2)^{-1}\left[\begin{array}{ccc} Q_0^2+Q_1^2-Q_2^2-Q_3^2 & 2(Q_1Q_2-Q_0Q_3) & 2(Q_1Q_3+Q_0Q_2) \\ 2(Q_2Q_1+Q_0Q_3) & Q_0^2-Q_1^2+Q_2^2-Q_3^2 & 2(Q_2Q_3-Q_0Q_1) \\ 2(Q_3Q_1-Q_0Q_2) & 2(Q_3Q_2+Q_0Q_1) & Q_0^2-Q_1^2-Q_2^2+Q_3^2 \end{array}\right] =$$

$$\left[\begin{array}{ccc} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_2q_1+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_3q_1-q_0q_2) & 2(q_3q_2+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{array}\right]. \tag{4}$$

We note that if the components of Q are b'-bit integers, then the entries of the matrix will have $b = 2b' + 2$ bits in the numerator and a common b-bit denominator.

2.2 Constructing $Q(M)$

The first task in the approximation of an orthogonal transformation M is the construction of a quaternion $Q(M)$ which generates M. If the axis u and the rotation angle θ are known, then Equation 3 yields $q(M)$ directly. If M is given as a matrix $(r_{ij})_{1\le i,j\le 3}$, then the following can be shown from Equation 4 (see [16]):

$$\left[\begin{array}{cccc} 1+r_{11}+r_{22}+r_{33} & r_{32}-r_{23} & r_{13}-r_{31} & r_{21}-r_{12} \\ r_{32}-r_{23} & 1+r_{11}-r_{22}-r_{33} & r_{12}+r_{21} & r_{13}+r_{31} \\ r_{13}-r_{31} & r_{12}+r_{21} & 1-r_{11}+r_{22}-r_{33} & r_{23}+r_{32} \\ r_{21}-r_{12} & r_{13}+r_{31} & r_{23}+r_{32} & 1-r_{11}-r_{22}+r_{33} \end{array}\right] =$$

$$4\left[\begin{array}{cccc} q_0^2 & q_0q_1 & q_0q_2 & q_0q_3 \\ q_1q_0 & q_1^2 & q_1q_2 & q_1q_3 \\ q_2q_0 & q_2q_1 & q_2^2 & q_2q_3 \\ q_3q_0 & q_3q_1 & q_3q_2 & q_3^2 \end{array}\right] = 4\left[\begin{array}{c} q_0 \\ q_1 \\ q_2 \\ q_3 \end{array}\right]\left[\begin{array}{cccc} q_0 & q_1 & q_2 & q_3 \end{array}\right], \tag{5}$$

where $q = q_0+q_1i+q_2j+q_3k$ is a unit quaternion that generates M. As one can see, every row is proportional to the components q. In case we are using floating point arithmetic, it is important to select the row which contains the maximum diagonal element $4q_i^2$. We call the row selected in this fashion $Q(M)$. By this selection, one avoids using a degenerate (all-zero) row. This computation is numerically accurate in floating point since $Q(M) = 4q_iq$, where $|4q_i| \ge 2$. The absolute error of 3μ per component, where μ is the rounding unit, is converted into an absolute error, $3\mu/(4q_i)$, after normalization. This error is bounded by 1.5μ.

Implementation note: If we define $r_{00} = r_{11} + r_{22} + r_{33}$, then each diagonal element can be written $1 + 2r_{ii} - r_{00}$, $i = 0, 1, 2, 3$. Hence, determining the maximum diagonal element is equivalent to finding $\max\{r_{00}, r_{11}, r_{22}, r_{33}\}$.

2.3 Generating an Approximation

The following lemma tells us how close we must approximate a unit quaternion q so that the corresponding matrix has accuracy ϵ.

Lemma 2.1 *Let q and q' be unit quaternions, and let q' approximate q to within $\epsilon/2$:*

$$|\Delta q| = |q' - q| \le \epsilon/2.$$

Then $M(q')$ approximates $M(q)$ to within $\epsilon + \mathrm{O}(\epsilon^2)$.

Proof. Using Equation 2:

$$\begin{aligned}
\max_{|v|=1} |(M - M_2)v| &= \max_{|v|=1} |q\hat{v}\overline{q} - q_2\hat{v}\overline{q_2}|, \\
&= \max_{|v|=1} |q\hat{v}\overline{q} - (q + \Delta q)\hat{v}(\overline{q} + \overline{\Delta q})|, \\
&= \max_{|v|=1} |\Delta q\hat{v}\overline{q} + q\hat{v}\overline{\Delta q} + \Delta q\hat{v}\overline{\Delta q}|, \\
&\leq \max_{|v|=1} (|\Delta q\hat{v}\overline{q}| + |q\hat{v}\overline{\Delta q}| + |\Delta q\hat{v}\overline{\Delta q}|), \\
&= \epsilon + \frac{\epsilon^2}{4}.
\end{aligned}$$

The term of order ϵ^2 is negligible. □

Using this lemma, it is easy to generate a $2b+2$-bit ϵ-approximation to any orthogonal matrix M. First generate $Q(M)$ and then scale it so it has length e^{-1}. Approximate each component by the nearest integer to generate $Q' = (p_0 + p_1 i + p_2 j + p_3 k)$. If we let q and q' be the unitized versions of Q and Q', they will satisfy the lemma, and thus $M(Q')$ will be the approximation we desire.

It is more general to reduce the approximation to that of finding three simultaneous rational approximations as follows. Define $Q_u(M) = Q(M)/Q_i(M)$, where $Q_i(M)$ is the largest magnitude component of $Q(M)$. Suppose, without loss of generality, that $i = 0$, and thus the components of $Q_u(M)$ are $(1, \alpha_1, \alpha_2, \alpha_3)$. We need to find integers p_1, p_2, p_3, p_0 such that,

$$|\alpha_i - \frac{p_i}{p_0}| \leq \frac{\epsilon}{2\sqrt{3}} \qquad i = 1, 2, 3. \tag{6}$$

It is easily shown that setting $Q_{\text{approx}} = p_0 + p_1 i + p_2 j + p_3 k$ and then generating the corresponding matrix M_{approx} also satisfies Lemma 2.1. Accuracy ϵ for M_{approx} is equivalent to accuracy ϵ for $Q_{\text{approx}}/|Q_{\text{approx}}|$ and is equivalent to accuracy $\epsilon/2\sqrt{3}$ for the approximation to $\alpha_1, \alpha_2, \alpha_3$.

2.4 Finding the Best Approximation

The following two lemmas show that the best (smallest bit-size) rational approximations p_1/p_0, p_2/p_0, p_3/p_0 to real numbers $\alpha_1, \alpha_2, \alpha_3$ with accuracy $\epsilon/2\sqrt{3}$ have bit-size at most $0.75b$ ($b = -\lg \epsilon$ as usual), and this leads to a $1.5b$-bit approximation to a matrix M. Furthermore, the best rational matrix approximation can be found by determining the best rational approximation $\alpha_1, \alpha_2, \alpha_3$ with accuracy $\epsilon/2\sqrt{3}$. Therefore, the optimal approximation $M_{\text{opt}}(M, \epsilon)$ to M with with accuracy ϵ has bit-size at most $1.5b$.

Lemma 2.2 *Given* $-1 \leq \alpha_1, \alpha_2, \alpha_3 \leq 1$ *and* $\epsilon/2\sqrt{3} \approx 2^{-b}$, *there exists* $0.75b$*-bit integers* p_1, p_2, p_3, p_0 *such that* $|\alpha_i - p_i/p_0| \leq \epsilon/2\sqrt{3}$, $i = 1, 2, 3$.

Proof. Ignoring the factor of $2\sqrt{3}$ for simplicity (it changes things by at most two bits), for $-2^{0.75b} \leq p_1, p_2, p_3, p_0 \leq 2^{0.75b}$, there are roughly 2^{3b} lattice points of the form $(p_1 - p_0\alpha_1, p_2 - p_0\alpha_2, p_3 - p_0\alpha_3)$ inside a cube of size $2^{0.75b}$. Therefore two lattice points must lie distance $2^{-0.25b}$ apart (under L_∞ norm). Their difference is also a lattice point. Dividing through by p_0 yields a point of the form $(p_1/p_0 - \alpha_1, p_2/p_0 - \alpha_2, p_3/p_0 - \alpha_3)$ at distance 2^{-b} from the origin. Thus the resulting p_1, p_2, p_3, p_0 satisfy the theorem. □

These $0.75b$-bit p's yield a $1.5b$-bit approximation matrix. We can turn this around to show that a good approximation matrix implies a small bit-size integer $Q(M)$.

Lemma 2.3 *A rational 2b-bit sized rotation matrix M_{approx} corresponds to a b-bit or smaller sized integer quaternion Q_{approx}.*

Proof. Given a rational matrix, we can clear fractions and solve for $2q_0, 2q_1, 2q_2, 2q_3$, using Equation 5 above (the 1's on the left are replaced by the common denominator in the matrix). Each $2q_i$ is the square root of an integer: $2q_i = Q_i\sqrt{m_i}$, where m_i is square-free for $i = 0, 1, 2, 3$. However, the fact that $4q_iq_j$ is integral for each pair implies that $m_0 = m_1 = m_2 = m_3$. Therefore, the quaternion $Q_0 + Q_1 i + Q_2 j + Q_3 k$ generates the rational matrix by Equation 4, and the Q's have at most half the bit-size of the entries of M. □

Comment: What if the rational matrix does not have a b-bit *common* denominator, as this proof requires? The answer is that rotating an integer vector by the matrix (which is the eventual application) requires that the entries be adjusted to have a common denominator anyway so that the fractions can be summed. In other words, it is only fair to compute the bit-size of a rotation matrix if its entries have a common denominator.

3 Generating Simultaneous Rational Approximations

We have reduced the problem of finding an approximating rational rotation matrix to that of finding rational approximations with a common denominator to three real values $\alpha_1, \alpha_2, \alpha_3$. It is easy to generate an ϵ-accurate ($\epsilon \approx 2^{-b}$) set of rationals by setting the denominator $p_0 = \lceil \epsilon^{-1} \rceil$ and choosing the best numerators p_1, p_2, p_3. The resulting approximation matrix $M_2(M, \epsilon)$ has bit-size $2b+2$. This section explores algorithms for finding smaller bit-size approximations. The first algorithm using basis reduction yields a $1.5b$-bit matrix in practice, but is not guaranteed to find the best approximation. A second method, based on integer programming, can find a near optimal approximation. Both algorithms run in polynomial time, but the first is probably the most practical.

3.1 Approximation by Basis Reduction

Basis reduction is a process that finds a "short" set of basis vectors for a lattice. To find an approximation to $\alpha_1, \alpha_2, \alpha_3$, one reduces the basis,

$$(1,0,0,0),\ \ (0,1,0,0),\ \ (0,0,1,0),\ \ (\alpha_1, \alpha_2, \alpha_3, -x),$$

where x is a parameter chosen to control the accuracy and bit-size of the approximation. Each reduced basis vector is some integer combination, $(p_1 - p_0\alpha_1, p_2 - p_0\alpha_2, p_3 - p_0\alpha_3, p_0 x)$ of the four original vectors. Assuming one can find a reduced basis vector such that the first three components are roughly equal to or smaller than the fourth component in magnitude, then $|p_i/p_0 - \alpha_i| \leq x$. One can set x equal to the desired accuracy $\epsilon/2\sqrt{3}$. Incidentally, it is required that $\alpha_1, \alpha_2, \alpha_3$ be rational, so that one must replace them with rational approximations $\alpha_1^{\text{rat}}, \alpha_2^{\text{rat}}, \alpha_3^{\text{rat}}$. It is easy to generate these rational approximations using continued fractions since they do not have to have a common denominator. If $|\alpha_i^{\text{rat}} - \alpha_i| \leq \epsilon/4\sqrt{3}$ for $i = 1, 2, 3$, we must set $x = \epsilon/4\sqrt{3}$ to obtain the desired accuracy.

The Lenstra, Lestra, Lovasz [9] algorithm for basis reduction is a part of MAPLE and other mathematical packages. It is not known how to determine the value of x that leads to the smallest p_0. However, in practice after a few tries with slightly different x, one can generate a good approximation.

As far as guaranteed behavior, Lovasz [10] indicates that basis reduction leads to a polynomial-time algorithm that takes $\eta > 0$ and a rational vector $(\alpha_1, \alpha_2, \ldots \alpha_n)$ as input and returns integers $p_1, p_2, \ldots, p_n, p_0$ such that $|\alpha_j - \dfrac{p_j}{p_0}| \leq \dfrac{\eta}{p_0}$ for $j = 1, 2, \ldots, n$, and $0 < p_0 \leq 2^{n(n+1)/4}\eta^{-n}$.

For $n = 3$, $0 < p_0 \le 8\eta^{-3}$, and for some $\eta \in \left[\frac{\epsilon}{4\sqrt{3}}, \left(\frac{4}{3}\right)^{\frac{1}{8}} \epsilon^{\frac{1}{4}}\right]$, Lovasz's algorithm will return a value of p_0 such that $\frac{\eta}{p_0} \le \frac{\epsilon}{4\sqrt{3}}$. Setting $\eta = \epsilon/4\sqrt{3}$ will always work, but may result in $3b$-bit p_0. It is best if η is as large as possible so that p_0 is as small as possible. In practice, one finds solutions for $\eta \approx \epsilon^{\frac{1}{4}}$ and thus p_0 has bit-size about $0.75b$.

3.2 Finding an Optimal Approximation

For any given accuracy ϵ, a near optimal[3] set of p_1, p_2, p_3, p_0 can be found in polynomial time using integer programming. Let ϵ' be a close rational approximation to $\epsilon/4\sqrt{3}$. It is sufficient to solve,

$$-\epsilon' p_0 \le p_i - \alpha_i^{\mathrm{rat}} p_0 \le \epsilon' p_0, \qquad i = 1, 2, 3,$$

for integers p_1, p_2, p_3 and minimum integer p_0. Recent results by Cook *et al* [3] show that integer programs of this type can be solved for up to 100 variables using a polynomial-time algorithm of Lovasz and Scarf [11], and we plan to test the practicality of finding optimal approximations with this system.

3.3 Computational Experiments

We performed some experiments with a simple basis reduction heuristic. To find rational approximations to $\alpha_1, \alpha_2, \alpha_3$, we set up a matrix,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & 1 \end{bmatrix}.$$

We treat the first three columns as four vectors in $\mathbf{R}^3$. (In other words, ignore the last column.) We repeatedly choose the vector with largest magnitude (under L_∞ norm). The we express it as a linear combination of the other vectors, by solving a system of equations. Then we round the coefficients of the linear combination to the nearest integers. Finally, we perform a row reduction on the matrix using the rounded coefficients.

If the coefficients had not been rounded, then the first three entries of that row would become zero. Since the coefficients have been rounded, the first three entries become much nearer to zero than before. In other words, the vector in $\mathbf{R}^3$ is much smaller. Now we repeat with the current largest vector.

The termination condition is as follows. The error of row i is

$$\epsilon = \frac{\max_{j=1,2,3} |a_{ij}|}{|a_{i4}|}.$$

When the error of some row i is less than the desired accuracy, in our case 10^{-6}, then a_{i4} equals the desired denominator p_0. Given $\alpha_1, \alpha_2, \alpha_3$, and p_0, it is simple to compute the corresponding p_1, p_2, p_3.

We ran this on 1000 randomly chosen inputs $\alpha_1, \alpha_2, \alpha_3$. On average, it required 7.6 iterations to converge to one part in a million accuracy. This required on average about 0.01 second on a DEC Alpha 3000/700.

[3] *Near* optimal because the use of rational approximations might lead to a solution that is one bit larger than actual optimum.

For each result we computed $\log p_0$ and $-\log \epsilon$, where ϵ is the error. The theory predicts (Lemma 2.2)

$$-\log \epsilon \approx \frac{4}{3} \log p_0.$$

For the experiment we ran, average value of $-\log \epsilon$ is 22.0034. The average value of $(4/3) \log p_0$ is 21.9858. Thus, this heuristic is only slightly worst than the theoretical optimum.

4 Conclusion

If one is willing to use a few seconds of running time on a typical modern workstation, then the basis reduction scheme is a practical method for finding close $1.5b$-bit rational approximations to 3D rotation matrices. If not, a larger $2b$-bit approximation can be found using a few dozen floating point operations.

Acknowledgements: The authors would like to thank Steve Fortune for suggesting Lemma 2.3. We would also like to thank Laslo Lovasz, Herbert Scarf, William Cook, and Jeff Lagarias for their helpful information and suggestions.

References

[1] J. Canny, B. Donald, E.K. Ressler. A Rational Rotation Method for Robust Geometric Algorithms. *Proceedings of the Eighth Symposium on Computational Geometry*, ACM, pages 251–160, June 1992.

[2] K.L. Clarkson. "Safe and Effective Determinant Evaluation." *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, IEEE, Oct. 24-27, 1992, pp. 387–395.

[3] W. Cook, T. Rugherford, H. E. Scarf and D. Shallcross. An Implementation of the Generalized Basis Reduction Algorithm for Integer Programming, Cowles Foundation Discussion Paper No. 990, Yale University, August 1991.

[4] E. Ehrhart. "Solution to Advanced Problems." *The American Mathematical Monthly*, Vol. 87, No. 10, Dec. 1980, p. 826.

[5] S. Fortune, C. Van Wyk. "Efficient Exact Arithmetic for Computational Geometry", manuscript, 1992.

[6] W.R. Hamilton. *Elements of Quaternions*, Chelsea Publishing Cl., N.Y. 1969.

[7] M. Karasick, D. Lieber, and L. Nackman. *Efficient Delaunay Triangulation using Rational Arithmetic.* IBM Research Report RC 14455, IBM Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, To appear in 1989.

[8] A. Khintchine. *Continued Fractions*, Moscow-Leningrad, 1935.

[9] A.D.Lenstra, H.W. Lenstra, Jr., L. Lovasz. Factoring Polynomials with Rational Coefficients. Mathematical Analysis 261, pp. 515–534, 1982.

[10] L. Lovasz. *An Algorithmic Theory of Numbers, Graphs and Convexity*, CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1986.

[11] L. Lovasz and H. E. Scarf. The Generalized Basis Reduction Algorithm, Cowles Foundation Paper No. 818, Yale University, 1992.

[12] V. J. Milenkovic and L. R. Nackman. "Finding Compact Coordinate Representations for Polygons and Polyhedra." *IBM Journal of Research and Development*, vol. 34, no. 35, September 1990, pp. 753–769.

[13] "Rounding Face Lattices in the Plane." *First Canadian Conference on Computational Geometry*, Montreal, Quebec, Canada, August 21-25, 1989 (abstract).

[14] "Rounding Face Lattices in d Dimensions." *Proceedings of the Second Canadian Conference on Computational Geometry*, Jorge Urrutia, Ed., University of Ottawa, Ontario, August 6-10, 1990, pp. 40–45.

[15] J. Rooney. A Survey of Representations of Spatial Rotation about a Fixed Point. *Environment and Planning B*, Vol. 4, 1977, pp. 185-210.

[16] S.W. Sheppard. *Quaternion from Rotation Matrix*, The Charles Stark Draper Laboratory, Inc., Cambridge, MA, 1978. NASA Contract NAS 9-13809.