1-12-2023 3:00 PM

Pythagorean Vectors and Rational Orthonormal Matrices

Aishat Olagunju, *The University of Western Ontario*

Supervisor: Jeffrey, David, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Applied Mathematics

Abstract

A Pythagorean vector is an integer vector having an integer 2-norm. Such vectors are closely related to Pythagorean n-tuples, since n-tuples are the building blocks for Pythagorean vectors. Pythagorean vectors are, in their turn, the building blocks for rational orthonormal matrices. The work in this thesis has a pedagogical application to the QR decomposition of matrices, widely used in Linear Algebra. A barrier for students learning the details of the QR decomposition of a given matrix A is the occurrence of square-roots that cannot be simplified during the application of the two standard algorithms, namely the Gram–Schmidt method and Householder transformations. This thesis studies Pythagorean vectors and their application to the construction of exercises and test questions in which a given matrix A can be factored into matrices Q and R, with all arithmetic operations resulting in rational quantities, free from square roots. This freedom from square roots applies to every step of the calculations, and not just the final result.

As a preliminary to QR decomposition, the thesis explores the properties of Pythagorean vectors, including their generation for an arbitrary specified dimension. Pythagorean triples, which correspond to Pythagorean vectors of dimension 2, have been widely and enthusiastically studied in the literature, but higher dimensions have been less studied, and this thesis adds some new observations to previous studies.

Summary for Lay Audience

This thesis is based on a pedagogical application, namely the teaching of a particular topic in Linear Algebra. Courses in advanced linear algebra include a study of a process called the QR decomposition of a given matrix. The existing textbook treatments usually require students to perform extensive arithmetic operations. Matters can become more difficult for students when their calculations throw up awkward arithmetic expressions containing radicals, such as $\sqrt{168}$, which cannot be simplified and which pollute the students' working. This thesis investigates ways in which instructors may construct exercises which are guaranteed to avoid unnecessary arithmetic difficulties for students.

In addition, Pythagorean triples are a popular subject for investigations in the literature. The thesis starts by happily joining this activity with new observations on the properties of triples, and adds observations of n-tuples for $n > 3$.

Keywords: Pythagorean n-tuples, rational matrices, orthonormal matrices

Contents

List of Figures

List of Tables

List of Algorithms

List of Appendices

Chapter 1

Introduction

1.1 Background

Linear Algebra, the ”fun” branch of Mathematics whose versatility ranges from matrices to geometry to engineering, has always been widely used and recognized. It comprises linear combinations such as linear equations, linear maps, and their expressions in vector spaces through matrices. The earliest study of linear algebra arose from the study of determinants, from Leibniz’s representation of coefficients with pairs of numbers to Maclaurin’s (1729) solutions of simultaneous linear equations up to 4 unknowns. Maclaurin made mention of Cramer’s rule, which gives a general method of reducing two quadratic forms simultaneously to sums of squares.

Pythagorean triples have been known for 4000 years. In 1922, George Arthur Plimpton bought a Babylonian clay tablet dating back to around 1800 BC, for $10 from Edgar James Banks[11]. Edgar discovered the tablet now known as the Plimpton 322. It was written in a sexagesimal number system and consists of two of the three numbers now known as Pythagorean triples. There have been numerous articles and papers on Pythagorean n-tuples and how to generate them. From Euclid’s formula for generating Pythagorean triples to ternary trees method with the use of Price matrices [24] and Berggren matrices[6].

Frisch and Vaserstein show that there exists a parametrization of Pythagorean triples by a single triple of integer-valued polynomial[15]. They took it a step further and showed for $n = 4$ or 6, the Pythagorean n-tuple admits a parametrization by a single n-tuple of polynomials with integer coefficients [16]. There have also been a collection of proofs without words for Pythagorean Triples[19], Pythagorean quadruples[29], and parametric representation of primitive Pythagorean triples[3].

As for most things already existing in the world of mathematics the origin of matrices can be traced back to ancient times. Gottfried Leibniz introduced the theory of matrices by taking the coefficients of linear equations and putting them in a rectangular array. This array would later on in 1848 be called a *Matrix* which is a Latin word for *”Womb”* by James Joseph Sylvester. Sylvester didn’t only name the array, he also introduced the process of finding a determinant.

Like everything in life, having friends with similar interests can be considered a major benefit such as in the case of Arthur Cayley, Sylvester’s friend who made great contributions to the matrix theory involving scalar multiplication of matrices, matrix multiplication, addition, and also inverse matrix theory. Both men were known as 'invariant twins" for their collaboration in the development of the theory of "forms" (or "quantics") [4].

Various operations can be and have been performed on matrices, one such is the decomposition of a matrix into a product of two other matrices. From Reduced Row Echelon Form (RREF) computations using Gaussian elimination to QR decomposition, a vital topic that involves decomposing a matrix into a product of an orthogonal matrix Q, with the property that its transpose is its inverse: $Q^T Q = I$, and an upper triangle matrix R. QR decomposition has always been a popular research area due to its versatility in applications to many fields of science and engineering, from machine learning, and least squares problems to reducing the dimensionality of matrices.

In numerical linear algebra, QR has played and is still playing numerous roles. Some of which is its application to any real or complex matrix regardless of its format or structure. Square invertible matrices can also be decomposed using QR factorization. The QR factorization also reveals rank.

To compute the QR decomposition, there are two main methods, the householder transformation and the most common method, the Gram-Schmidt process. The Gram-Schmidt process has proved useful in many areas, but there have been some unfavorable articles against it. Staib in 1969 raised a couple of objections to the Gram-Schmidt process, he found the process "cumbersome" and "inelegant" and therefore presented an alternative[33], a matrix method. Hoffman in 1970 addressed Staib's objections with the paper *The Gram-Schmidt Process Is Not so Bad!*, where he compared the Gram-Schmidt method and Staib's alternative method[20].

Householder transformations are widely used in many areas such as in geometric optics, and numerical linear algebra, for tridiagonalization and has such been discussed in different contexts. In sparse matrices[21], in scalar product spaces[27] and also in solving complex symmetric eigenvalue problem[30]. Solutions to least squares problems with Gram-Schmidt and Householder transformations have also been tested for their accuracy[26].

While the Gram-Schmidt processes are common and efficient, it often falls victim to catastrophic cancellation. This is when the initial values are large, and the final values become small with relative errors. The Householder QR decomposition's significance lies in the fact that it can be used to transform a given vector into another vector with specified zero components while preserving length, thus preventing this phenomenon.

1.2 Properties and Applications of Orthonormal matrices

1.2.1 Properties of Orthonormal matrices

With Q being a matrix and u, v, x being vectors, we have the following important properties of Orthonormal matrices are described below

1. Orthonormal matrices are rotations of the coordinates. This is demonstrated by the results:

   (a) vector lengths are unchanged. That is $||Qx||_2 = ||x||_2$.

Proof: This can be easily proved by
$(||Qx||_2)^2 = (Qx)^T(Qx)$
$(Qx)^T(Qx) = x^TQ^T(Qx) = x^T(Q^TQ)x = x^Tx = (||x||_2)^2$

(b) angles between vectors are unchanged. That is $(Qu).(Qv) = u.v$.

Proof: Again this can be proved by converting to matrix notation: $u.v = u^Tv$
$Qu.Qv = (Qu)^TQv = (u^TQ^T)Qv = u^T(Q^TQ)v = u^Tv = u.v$

2. An orthonormal matrix has columns of length 1. So $Q = (q_1\ q_2\ \ldots q_n)$ has $||q_k|| = 1$ for all k. This implies that if the entries are rational, then we can put the entries over a common denominator and get a Pythagorean n-tuple or vector.

   Proof: $q = (a_1/b_1 \ldots a_n/b_n)$ then $\sum(a_i/b_i)^2 = 1$.
   Let d be LCM of the b_i, then $b_i = d/c_i$ and $a_i/b_1 = a_ic_i/d$.
   Then $\sum(a_ic_i)^2 = n^2d^2$.

The definition of Pythagorean n-tuples can be found in 2.1.

1.2.2 Applications of Orthonormal matrices

1. Least-squares approximation: This method is used for estimating the true value of some quantity based on a consideration of errors in observations of measurements. Suppose A, b are real matrices and A has linearly independent columns. Then we find x such that $Ax = b$, x which minimizes $||Ax - b||_2$ is called the least square problem.[34] Given the equation

$$Ax = b.$$

   We start off by solving for the QR decomposition of A,

$$A = QR$$

   Next, we compute the reduced QR decomposition

$$Q = [Q_1, Q_2], \quad R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

$$A = Q_1R_1$$

   Then solve for x

$$R_1x = Q_1^Tb$$

   In essence, when an orthonormal transformation is applied to a vector or matrix, the error will not grow.

2. Pedagogical application: Creating an example of lines intersecting at a given angle. The aim is to construct 2 straight lines that intersect at 45 degrees $(\pi/4)$. The vector form of a line is $r(t) = a + v * t$ where a, v are vectors and t is a parameter. Starting with two lines meeting at the origin at 45. Then we use Q matrices to rotate them to a new position. Now we have two lines $r1(t) = u * t$ and $r2(s) = v * s$. Next, we add a vector to each to move them from the origin, this gives $r1 = a + u * t$ and $r2 = a + v * t$. In this form, it is too easy for the students because $t = s = 0$ shows the lines meet at a, so, the parameters t and s are redefined so that we get $r1 = a + u * t$ and $r2 = b + v * t$.

1.3 Research Purpose

At this age, more and more students from their high school years (some earlier) develop some hatred for mathematics because they find it dull compared with other subjects that are easier to connect to. Some find that it requires too much memory capacity which is not always the case. Some mathematics problems most times require understanding more than one's ability to memorize formulas.

A relatable reason would be the repetitive solving pattern sometimes filled with mistakes that are needed to gain a better understanding of the solution process of the problem, which is seen in RREF. Making a mistake in the pivoting process of RREF would trigger an avalanche of confusion and frustration when the realization comes of the need to go back and fix the mistakes or start afresh in some cases.

As an instructor of Linear Algebra, the topic of matrix reduction can be a fun experience or a stress-filled experience, since it involves a lot of arithmetic. For students who would demand partial marks for having an idea of how the solution process should be carried out and for other students who actually did the work but encountered mistakes of irreducible roots which made it more complicated.

Having to teach and examine this topic stress and worry-free is expected and justified. Instructors could even help their students by informing them that all square roots they would encounter while solving have been constructed to simplify. Especially for exams without computer aid.

As a student learning the various methods of matrix decomposition, the rigorous steps of Reduced Row Echelon Form with Gaussian Elimination are daunting on their own not to talk of being followed by Gram Schmidt process and Householder transformations.

One of my favorite math quotes says, "Mathematics is like love, a simple idea, but it can get complicated". To avoid such further complications and to make the study of matrix decomposition a little easier and less daunting for students and teachers, the introduction of Pythagorean vectors to generate rational orthonormal matrices which when the reduction process is being performed without the computation aid would be rational. This idea is the main driving force for this research.

An outline for the purposes are

- Show that problems can be constructed by choosing Q and R and multiplying,
- Showing that *all* steps in Gram-Schmidt, Householder are rational,
- Create a public repository of Q matrices for use in applications.

Chapter 2

Pythagorean n-tuples

2.1 Pythagorean n-tuples:

Definition A Pythagorean n-tuple is a set of n positive integers x_i such that

$$\sum_{i=1}^{n-1} (x_i)^2 = (x_n)^2.$$

A *primitive* n-tuple is one in which the integers are co-prime and arranged in ascending order[1]. Since the focus is linear algebra, vectors are defined with similar properties.

Definition A vector $x = [x_1, \ldots, x_m]$ is Pythagorean if $\forall k$, $x_k \in \mathbb{Z}$ and $\|x\|_2 \in \mathbb{N}$.

This definition relaxes some of the requirements for an n-tuple: zero entries and negative entries are allowed. Later in the thesis, we count Pythagorean vectors, and for this purpose we define primitive Pythagorean vectors as having elements that are positive integers, and co-prime.

2.2 Pythagorean triple

Definition A Pythagorean triple consists of three numbers, a, b, c, which satisfy the equation $a^2 + b^2 = c^2$.

Pythagorean triples are called *primitive* when the $gcd(a,b,c) = 1$, in other words, a, b, c are co-prime.

2.3 Generating Pythagorean triples

There are numerous ways of generating Pythagorean triples, such as Euclid's method, Fibonacci's formula[14], areas proportional to the sum of squares, and the use of matrices and linear transformation to mention a few.

[1] Ascending order is not required by all authors, and there will be places in this thesis where the requirement will be abandoned.

2.3.1 Euclid's formula

The common way of generating Pythagorean triples is by using Euclid's formula which incorporates an arbitrary pair of integers m and n with $m > n > 0$ and states that

$$a = m^2 - n^2 \ , \qquad b = 2mn \ , \qquad c = m^2 + n^2 \ , \tag{2.1}$$

where (a, b, c) are Pythagorean triples. This formula generates non-primitive triples when m and n are odd and generates primitive triples if m and n are co-prime and one of them is even. While this formula generates all primitive triples it otherwise does not generate all triples. To generate such triples another parameter k is introduced;

$$a = k(m^2 - n^2) \ , \qquad b = k(2mn) \ , \qquad c = k(m^2 + n^2) \ , \tag{2.2}$$

where m, n and k are positive numbers and m and n are co-prime with $m > n$ and at least one is even.

Algorithm 1 Euclid's method

```
procedure EUCLID(max)                                   ▷ maximum number desired
    Outlist := []
    for m from 2 to max do
        for n from 1 to modp(m,2) by 2 to m do
            if igcd(m,n) = 1 then
                Outlist := [op(Outlist), [m^2 − n^2, 2nm, m^2 + n^2]]   ▷ compile the list
```

2.3.2 Fibonacci's method

Fibonacci's method uses the sequence of consecutive odd integers, $[1, 3, 5, ...]$, with the sum of the first n terms of the sequence being n^2[14]. The method starts by choosing any odd square integer k from $(k = a^2)$, making it the n^{th} term, then letting b^2 be the sum of the previous $n - 1$ terms and c^2 be the sum of all the n terms. The following algorithm depicts these steps;

Algorithm 2 Fibonacci's method

```
procedure FIBONACCI(k)                                  ▷ k is any odd square number
    n = (k + 1)/2
    a = sqrt(k)
    b = sqrt(add(seq(i, i = 1..n − 1, 2)))              ▷ sum of the n − 1 terms
    c = sqrt(add(seq(i, i = 1..n, 2)))                  ▷ sum of the n terms
    if a^2 + b^2 = c^2 then
        Return c, a, b
```

2.3.3 Ternary trees

A ternary tree is a tree data structure with at most three child nodes, in plain terms, a tree with at most three branches. Recent publications on generating Pythagorean triples include the use of ternary trees with Berggren matrices[6] or Price matrices[24]. This method involves the generation of three successors of a fixed triple a, b, c with a odd using a set of three matrices, M_1, M_2, M_3, composed of constants.

Note that all primitive Pythagorean triples can be generated with this method. This means all primitive Pythagorean triples can be given a tree-like structure with each branch a representation of multiplication by M_j.[6]

If M_j are defined

$$M_1 = \begin{bmatrix} -1 & 2 & 2 \\ -2 & 1 & 2 \\ -2 & 2 & 3 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \end{bmatrix}, \quad M_3 = \begin{bmatrix} 1 & -2 & 2 \\ 2 & -1 & 2 \\ 2 & -2 & 3 \end{bmatrix}$$

each branch in Fig 2.1 represents a matrix multiplication. An example would be $M_1(3,4,5)^T = (15,8,17)^T$ where T is the transpose.

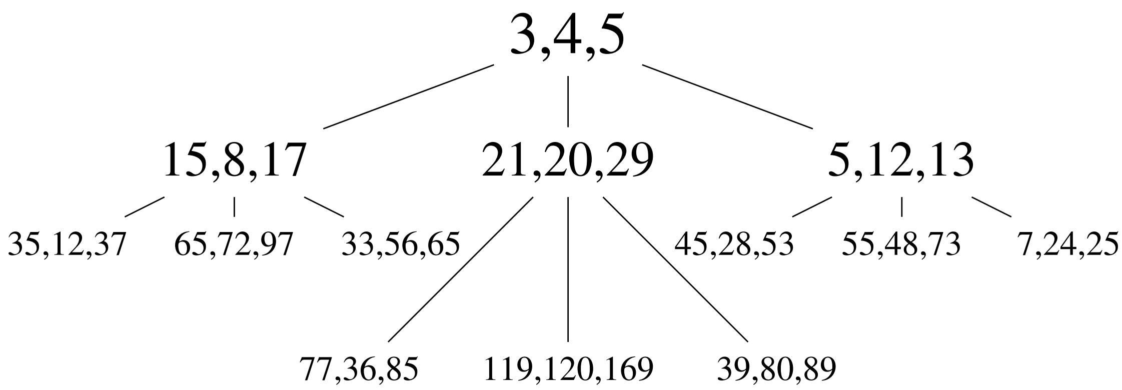

Figure 2.1: First 3-generations with Berggren matrices

Price discovered an entirely different ternary tree seen in Fig 2.2 [24], where the same process can also be applied with P_j defined as

$$P_1 = \begin{bmatrix} 2 & 1 & -1 \\ -2 & 2 & 2 \\ -2 & 1 & 3 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 2 & 1 & 1 \\ 2 & -2 & 2 \\ 2 & -1 & 3 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ 2 & 1 & 3 \end{bmatrix}$$

Thus, $P_1(3,4,5)^T = (5,12,13)^T$, a different triple which could also be generated with the third Berggren matrix $M_3(3,4,5)^T = (5,12,13)^T$.
Another comparison example with both the Berggren matrices and Price matrices would be $M_3(5,12,13)^T = (7,24,25)^T = P_3(3,4,5)^T$. An interesting thing to note would be that the three Berggren matrices are alike except for the signs, same with the Price matrices.

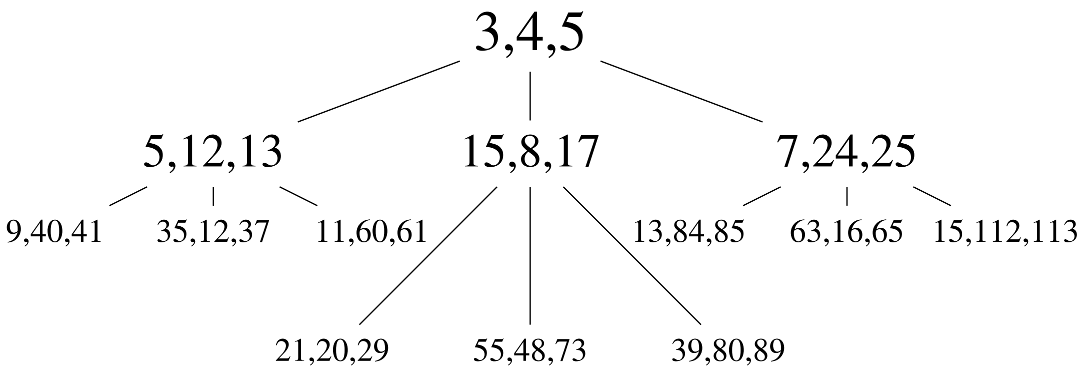

Figure 2.2: First 3-generations with Price matrices

Algorithm 3 Berggren matrices

procedure BERGGREN(x,y,z) ▷ x,y,z being a Pythagorean triple
$a =<< -1,-2,-2 > | < 2,1,2 > | < 2,2,3 >>$
$b =<< 1,2,2 > | < 2,1,2 > | < 2,2,3 >>$
$c =<< 1,2,2 > | < -2,-1,-2 > | < 2,2,3 >>$
$A =< x,y,z >$
$PT1 = a.A$ ▷ Matrix multiplication
$PT2 = b.A$
$PT3 = c.A$
Return $PT1, PT2, PT3$

Algorithm 4 Price matrices

procedure PRICE(x,y,z) ▷ x,y,z being a Pythagorean triple
$a =<< 2,-2,-2 > | < 1,2,1 > | < -1,2,3 >>$
$b =<< 2,2,2 > | < 1,-2,-1 > | < 1,2,3 >>$
$c =<< 2,2,2 > | < -1,2,1 > | < 1,2,3 >>$
$A =< x,y,z >$
$PMT1 = a.A$ ▷ Matrix multiplication
$PMT2 = b.A$
$PMT3 = c.A$
Return $PMT1, PMT2, PMT3$

2.3.4 Area proportional to the sum of squares

This involves generating b and c from a, where a needs to be an odd integer and b and c are consecutive $(b+1=c)$ integers. From the generated formula

$$Area = 6[1^2+2^2+...+((a-1)/2)^2]$$

where $b=(a^2-1)/2$ and $c=(a^2+1)/2$.[2]

Algorithm 5 Sum of squares method

procedure SQUARESMETHOD(a)
    if a **mod**$2=1$ **then**
        $b=(a^2-1)/2$
        $c=(a^2+1)/2$
        $A=add(seq(i^2,i=1..(a-1)/2))$
        if $b+1=c$ **and** $6A=1/(2ba)$ **then**
            Return c,a,b

2.3.5 Polynomial Parametrization

Frisch & Vaserstein [15] posed the following interesting variant on the problem of generating Pythagorean triples. They started by considering $(3,4,5)$ and $(4,3,5)$ as separate triples[2]. In accordance with this convention, they declared the parameterizations $(m^2-n^2,2mn,m^2+n^2)$ and $(2mn,m^2-n^2,m^2+n^2)$ to be two different parametrizations of triples. The question then can be posed whether there is a *single* parametrization that will by itself generate both variants of each triple. They show that no single triple of polynomials with integer coefficients in any number of variables is sufficient for generating all Pythagorean triples. Note the requirement that the polynomials have integer coefficients. A polynomial with integer coefficients must take integer values, given integer arguments. There exist, however, polynomials with rational coefficients which take integer values for integer arguments. If these are allowed, then a parametrization is possible.

The method was derived with the following steps:

- Given a Pythagorean triple (a,b,c) with $\gcd(a,b,c)=1$ and $c>0$. The triple is one of the two forms

$$T_1(m,n)=(m^2-n^2,2mn,m^2+n^2)\ ,$$
$$T_2(m,n)=(2mn,m^2-n^2,m^2+n^2)\ .$$

- Then
$$2T_2(m,n)=(4mn,2(m^2-n^2),2(m^2+n^2))=T_1(m+n,m-n)\ .$$

- Thus every Pythagorean triple is of the form $kT_1(m,n)/2$ with $k\in 1,2$ and $m,n\in\mathbb{Z}$.

[2] In §2.1 above, triples were defined as being in ascending order. Frisch & Vaserstein clearly set this aside.

- Let $T(m,n,k) = \frac{k(m^2-n^2)}{2}$, kmn, $\frac{k(m^2+n^2)}{2}$. Then every Pythagorean triple is of the form $T(m,n,k)$ with $m,n,k \in \mathbb{Z}$.

- Let $m = y+zw$, $n = z-yw$ and $k = 2x-xw$, then $T(m,n,k)$ gives a parametrization of the set of Pythagorean triples by a triple of integer-valued polynomials.

The parametrization of positive Pythagorean triples is given by

$$\left(\frac{(x+(1-w)^2x)((y+(w+1)z)^2-y^2)}{2}, (x+(1-w)^2x)(y+(w+1)z)y, \frac{(x+(1-w)^2x)((y+(w+1)z)^2+y^2)}{2}\right) \quad (2.3)$$

where x,y,z range through the positive integers and w through the non-negative integers. A Maple session illustrates this below.

```
> SV:=proc(w,x,y,z) local a,b,c;
>      if x>0 and y>0 and z>0 and w>=0 then
>        a:= (x + (1 - w)^2 *x)*((y + (w + 1)*z)^2 - y^2);
>        b:= (x + (1 - w)^2 *x)*(y + (w + 1)*z)*y;
>        c:= (x + (1 - w)^2*x)*((y + (w + 1)*z)^2 + y^2);
>        return ([a/2,b,c/2]);
>      else
>        error("Incorrect values");
>      end if;
>    end proc:
> SV(0,1,1,1)
                          [3, 4, 5]
> SV(1,1,1,1)
                          [4, 3, 5]
> SV(1,1,2,1)
                          [6, 8, 10]
> SV(0,1,1,2)
                          [8, 6, 10]
> SV(0,1,2,1)
                         [5, 12, 13]
> SV(1,1,1,2)
                         [12, 5, 13]
```

2.4 Pictures of triples

Since a triple (a,b,c) is also a Pythagorean vector $[a,b]$, it can be treated as the coordinates of a point. Before plotting such points, note that there are several ways to create the vectors. Each method above generates an ordering of the vectors and in addition, the Maple `sort` command can be used to change the order. A list can be ordered based on the size of the largest (last) element, or the first. Also, the list can be restricted to primitive (relatively prime) triples or all triples.

With this in mind, we take a list of triples a,b,c and plot (a,b) as a point. The plots look different, depending upon the ordering. In figure 2.5, the points have been generated using only the parametrization (2.1). Thus the vectors always start with an odd integer followed by an even integer, and in addition, the elements of the vectors are sometimes in ascending order and sometimes in descending order. The plot is displayed in figure 2.3.

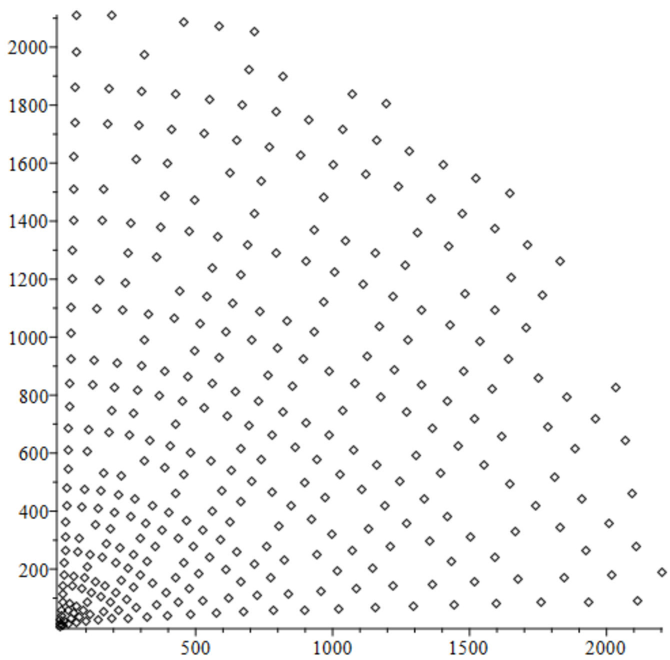

Figure 2.3: Plot of all primitive Pythagorean vectors $\langle a,b\rangle$ calculated using (2.1).

Looking at the plot, our eyes pick out collections of points that appear to form curves. We can see rising curves and falling curves. Is this a case of *apophenia*, which is a perception of a pattern which is not really there? To explore this question, Figure 2.5 shows an enlarged plot with curves fitted through some points, according to the Maple code shown in figure 2.4. In figure 2.4, the expression $p2$ is used to plot the red and the blue line, and $p3$ is the outer green curve. For the red line, a is expressed as $2p+1$ be and b is expressed as $2p(p+1)$ with p being an odd number. Note that the points on the red line have norms c that are consecutive to b. The points on the blue line have the difference of 6 for a, so a is expressed as $6p+9$ and b as $2p(p+3)$ with p an even number. Also, note that the norms of the points on the blue line have a difference of 9 with b. The green line represents the points sloping down where the difference is $2k$ ($k =$ even numbers) e.g the outer curve is $2*40 = 80$, the next is $2*38 = 76$ and so on. Note that 4 is a significant number because the difference between the differences is 4 for both the vertical curves and the sloping curves.

```
> with(plots):
> p1 := pointplot(ForPlot3):
  p2 := plot([[2*p + 1, 2*p*(p + 1), p = 2 .. 20], [6*p + 9, 2*p*(p
  + 3), p = 2 .. 20]]):
  p3 := plot([400 - (2*p + 1)^2, 40*(2*p + 1), p = 2 .. 6]):
  display(p1, p2, p3);
```

Figure 2.4: maple command for the lines in Fig.2.5

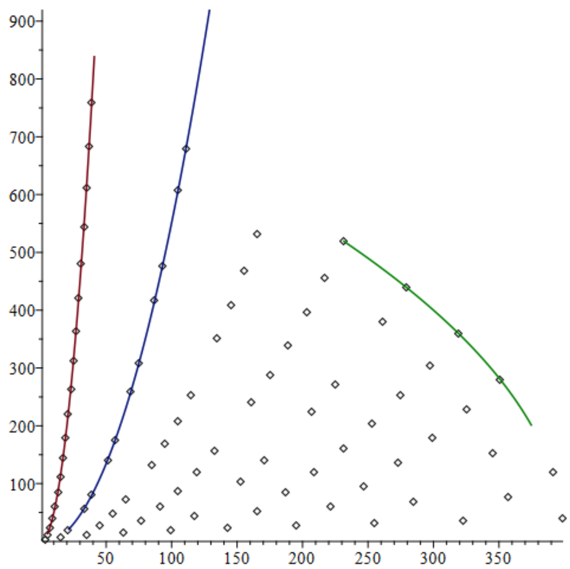

Figure 2.5: Plot of a and b with 2.1

Another way to consider the lines is to refer back to the parametrization used in (2.1), which was also called $T1(m,n)$ in §2.3.5. The blue line is now seen to correspond to the one-parameter family $T1(p,p-1)$. We can note that $T1(p+1,p) = (2p+1,2p(p+1))$ which accords with the plot command in figure 2.4. Since $T(p+2,p)$ is not primitive, there is no line corresponding to it. The blue line is then $T1(p+3,p) = (6p+9,2p(p+3))$. Not that there are gaps in the points defining the blue line; they are non-primitive points. The family of curves illustrated by the green line corresponds to $T1(M,M-(2p+1))$ if M is even, and $T1(M,M-2p)$ if M is odd. It seems then that our eyes are picking out one-parameter families from the plots.

Ordering the list of a,b in ascending order, then plotting with the lesser value as the x-axis gives the visually pleasing figure in Fig.2.6. While it may look like there is a pattern to it, there really isn't one. The lower triangle is empty because the plot has been restricted to ascending order. If the converse is plotted, that is the descending order, the triangle-like pattern of the plot would reflect on the lower triangle and the upper would be left empty. Which can be seen in fig. 2.7.

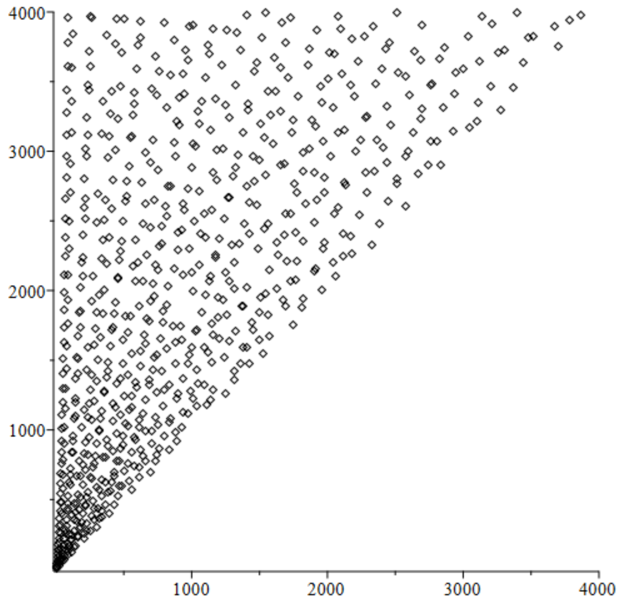

Figure 2.6: Plot of a and b in ascending order

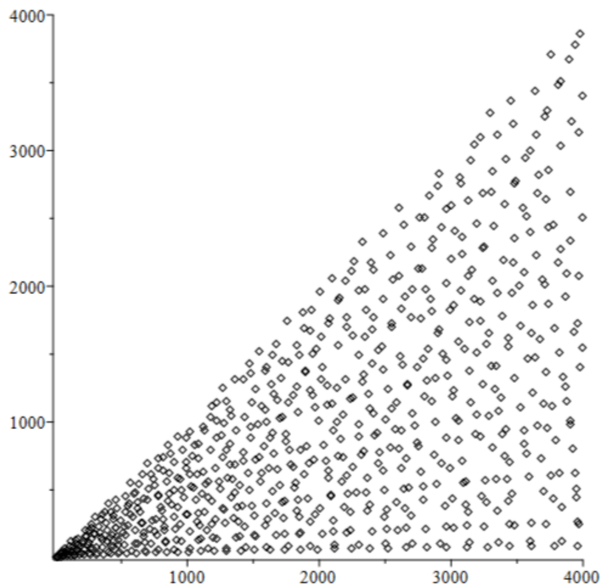

Figure 2.7: Plot of a and b in descending order

2.5 Pythagorean Quadruple

Definition A Pythagorean quadruple is a set of 4 numbers $[a,b,c,d]$ such that

$$a^2+b^2+c^2=d^2$$

A *primitive* Pythagorean Quadruple is one in which its gcd is 1 (i.e co prime).

2.6 Generating Pythagorean Quadruples

2.6.1 Chain method

Pythagorean quadruples can be generated from Pythagorean triples through the chain method. Taking the triple $(5,12,13)$ another triple is generated starting with 13: $(13,84,85)$. This gives $5^2+12^2=13^2$ and $13^2+84^2=85^2$, from these, $5^2+12^2+84^2=85^2$ so, the quadruple $(5,12,84,85)$ is generated. Note that each middle number of the Pythagorean triple is a multiple of the preceding one.[2]

2.6.2 Diophantine method

The parametric generators for primitive Pythagorean quadruples[32][5],

$$\begin{aligned} a &= m^2+n^2-p^2-q^2, \\ b &= 2(mq+np), \\ c &= 2(nq-mp), \\ d &= m^2+n^2+p^2+q^2. \end{aligned} \tag{2.4}$$

gives a odd, where m,n,p,q are non-negative integers and co-prime such that $m+n+p+q$ is odd. All primitive Pythagorean quadruple can be found with the parametrization

$$(m^2+n^2-p^2-q^2)^2+(2(mq+np))^2+(2(nq-mp))^2=(m^2+n^2+p^2+q^2)^2. \tag{2.5}$$

And non-primitive solutions where $(a,b,c)>1$ can be found from primitive ones by multiplication.

Theorem 2.6.1 *If the parameters n,m,q,p of Eq.2.4 are subjected to the conditions*

$$\begin{aligned} &(a) nq > mp, \quad (b) n^2+m^2 > q^2+p^2 \\ &(c_1) n >= 1, m >= 0, \qquad (c_2) q >= 1, p >= 0 \qquad (c_3) p+m >= 1, \\ &(d) n+m+q+p \equiv 1 (mod 2), \\ &(e)(n^2+m^2, q^2+p^2, np+mq) = 1, \\ &(f) p = 0 \to n <= m, \quad (g) m = 0 \to q <= t, \end{aligned}$$

then each primitive solution of $a^2+b^2+c^2=d^2$ is obtained once and only once.

A detailed proof and application of this theorem can be found in [32].

2.6.3 Polynomial Parametrization

Just like in Polynomial parametrization section of generating Pythagorean triples, Frisch and Vaserstein found a single polynomial Pythagorean quadruple which covers all Pythagorean quadruples. Given

$$f(k,n,m,p,q) = k(2np+mq, 2nq-mp, n^2+m^2-p^2-q^2, n^2+m^2+p^2+q^2).$$

The polynomial Pythagorean quadruple

$$g = f(k/2, n, m, p, n+m+p+2z) \in \mathbb{Z}[k, n, m, p, z] \tag{2.6}$$

in 5 parameters covers all Pythagorean quadruples.

2.7 Higher Dimensions

The generation of Pythagorean $n-$tuples, and hence Pythagorean vectors has been explored in a number of papers. Parametric expressions for quintuples and sextuples have been given in [16], but the number of required parameters increases rapidly with size. The quintuple parametrization, for example, requires 14 parameters (presented in all its glory in the appendix). It is unlikely that a septuple parametrization will be published. Other schemes categorize n-tuples as elemental or compound. Thus the well-known triples 3,4,5 and 5,12,13 can be compounded to form the quadruple 3,4,12,13, whereas the quadruple 1,2,2,3 is elemental. A compound $(4+3^n)$-tuple is

$$1, 2, 2, 2\cdot 3, 2\cdot 3, 2\cdot 3^2, 2\cdot 3^2, \ldots, 2\cdot 3^n, 2\cdot 3^n, 3^{n+1}$$

Another method of generating Pythagorean n-tuples stems from a method given by Schwaller[31] for generating Pythagorean triples. Schwaller proposed that x_1 be an odd integer greater than 1 written in the form $2i+1$ where i is a positive integer, then $(2i+1, 2i^2+2i, 2i^2+2i+1)$ is a triple which is a solution for every integer i. With the method described by Schwaller, Landauer[23] generalized the method to generate solutions to the equation

$$x_1^2 + x_2^2 + x_3^2 + \ldots + x_{n-1}^2 = x_n^2$$

for any preselected values of x_1 and n, with x_1 being a positive integer and $n >= 4$.

It is known that $[k, k+1, k(k+1), k(k+1)+1]$ is a solution to the quadruple $x_1^2+x_2^2+x_3^2 = x_4^2$. For quadruple, using this solution with Schwaller's solution, a solution to the Pythagorean quintuple can be found. Let k, $k+1$ and $k(k+1)$ represent x_1, x_2 and x_3 of the equation $x_1^2+x_2^2+x_3^2+x_4^2 = x_5^2$. It can be seen that $x_1^2+x_2^2+x_3^2$ is a perfect square, the problem therefore becomes a Pythagorean triple.
For $x_1^2+x_2^2+x_3^2 = (2i+1)^2$, integer i needs to be found.

$$k^2 + (k+1)^2 + [k(k+1)]^2 = (2i+1)$$

recall the solution to the quadruple

$$\begin{aligned}
&(k(k+1)+1)^2 = (2i+1)^2 \\
&k(k+1)+1 = 2i+1 \\
&k(k+1) = 2i \\
&i = \frac{k(k+1)}{2}
\end{aligned}$$

Let the solution to i be I. Since k is an integer, $k+1$ is also an integer either k or $k+1$ is even so I is therefore an integer. A solution to the Pythagorean quintuple can therefore be obtained $[k, k+1, k(k+1), 2I^2+2I, 2I^2+2I+1]$. Landauer goes on to give a solution to the Pythagorean sextuple and illustrates with some examples[23].

We have written a simple exhaustive search which can list systematically all n-tuples for a specified n. The algorithm below generates Pythagorean n-tuples or vectors of dimension n. This algorithm returns a list of lists. Maple code is included in Appendix B.

Algorithm 6 Pythagorean n-tuples

Input: d, dimension of Pythagorean vector; m, the maximum element size; opt, preferred order.

Output: Pythagorean vector/n-tuple

1. Initialise $outlist = []$, $v = Vector(d)$, and $inc = 1$.
2. Until $inc = 0$;
   (a) Find the initial value, $vinit = v[inc] + 1$ and the remaining values $v[inc..d] = Vector(d - inc + 1, fill = vinit)$, re-initialise $inc = d$.
   (b) Until $m < v[inc]$, find the 2-Norm of v and if it is an integer and its igcd is 1 then set a loop for the optional argument of ordering, opt, convert v and the norm as the case may be into a list and store in outlist. Exit the loop.
   (c) Set $v[inc] = v[inc] + 1$.
   (d) Reduce inc by 1.
   (e) While $0 < inc$ and $m < v[inc]$ reduce the increment value by 1.
3. Return outlist.

2.7.1 How common are Pythagorean vectors?

Primitive Pythagorean triples appear to be rare. After the famous $3,4,5$ the next two are $5,12,13$ and $8,15,17$. This rarety may be one reason for the interest in them, apart from the practical applications. Since we intend to use these vectors to build orthonormal matrices, there is some interest in the statistics of their distribution. Will the repository be able to offer a useful selection of matrices? There are a number of ways one can count the vectors. One way is to count with respect to their length (equivalently the last entry in an n-tuple). In table 2.1, the number of vectors with lengths less than or equal to 20 are given for different dimensions. As dimension increases, the numbers increase to a maximum and then decrease as the minimum length of a vector increases, since the length of a vector of dimension n is bounded below by $\sqrt{n}$. The table could be extended to dimension 400 before reaching zero, but here it stops at 8, after reaching a maximum.

Dimension	2	3	4	5	6	7	8
Number	3	14	53	173	421	1616	1590

Table 2.1: Numbers of primitive Pythagorean vectors with length less than 21 for various dimensions.

Another way to count is to limit the size of the elements. This is more relevant to constructing a repository, since anyone using it will likely want to select matrices with small elements, rather than small length. The number of vectors will increase indefinitely, although viewed as a percentage of the total number of possible vectors at any dimension shows a decrease, so in a sense they become rarer.

Dimension	2	3	4	5	6	7	8
Number	3	12	60	228	851	2444	6608

Table 2.2: Numbers of primitive Pythagorean vectors with elements less than 16 for various dimensions.

One final exploration of the properties is prompted by the exhaustive search. The procedure increments the elements in steps of 1, and yet this is clearly too cautious. Given a vector $\langle 1,2,2,4\rangle$, what is the smallest n for which $\langle 1,2,2,n\rangle$ is also Pythagorean, with the obvious generalization? Some results relevant to this question can be obtained from the parametric expressions.

For dimension 2, we consider $[2mn, m^2-n^2]$. Consider $m=3, n=2 \rightarrow [12,5]$. Now take $m=6, n=1 \rightarrow [12,35]$. This is enough to see a pattern: starting from $n=1$ and $m=p_1p_2p_3\ldots$, where the p_k are prime, the factors can be passed from m to n to generate vectors with the same first (even) element. Therefore, the smallest jump is $35-5=30$. The next smallest is 78, from the triples $20,21,29$ and $20,99,101$. A search shows that a similar possibility exists for vectors starting with an odd number, such as $[15,8],[15,112]$ or $[63,16],[63,216]$. Notice, however, that the last vector is not primitive. We do not have a simple pattern for these examples.

For dimension 3, we can consider when vectors share the first 2 elements, such as $[1,12,12]$, $[1,12,72]$ and $[6,6,7],[6,6,17]$. Since we have the parametrization (2.4), a similar analysis is possible for this case. For vectors of dimension 3, amongst the first 350 vectors, there are 62 pairs of vectors sharing the same first 2 elements. The smallest difference between third elements is 10, for $[6,6,7]$ and $[6,6,17]$. For vectors of dimension 4, among the first 300 vectors, there are 54 pairs with the first 3 elements common. The smallest difference between 4th elements is 8, the smallest example being $[1,4,26,26]$, $[1,4,26,34]$.

Chapter 3

QR Matrices

3.1 Orthogonalization

Definition Two vectors are called orthogonal if they are perpendicular to each other or their dot product is zero.

Definition A real matrix A is orthogonal when its columns and rows are orthonormal vectors

$$A^T A = AA^T = I$$

where A^T is the transpose of A and I the identity matrix.

A few properties of an Orthogonal matrix includes invertible $A^{-1} = A^T$, unitary $A^{-1} = A^*$, with A^* the Hermitian adjoint, normal $A^*A = AA^*$ over the real numbers. Another important property to note is that the determinant of an orthogonal matrix is ± 1, but the converse doesn't always hold, that is if the determinant of a matrix is ± 1 doesn't always mean the matrix is orthogonal.

In the subject of Linear Algebra, an important topic is the *QR* decomposition, calculated using either the Gram–Schmidt process or Householder transformations [1, 7]. It is standard to regard a matrix *A* as a set of vectors, with each column being one vector.

$$A = [\vec{u}_1 \quad \vec{u}_2 \quad \vec{u}_3] \ ,$$

in the usual partitioned notation. Given a set of vectors, the Gram–Schmidt process calculates an orthonormal basis having the same span. For example, Anton [1] gives the following problem and solution. The vector set is

$$\vec{u}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \ , \quad \vec{u}_2 = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} \ , \quad \vec{u}_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \ .$$

The solution is

$$\vec{q}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}, \quad \vec{q}_2 = \begin{bmatrix} \frac{\sqrt{2}}{2\sqrt{19}} \\ -\frac{\sqrt{2}}{2\sqrt{19}} \\ \frac{3\sqrt{2}}{\sqrt{19}} \end{bmatrix}, \quad \vec{q}_3 = \begin{bmatrix} -\frac{3}{\sqrt{19}} \\ \frac{3}{\sqrt{19}} \\ \frac{1}{\sqrt{19}} \end{bmatrix}.$$

The square roots arise because each vector is normalized with respect to the vector 2-norm. If vector $\vec{u} = [u_1, u_2 \ldots u_n]$ then

$$\|\vec{u}\|_2 = \sqrt{|u_1|^2 + |u_2|^2 \ldots |u_n|^2}\,.$$

The normalization is particularly important in *Numerical* Linear Algebra, because it minimizes numerical errors in subsequent calculations [7]. The Gram-Schmidt vectors form a matrix which is orthonormal. Since an orthonormal matrix can be interpreted as a rotation of the basis vectors of a subspace, the QR decomposition

$$A = QR\,,$$

with R upper triangular, can be interpreted as a rotation into a basis which reduces A to a triangular matrix.

For a teacher of Linear Algebra, the Gram–Schmidt process is a frustrating topic to teach and examine. The subject generally contains large amounts of arithmetic, and students are notoriously bad at arithmetic. Students struggle through Reduced Row Echelon Form (RREF) computations using Gaussian Elimination, and then they are confronted with the Gram–Schmidt process. Gram–Schmidt is even worse for them than Gaussian elimination, because of all the square roots that disrupt the students' work. Marking assignments and examination answers becomes a painful chore, because arithmetic mistakes are so common, and students demand part marks for "having the right idea".

Students could be helped by giving them Gram–Schmidt examination questions in which each normalization works out to be square-root free. One could even imagine helping students reach the end of the calculation by advising them that all square-roots have been constructed to simplify. It is the object here to show that this is possible. It should be noted that textbooks on numerical linear algebra are probably not aware of this issue, or at least not concerned about it, because all working is reduced to floating-point data and ideally is performed on a computer. Nonetheless, many courses, even numerical ones, still have traditional exams without computational aids. The problem is very much a problem tied into working by hand, often in an exam room.

Investigations of how to create linear algebra problems avoiding algebraic numbers have been published for the eigenvalue problem [18], but here we discuss the Gram–Schmidt process.

The general format for QR decomposition involves reducing a matrix A into two matrices Q and R, $A = QR$, with Q an orthonormal matrix containing only rational entries and R an upper triangular matrix also containing only rational entries. The significance of Pythagorean vectors can be observed by taking a,b,c as a Pythagorean triple, then

$$Q = \begin{bmatrix} a/c & -b/c \\ b/c & a/c \end{bmatrix}$$

is orthonormal.

The goal becomes finding matrices A that have rational QR factors. To find A, an efficient way would be to start with a rational Q matrix and multiply it by an arbitrary upper triangular matrix R having rational elements. Before finding matrix Q, we consider the question of whether the matrix A constructed from a matrix Q would give rational quantities at every reduction step.

For practicality, A and Q matrices are written in terms of their columns, but the R matrix with its elements.

$$A = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix} = QR = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix} \tag{3.1}$$

Then

$$a_k = \sum_{i=1}^{k} q_i r_{ik} \tag{3.2}$$

where the scalar r_{ik} is placed after the vector q_i to line up the indices.

3.2 Gram-Schmidt

Definition The classical Gram-Schmidt process in matrix notation starts from $k = 1$ to n

$$b_k = a_k - \sum_{j=1}^{k-1} (a_k \cdot q_j) q_j, \tag{3.3}$$

$$q_k = \frac{b_k}{||b_k||}, \tag{3.4}$$

where, throughout this chapter, the norm is the 2-norm.

Definition The modified Gram-Schmidt process follows, if $(b_1, b_2, ..., b_n)$ is a set of vectors forming a basis then an orthonormal basis $(u_1, u_2, ..., u_n)$ can be constructed by [10]

$$\tilde{u_j} = b_j - \sum_{k=1}^{j-1} \frac{(\tilde{u_j}^T b_k)}{\tilde{u_j}^T \tilde{u_j}} \tilde{u_j} \tag{3.5}$$

$$u_j = \frac{\tilde{u_j}}{||\tilde{u_j}||}. \tag{3.6}$$

The following Maple algorithm implements the modified Gram-Schmidt process:

Algorithm 7 Modified Gram-Schmidt

```
Require: A :: Matrix
n = ColumnDimension(A)
r = Matrix(n, n, shape = trianguar[upper])
q = Matrix(n)
a = Copy(A)
for i to n do
    r[i,i] = Norm(a[..,i], 2)
    q[..,i] = a[..,i]/r[i,i]
    for k from i+1 to n do
        r[i,k] = DotProduct(q[..,i], a[..,k])
        a[..,k] = -r[i,k] * q[..,i] + a[..,k]
Return q, r
```

3.3 Householder Transformation

A second method for constructing QR factors uses Householder[1] transformations (also often called reflections) [7]. Given a matrix A of size $n \times n$, the method constructs a sequence of orthonormal matrices H_i such that the product $H = \prod_{i=n-1}^{1} H_i$ reduces A to upper triangular form. That is

$$HA = H_{n-1}H_{n-2}\dots H_1 A = R \ .$$

This corresponds to the common QR factoring because

$$A = H^{-1}R = QR \ .$$

The computation of the H_i is a simple generalization of the computation of H_1. We label each column of A using the usual partition notation.

$$A = [\vec{u}_1 \quad \vec{u}_2 \quad \dots \quad \vec{u}_n] \ .$$

Define

$$v = \|u_1\| e_1 - u_1 \ ,$$

where $e_1^T = [1, 0, \dots, 0]$, and the norm is a 2-norm. The Householder transformation of v is its reflection with respect to a hyperplane v in R^n orthogonal to v, through the origin represented by the outer product of v with itself vv^T, then the $n \times n$ orthogonal matrix

$$H_{v^\perp} = I - \frac{2vv^T}{v^T v} \tag{3.7}$$

is called the Householder matrix. Then H_1 is this matrix using the first column.

$$H_1 = I - \frac{2vv^T}{v^T v} \ .$$

[1] Alston Scott Householder 1904–1993

Then H_1 is orthonormal and H_1A is a matrix in which the first column is $\|u_1\|e_1$. We then continue with the submatrix obtained from H_1A by eliminating the first row and column.

We are interested here in the pedagogical aspects of this procedure. If students try to apply this procedure working without computer assistance, most matrices will quickly lead to a series of awkward square-root contaminated calculations. Even with a system such as Maple, the simplest result is not straightforward. For example, consider the matrix

$$T = \begin{bmatrix} 1 & 4 & 3 \\ 4 & 1 & 2 \\ 3 & 2 & 1 \end{bmatrix}. \tag{3.8}$$

Using the first column as the vector z above: $z = [1, 4, 3]$ and then

$$v = \begin{bmatrix} \sqrt{26} - 1 \\ -4 \\ -3 \end{bmatrix}. \tag{3.9}$$

Before simplification, the Householder matrix H_1 is

$$H_1 = \begin{bmatrix} 1 - \dfrac{2(\sqrt{26}-1)^2}{25+(\sqrt{26}-1)^2} & \dfrac{8(\sqrt{26}-1)}{25+(\sqrt{26}-1)^2} & \dfrac{6(\sqrt{26}-1)}{25+(\sqrt{26}-1)^2} \\ \dfrac{8(\sqrt{26}-1)}{25+(\sqrt{26}-1)^2} & 1 - \dfrac{32}{25+(\sqrt{26}-1)^2} & \dfrac{-24}{25+(\sqrt{26}-1)^2} \\ \dfrac{6(\sqrt{26}-1)}{25+(\sqrt{26}-1)^2} & \dfrac{-24}{25+(\sqrt{26}-1)^2} & 1 - \dfrac{18}{25+(\sqrt{26}-1)^2} \end{bmatrix}$$

Simplifying the $(1, 1)$ element gives

$$1 - \frac{2(\sqrt{26}-1)^2}{25+(\sqrt{26}-1)^2} = \frac{\sqrt{26}-1}{26-\sqrt{26}} = \frac{\sqrt{26}}{26}. \tag{3.10}$$

This task somewhat basic to some can end up being an error prone task to a student who without the computer or calculator aid finds it difficult to simplify. It should be noted that it is not only a student who might find it difficult to simplify: Maple `simplify` command does not succeed in obtaining the best form. For the expression below, the `evala` command in needed. The simplified H_1 matrix is

$$H_1 = \begin{bmatrix} \dfrac{\sqrt{26}}{26} & \dfrac{2\sqrt{26}}{13} & \dfrac{3\sqrt{26}}{26} \\ \dfrac{2\sqrt{26}}{13} & \dfrac{9}{25} - \dfrac{8\sqrt{26}}{325} & -\dfrac{12}{25} - \dfrac{6\sqrt{26}}{325} \\ \dfrac{3\sqrt{26}}{26} & -\dfrac{12}{25} - \dfrac{6\sqrt{26}}{325} & \dfrac{16}{25} - \dfrac{9\sqrt{26}}{650} \end{bmatrix} \tag{3.11}$$

and thus

$$H_1A = \begin{bmatrix} \sqrt{26} & \frac{7\sqrt{26}}{13} & \frac{7\sqrt{26}}{13} \\ 0 & \frac{36\sqrt{26}}{65} - \frac{3}{5} & \frac{128\sqrt{26}}{325} + \frac{6}{25} \\ 0 & \frac{27\sqrt{26}}{65} + \frac{4}{5} & \frac{96\sqrt{26}}{325} - \frac{8}{25} \end{bmatrix}. \tag{3.12}$$

The second Householder reflection uses the submatrix

$$\begin{bmatrix} \frac{36\sqrt{26}}{65} - \frac{3}{5} & \frac{128\sqrt{26}}{325} + \frac{6}{25} \\ \frac{27\sqrt{26}}{65} + \frac{4}{5} & \frac{96\sqrt{26}}{325} - \frac{8}{25} \end{bmatrix}.$$

From this the second Householder matrix is found as

$$H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{36\sqrt{14}}{175} - \frac{3\sqrt{91}}{175} & \frac{27\sqrt{14}}{175} + \frac{4\sqrt{91}}{175} \\ 0 & \frac{27\sqrt{14}}{175} + \frac{4\sqrt{91}}{175} & -\frac{36\sqrt{14}}{175} + \frac{3\sqrt{91}}{175} \end{bmatrix} \tag{3.13}$$

which is the simplified form that still contains large square-roots that would increase with subsequent steps. We now obtain the R and Q matrices as

$$H_2H_1A = R = \begin{bmatrix} \sqrt{26} & \frac{7\sqrt{26}}{13} & \frac{7\sqrt{26}}{13} \\ 0 & \frac{5\sqrt{91}}{13} & \frac{22\sqrt{91}}{91} \\ 0 & 0 & \frac{2\sqrt{14}}{7} \end{bmatrix} \tag{3.14}$$

and

$$(H_2H_1)^T = Q = \begin{bmatrix} \frac{\sqrt{26}}{26} & \frac{9\sqrt{364}}{182} & \frac{\sqrt{14}}{14} \\ \frac{2\sqrt{26}}{13} & -\frac{3\sqrt{364}}{182} & \frac{\sqrt{14}}{7} \\ \frac{3\sqrt{26}}{26} & \frac{\sqrt{364}}{182} & -\frac{3\sqrt{14}}{14} \end{bmatrix}. \tag{3.15}$$

The maple implementation of the householder process is indicated in Algorithm 8

Algorithm 8 Householder

Require: A

```
m,n = Dimensions(A)
R := copy(A)
Q := IdentityMatrix(m, compact = false)
for k to n-1 do
    z := R[k..n,k]
    z[1] := z[1] + sign(z[1]) * Norm(z,2)
    z := z/Norm(z,2)
    v := Transpose(z)
    R[k..n,k] := R[k..n,k] - ((2*z).(v.(R[k..n,k])))
    Q[k..n,..] := Q[k..n,..] - ((2*z).(v.(Q[k..n,..])))
Return simplify(Transpose(Q)), simplify(R)
```

3.4 Gram-Schmidt Rational Orthonormal Bases

It is obvious that if a matrix A is created using (3.1), then applying the Gram-Schmidt process has to lead a student back to the same product. What should be proved, however, is that all of the *intermediate steps* also require only rational arithmetic.

For the Gram-Schmidt method, we have this theorem:

Theorem 3.4.1 *If A obeys* (3.1) *and* (3.2) *and the q_k and r_{ik} are rational then, for all k, all quantities in* (3.3) *and* (3.4) *are rational.*

Proof. By induction, the values obtained by the reduction are denoted as $\hat{q}_i$ and $\hat{r}_{ik}$. For $k = 1$, $b_1 = a_1 = q_1 r_{11}$ and hence $\hat{q_1} = q_1 \mathrm{sgn}(r_{11})$ and $\hat{r_{11}} = |r_{11}|$.
For general k,

$$b_k = a_k - \sum_{j=1}^{k-1} (a_k.\hat{q}_j)\hat{q}_j = a_k - \sum_{j=1}^{k-1} (a_k.q_j)q_j$$

$$= \sum_{i=1}^{k} q_i r_{ik} - \sum_{j=1}^{k-1} \sum_{i=1}^{k} (q_i r_{ik}).q_j q_j = q_k r_{kk}$$

hence $\hat{q_k} = q = k\,\mathrm{sgn}(r_{kk})$ and $\hat{r_{kk}} = |r_{kk}|$ and all inner products are rational.

3.5 Householder Rational Orthonormal Bases

In the case of Householder transformations, the theorems are more intricate, and suggest some interesting side results regarding Pythagorean vectors. As before, in order to prevent square roots from occurring, a matrix is constructed using a known rational Q and rational R. We illustrate the process with an example, with R containing symbolic entries for generality.

$$Q=\begin{bmatrix} \frac{2}{9} & \frac{1}{13} & \frac{46}{117} & \frac{8}{9} \\ \frac{4}{9} & -\frac{10}{13} & \frac{47}{117} & -\frac{2}{9} \\ \frac{5}{9} & -\frac{2}{13} & -\frac{92}{117} & \frac{2}{9} \\ \frac{2}{3} & \frac{8}{13} & \frac{10}{39} & -\frac{1}{3} \end{bmatrix} \qquad R=\begin{bmatrix} 9 & 18 & d & g \\ 0 & 7 & e & h \\ 0 & 0 & 157 & i \\ 0 & 0 & 0 & j \end{bmatrix}.$$

Taking the dot product, we have

$$QR=A=\begin{bmatrix} 2 & \frac{59}{13} & \frac{2d}{9}+\frac{e}{13}+\frac{7222}{117} & \frac{2g}{9}+\frac{h}{13}+\frac{46i}{117}+\frac{8j}{9} \\ 4 & \frac{34}{13} & \frac{4d}{9}-\frac{10e}{13}+\frac{7379}{117} & \frac{4g}{9}-\frac{10h}{13}+\frac{47i}{117}-\frac{2j}{9} \\ 5 & \frac{116}{13} & \frac{5d}{9}-\frac{2e}{13}-\frac{14444}{117} & \frac{5g}{9}-\frac{2h}{13}-\frac{92i}{117}+\frac{2j}{9} \\ 6 & \frac{212}{13} & \frac{2d}{3}+\frac{8e}{13}+\frac{1570}{39} & \frac{2g}{3}+\frac{8h}{13}+\frac{10i}{39}-\frac{j}{3} \end{bmatrix}. \tag{3.16}$$

From the matrix in (3.16), vector v is found as

$$v_1=\begin{bmatrix} 7 \\ -4 \\ -5 \\ -6 \end{bmatrix}. \tag{3.17}$$

Then Householder matrix H_1 is

$$H_1=\begin{bmatrix} \frac{2}{9} & \frac{4}{9} & \frac{5}{9} & \frac{2}{3} \\ \frac{4}{9} & \frac{47}{63} & -\frac{20}{63} & -\frac{8}{21} \\ \frac{5}{9} & -\frac{20}{63} & \frac{38}{63} & -\frac{10}{21} \\ \frac{2}{3} & -\frac{8}{21} & -\frac{10}{21} & \frac{3}{7} \end{bmatrix}. \tag{3.18}$$

And the product

$$H_1A=\begin{bmatrix} 9 & 18 & d & g \\ 0 & -\frac{66}{13} & -\frac{66e}{91}+\frac{8949}{91} & -\frac{66h}{91}+\frac{57i}{91}+\frac{2j}{7} \\ 0 & -\frac{9}{13} & -\frac{9e}{91}-\frac{7222}{91} & -\frac{9h}{91}-\frac{46i}{91}+\frac{6j}{7} \\ 0 & \frac{62}{13} & \frac{62e}{91}+\frac{8478}{91} & \frac{62h}{91}+\frac{54i}{91}+\frac{3j}{7} \end{bmatrix}. \tag{3.19}$$

Then applying the transformation to the $3x3$ submatrix in (3.19)

$$\begin{bmatrix} -\frac{66}{13} & -\frac{66e}{91}+\frac{8949}{91} & -\frac{66h}{91}+\frac{57i}{91}+\frac{2j}{7} \\ -\frac{9}{13} & -\frac{9e}{91}-\frac{7222}{91} & -\frac{9h}{91}-\frac{46i}{91}+\frac{6j}{7} \\ \frac{62}{13} & \frac{62e}{91}+\frac{8478}{91} & \frac{62h}{91}+\frac{54i}{91}+\frac{3j}{7} \end{bmatrix}$$

gives

$$v_2 = \begin{bmatrix} \frac{157}{13} \\ \frac{9}{13} \\ -\frac{62}{13} \end{bmatrix}.$$

The second Householder reflection becomes

$$H_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{66}{91} & -\frac{9}{91} & \frac{62}{91} \\ 0 & -\frac{9}{91} & \frac{14206}{14287} & \frac{558}{14287} \\ 0 & \frac{62}{91} & \frac{558}{14287} & \frac{10443}{14287} \end{bmatrix} \tag{3.20}$$

which gives

$$A_2 = H_2 A_1 = \begin{bmatrix} 9 & 18 & d & g \\ 0 & 7 & e & h \\ 0 & 0 & -85 & -\frac{85i}{157}+\frac{132j}{157} \\ 0 & 0 & 132 & \frac{132i}{157}+\frac{85j}{157} \end{bmatrix}. \tag{3.21}$$

At this stage we can make a number of interesting observations. The Householder matrix in (3.20) is rational, meaning its columns are Pythagorean (up to normalization), and thus new Pythagorean vectors were derived from the (not necessarily Pythagorean) columns of A. This implies a parametric expression for Pythagorean vectors.

Theorem 3.5.1 *Given an integer vector $[p_1, ..., p_n]$ not necessarily Pythagorean, then*

$$V_n = [2p_1^2 - \sum_{i=1}^{n} p_i^2, 2p_1p_2, ..., 2p_1p_n]$$

is a Pythagorean vector.

Proof: It is straightforward to prove that

$$\|\mathbf{v}_n\| = \sum_{k=0}^{n} p_k^2 \ .$$

By induction, for $\mathbf{v}_1$, it is well known that

$$\|\mathbf{v}_1\| = \|\langle p_0^2 - p_1^2, 2p_0p_1\rangle\| = p_0^2 + p_1^2 \ .$$

Assuming the theorem is true for $i = 1..n-1$. That is, assuming

$$\begin{aligned}
\|v_{n-1}\|^2 &= \|\langle p_0^2 - \sum_{i=1}^{n-1} p_i^2, 2p_0pi, ..., 2p_0p_{n-1}\rangle\| \\
&= \left(p_0^2 - \sum_{i=1}^{n-1} p_i^2\right)^2 + \sum_{i=1}^{n-1} 4p_0^2p_i^2, \\
&= p_0^4 + 2p_0^2 \sum_{i=1}^{n-1} p_i^2 + \sum_{i=1}^{n-1} p_i^2, \\
&= \left(\sum_{k=0}^{n-1} p_k^2\right)^2 .
\end{aligned}$$

Then

$$\begin{aligned}
\|v_n\|^2 &= \|p_0^2 - \sum_{i=1}^{n-1} p_i^2 - p_n^2, 2p_0pi, ..., 2p_0p_{n-1}, 2p_0p_n\|, \\
&= \left(p_0^2 - \sum_{i=1}^{n-1} p_i^2 - p_n^2\right)^2 + \sum_{i=1}^{n-1} 4p_0^2p_i^2 + 4p_0^2p_n^2 \ , \\
&= \left(p_0^2 - \sum_{i=1}^{n-1} p_i^2\right)^2 - 2\left(p_0^2 - \sum_{i=1}^{n-1} p_i^2\right) p_n^2 + p_n^4 + \sum_{i=1}^{n-1} 4p_0^2p_i^2 + 4p_0^2p_n^2 \ , \\
&= \left(p_0^2 + \sum_{i=1}^{n-1} p_i^2\right)^2 + 2p_0^2p_n^2 + 2\left(\sum_{i=1}^{n-1} p_i^2\right) p_n^2 + p_n^4 \ , \\
&= \left(p_0^2 + \sum_{i=1}^{n-1} p_i^2 + p_n^2\right)^2 , \\
&= \left(\sum_{k=0}^{n} p_k^2\right)^2 .
\end{aligned}$$

Now, returning to our example, we now want to apply the transformation to the submatrix seen in (3.21).

$$\begin{bmatrix} -85 & -\frac{85i}{157} + \frac{132j}{157} \\ 132 & \frac{132i}{157} + \frac{85j}{157} \end{bmatrix} .$$

In order for Householder H_3 to be rational, the first column must again be Pythagorean. We recall that the original matrix was generated using Pythagorean vectors of dimension 4, but now a Pythagorean vector of dimension 2 is needed, which is true for the example, since $[85, 132, 157]$ is a Pythagorean triple. Lady luck was not involved in the calculation process, as the next theorem shows.

Theorem 3.5.2 *Let $v_1 = [x_1, ..., x_n]$ and $v_2 = [y_1, ..., y_n]$ be Pythagorean. Let v_1 be orthogonal to v_2. Then the $(n-1)$-dimensional vector*

$$w = (X - x_1)[y2, ..., y_n] + y_1[x_2, ..., x_n],$$

where $X = ||v_1||$ and $Y = ||v_2||$ is Pythagorean and $||w|| = (||v_1|| - x_1)||v_2||$.

Proof:

$$\begin{aligned}
||w||^2 &= \sum_{k=2}^{n}[(X - x_1)y_k + y_1x_k]^2 \\
&= (X - x_1)^2\sum_{k=2}^{n} y_k^2 + 2(X - x_1)y_1\sum_{k=2}^{n} x_ky_k + y_1^2\sum_{k=2}^{n} x_k^2 \\
&= (X - x_1)^2(Y^2 - y_1^2) + 2(X - x_1)y_1(-x_1y_1) + y_1^2(X^2 - x_1^2) \\
&= (X - x_1)[XY^2 - Xy_1^2 - x_1Y^2 + x_1y_1^2 - 2x_1y_1^2 + y_1^2X + x_1y_1^2] \\
&= (X - x_1)[XY^2 - Xy_1^2 - x_1Y^2 + y_1^2X] \\
&= (X - x_1)[XY^2 - x_1Y^2].
\end{aligned}$$

We continue with

$$v_3 = \begin{bmatrix} 242 \\ -132 \end{bmatrix}$$

to find the third Householder reflection

$$H_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{85}{157} & \frac{132}{157} \\ 0 & 0 & \frac{132}{157} & \frac{85}{157} \end{bmatrix}.$$

Completing our calculation, we finally return to

$$H_3H_2H_1A = R = \begin{bmatrix} 9 & 18 & d & g \\ 0 & 7 & e & h \\ 0 & 0 & 157 & i \\ 0 & 0 & 0 & j \end{bmatrix}$$

and

$$(H_3H_2H_1)^T = Q = \begin{bmatrix} \frac{2}{9} & \frac{1}{13} & \frac{46}{117} & \frac{8}{9} \\ \frac{4}{9} & -\frac{10}{13} & \frac{47}{117} & -\frac{2}{9} \\ \frac{5}{9} & -\frac{2}{13} & -\frac{92}{117} & \frac{2}{9} \\ \frac{2}{3} & \frac{8}{13} & \frac{10}{39} & -\frac{1}{3} \end{bmatrix}$$

Thus we see that at each reduction step the entries are rational.

Chapter 4

Generating Rational Q Matrices

It has been shown above that if a linear algebra problem is created by choosing a rational orthonormal Q matrix and a rational R matrix, then the matrix A can be given to a student as a basis for an exercise in calculating the QR factors. The student who performs the calculation correctly is guaranteed to need only rational arithmetic. Indeed, an instructor could choose to warn students that if they encounter square roots which do not simplify exactly, then they can assume they have made a mistake.

For an instructor to use this as a strategy while setting exercises or exams, there must be available a source of rational orthonormal Q matrices from which the instructor can make a selection. It is therefore desirable to create a data-base resource containing a collection of matrices from which users can make a selection. This chapter discusses the creation of such a database.

4.1 Cayley's formula

Cayley [17] gave a formula for obtaining an orthonormal matrix. Liebeck & Osborne [25] showed that every orthonormal matrix up to reflections in the basis can be obtained from Cayley's formula, given a suitable choice of an input matrix S. We give here a simple proof of the formula and its inverse.

Theorem 4.1.1 *Let S be a skew-symmetric matrix over $\mathbb{R}^{n\times n}$ with no eigenvalue equal to* 1. *The matrix*

$$Q=(S-I)^{-1}(S+I) \tag{4.1}$$

is orthonormal.

Proof: We note that if λ is an eigenvalue of A, then $A-\lambda I$ is singular. Therefore, if S has eigenvalue 1, then the term $(S-I)^{-1}$ in Cayley's formula does not exist.

We show that the formula implies $Q^TQ=I$.

$$\begin{aligned}
Q^TQ &= [(S-I)^{-1}(S+I)]^T[(S-I)^{-}1(S+I)] = (S+I)^T[(S-I)^T]^{-1}(S-I)^{-1}(S+1)\\
&= (S^T+I^T)[(S-I)^T]^{-1}(S-I)^{-1}(S+I) = (-S+I)(-S-I)^{-1}(S-I)^{-1}(S+I)\\
&= (S-I)[S^2-I^2]^{-1}(S+I) = (S-I)[(S+I)(S-I)^{-1}(S+I)\\
&= (S-I)(S-I)^{-1}(S+I)^{-1}(S+I) = I
\end{aligned}$$

QED.

The inverse transform can also be checked.

Theorem 4.1.2 *Let $Q \in \mathbb{R}^{n\times n}$ be an orthonormal matrix. The matrix S given by*

$$S = (Q-I)^{-1}(Q+I) \tag{4.2}$$

is skew symmetric.

Proof:

$$\begin{aligned}
S^T &= \Big[(Q-I)^{-1}(Q+I)\Big]^T = (Q^T+I)(Q^T-I)^{-1} = (Q^{-1}+I)(Q^{-1}-I)^{-1} \\
&= Q^{-1}(Q^{-1}-I)^{-1} + I(Q^{-1}-I)^{-1} = \Big[(Q^{-1}-I)Q\Big]^{-1} + \Big[Q^{-1}(I-Q)\Big]^{-1} \\
&= (I-Q)^{-1} + (I-Q)^{-1}Q = (I-Q)^{-1}(I+Q) \\
&= -(Q-I)^{-1}(Q+I) = -S
\end{aligned}$$

QED

The fact that the transforms are inverses can be proved as follows.

Theorem 4.1.3 *The pair of equations*

$$\begin{aligned}
Q &= (S-I)^{-1}(S+I)\ . \\
S_1 &= (Q-I)^{-1}(Q+I)
\end{aligned}$$

imply that $S_1 = S$.

Proof

$$\begin{aligned}
S_1 = (Q-I)^{-1}(Q+I) &= \Big[(S-I)^{-1}(S+I) - I\Big]^{-1}\Big[(S-I)^{-1}(S+I)+I\Big]\ , \\
&= \Big[(S-I)^{-1}\Big(S+I-(S-I)\Big)\Big]^{-1}\Big[(S-I)^{-1}\Big(S+I+(S-I)\Big)\Big]\ , \\
&= \Big[(S-I)^{-1}2I\Big]^{-1}\Big[(S-I)^{-1}\Big(2S\Big)\Big]\ , \\
&= \tfrac{1}{2}(S-I)(S-I)^{-1}2S = S\ .
\end{aligned}$$

QED

4.1.1 Disadvantages of Cayley

Our aim is to compile, in some systematic order, a gallery of rational orthonormal matrices, for on-line reference purposes. One way to generate the matrices is to use (4.1) repeatedly, with the entries of the skew-symmetric S matrix serving as parameters which can be varied either randomly or systematically. There are inconveniences, however. First, the matrices need to be non-singular, and secondly if one wants to generate themed families, for example, see below for some themed families, or families with some sort of order, there is not as yet a way to direct the selection of the S matrix to yield matrices with a given characteristic. Thirdly, different S matrices can give essentially the same matrix. For example, the two matrices

$$S_1 = \begin{bmatrix} 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \qquad S_2 = \begin{bmatrix} 0 & -1 & -1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \tag{4.3}$$

give the matrices $Q_k = (S_k - I)^{-1}(S_k + I)$

$$Q_1 = \begin{bmatrix} 1/2 & 1/2 & 1/2 & 1/2 \\ -1/2 & -1/2 & 1/2 & 1/2 \\ -1/2 & 1/2 & -1/2 & 1/2 \\ -1/2 & 1/2 & 1/2 & -1/2 \end{bmatrix}, \qquad Q_2 = \begin{bmatrix} 1/2 & 1/2 & 1/2 & -1/2 \\ -1/2 & -1/2 & 1/2 & -1/2 \\ -1/2 & 1/2 & -1/2 & -1/2 \\ 1/2 & -1/2 & -1/2 & -1/2 \end{bmatrix}.$$

We can partially address these difficulties by analyzing simple S matrices, and the matrices they generate. As an example, we considered the following case, generalized from (4.3) above.

Theorem 4.1.4 *Let e be an $n-1$ column vector with entries 1, and let S be a skew-symmetric matrix defined by*

$$S = \begin{bmatrix} 0 & -e^T \\ e & 0 \end{bmatrix}. \tag{4.4}$$

Let I_k be the k-dimensional identity matrix, then $Q = (S - I_n)^{-1}(S + I_n)$ is given by

$$Q = \begin{bmatrix} 1 - \dfrac{2}{n} & \dfrac{2e^T}{n} \\ \dfrac{-2e}{n} & \dfrac{2ee^T}{n} - I_{n-1} \end{bmatrix} \tag{4.5}$$

Proof:

$$(S - I_n)^{-1} = \begin{bmatrix} -1 & -e^T \\ e & -I_{n-1} \end{bmatrix}^{-1} = \begin{bmatrix} \dfrac{-1}{n} & \dfrac{e^T}{n} \\ \dfrac{-e}{n} & \dfrac{ee^T}{n} - I_{n-1} \end{bmatrix}.$$

Then we have

$$(S - I_n)^{-1}(S + I_n) = \begin{bmatrix} \dfrac{-1}{n} & \dfrac{e^T}{n} \\ \dfrac{-e}{n} & \dfrac{ee^T}{n} - I_{n-1} \end{bmatrix} \begin{bmatrix} 1 & -e^T \\ e & I_{n-1} \end{bmatrix} = Q\,.$$

Coding this outcome in maple

```
proc(n) ;
e = Vector(n-1,fill=1);
<<1-2/n,-2*e/n>|<2*Transpose(e)/n,
2*e.Transpose(e)/n-IdentityMatrix(n-1)>>;
```

And the Cayley formula

```
proc(S);
eye = IdentityMatrix(RowDimension(S));
simplify(MatrixInverse(S-eye).(S+eye));
end proc;

proc(n);
v = Vector(n-1,fill=1);
<<0,v>|<-Transpose(v),ZeroMatrix(n-1,n-1)>>;
end proc;
```

4.1.2 Other previous methods

The papers [22, 9] worked by searching for entries in a matrix A that after reduction had rational QR factors. This approach is not appropriate for tabulating, because the present method allows a user to generate many more matrices A than can be tabulated. Also, the present investigations have the advantage of a guarantee that the intermediate steps in QR factoring are rational.

4.2 A simple search approach

In addition to the above approaches, we have programmed a simple exhaustive search. The aim is to compile lists of rational orthonormal matrices, one list for each matrix dimension $n \times n$, with $n = 3, 4$ assumed to be the dimensions of greatest interest. Since the columns of the matrices will be Pythagorean vectors, we limit our search to primitive vectors. Other vectors can be obtained using suitable factors in the R matrix, so only primitive vectors need to be tabulated.

After selecting the desired size, we generate a list of primitive vectors with the given dimension. The search then starts by choosing a Pythagorean vector from the list and assigning it to the first column. We then work through all other vectors in the generated list and test each one to see whether it can be made orthogonal to the first column. Since the vectors all contain only positive elements, there is a search over possible negations of some elements in each column. The search is repeated making selections from the list for a third column. An interesting fact is that the last column does not need to be searched for, because it is uniquely determined. It is also always rational, given that the other columns are rational [22]. It can be noted that the calculations are all done with integer vectors. Since the primary test is orthogonality, the normalization can be left until after the search is successful. The combinatorial demands of the search mean that it is impractical to extend it to larger than dimension 4. Further, the matrices found are dense, because all columns are selected from lists of primitive vectors of a given dimension. There are many possibilities of non-dense, or sparse, matrices, which would require mixing lists of vectors of different dimensions.

4.2.1 Interesting Patterns

By taking out the Least Common Multiple of the elements in a rational orthonormal matrix, we obtain an integer matrix, that is $Q = \frac{1}{\alpha}\Theta$, with $\Theta \in \mathbb{Z}^{n\times n}$. Examining these integer matrices, we can see interesting patterns emerge, which are not so obvious when the matrix elements are rational. We now present several of these patterns.

Circulant-like matrices

A circulant matrix contains rows which are cyclic permutations of the first row. We see a similar pattern in many of the matrices found here. The matrices are called circulant-*like* because the rows vary in their signs. For example

$$\frac{1}{3}\begin{bmatrix}1 & 2 & 2\\ 2 & 1 & -2\\ 2 & -2 & 1\end{bmatrix}, \quad \frac{1}{7}\begin{bmatrix}2 & 6 & 3\\ 3 & 2 & -6\\ 6 & -3 & 2\end{bmatrix}, \quad \frac{1}{19}\begin{bmatrix}6 & 15 & 10\\ 10 & 6 & -15\\ 15 & -10 & 6\end{bmatrix} \tag{4.6}$$

In fact, we can do better than this. We notice that the three examples all correspond to the pattern

$$\begin{bmatrix}ab & bc & ca\\ ca & ab & -bc\\ bc & -ca & ab\end{bmatrix}. \tag{4.7}$$

If we take the inner product of the first and second columns, we get $abca + bcab - caab = abc(a+b-c)$. Thus for orthogonality, we require $a+b=c$. This is reflected in the examples. We also require that $[ab, bc, ca]$ is Pythagorean. All examples we have seen fit the pattern, but we have not proved that this always applies.

There are also 4×4 examples.

$$\frac{1}{7}\begin{bmatrix}1 & 4 & 4 & 4\\ 4 & -1 & -4 & 4\\ 4 & 4 & -1 & -4\\ 4 & -4 & 4 & -1\end{bmatrix}, \quad \frac{1}{13}\begin{bmatrix}11 & 4 & 4 & 4\\ 4 & -11 & -4 & 4\\ 4 & 4 & -11 & -4\\ 4 & -4 & 4 & -11\end{bmatrix}, \quad \frac{1}{14}\begin{bmatrix}11 & 5 & 5 & 5\\ 5 & -11 & -5 & 5\\ 5 & 5 & -11 & -5\\ 5 & -5 & 5 & -11\end{bmatrix}. \tag{4.8}$$

The pattern here is

$$\begin{bmatrix}a & b & b & b\\ b & -a & -b & b\\ b & b & -a & -b\\ b & -b & b & -a\end{bmatrix} \tag{4.9}$$

which is a special case of permutations below.

Bordered circulant-like

The interior block is circulant-like, but not itself rational orthonormal.

$$\frac{1}{4}\begin{bmatrix} 0 & 2 & 2 & 2 & 2 \\ 2 & -3 & 1 & 1 & 1 \\ 2 & 1 & -3 & 1 & 1 \\ 2 & 1 & 1 & -3 & 1 \\ 2 & 1 & 1 & 1 & -3 \end{bmatrix}$$

Permutations

Examples can be found in which the columns are permuted, but not cyclically.

$$\frac{1}{9}\begin{bmatrix} 2 & 4 & 5 & 6 \\ 4 & -2 & -6 & 5 \\ 5 & 6 & -2 & -4 \\ 6 & -5 & 4 & -2 \end{bmatrix}, \quad \frac{1}{9}\begin{bmatrix} 2 & 2 & 3 & 8 \\ 2 & -2 & -8 & 3 \\ 3 & 8 & -2 & -2 \\ 8 & -3 & 2 & -2 \end{bmatrix}, \quad \frac{1}{7}\begin{bmatrix} 2 & 2 & 4 & 5 \\ 2 & -2 & -5 & 4 \\ 4 & 5 & -2 & -2 \\ 5 & -4 & 2 & -2 \end{bmatrix}$$

The pattern here is

$$\begin{bmatrix} a & b & c & d \\ b & -a & -d & c \\ c & d & -a & -b \\ d & -c & b & -a \end{bmatrix}$$

We also have special cases when $b = c$

$$\frac{1}{10}\begin{bmatrix} 1 & 5 & 5 & 7 \\ 5 & -1 & -7 & 5 \\ 5 & 7 & -1 & -5 \\ 7 & -5 & 5 & -1 \end{bmatrix}$$

and when $b = c = d$, which as we can see above is the form of the $4x4$ circulant-like matrix.

Hadamard or Bohemian matrices

There are also matrices that are Hadamard or Bohemian [8].

$$\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{4.10}$$

4.2.2 Fraction-free QR

In [12] and [35] a variation on standard QR factoring was defined. This is called exact-division or fraction-free QR (FFQR), and is analogous to the better-known fraction-free LU popularized by Bareiss. For a matrix $A \in \mathbb{Z}^{n\times n}$, for some $n \in \mathbb{Z}$, it takes the form

$$A = \Theta D^{-1} R\,, \tag{4.11}$$

where $\Theta, D, R \in \mathbb{Z}^{n\times n}$ and D is diagonal and R is upper triangular. Clearly, the Θ in (4.11) will be related to the integer matrices discussed above.

Using the algorithm presented in [12], we obtain the following decomposition.

$$A = \begin{bmatrix} 3 & 27 & 43 & 35 \\ 3 & -3 & -5 & 29 \\ 9 & 41 & 55 & 27 \\ 15 & -15 & -19 & 7 \end{bmatrix} = \Theta D^{-1} R\,, \tag{4.12}$$

where

$$\Theta = \begin{bmatrix} 3 & 8100 & 4050000 & 145800000 \\ 3 & -1620 & -810000 & 845640000 \\ 9 & 11340 & -2430000 & -87480000 \\ 15 & -8100 & 810000 & -145800000 \end{bmatrix},$$
$$D = \begin{bmatrix} 108 & 0 & 0 & 0 \\ 0 & 162000 & 0 & 0 \\ 0 & 0 & 29160000 & 0 \\ 0 & 0 & 0 & 26244000000 \end{bmatrix},$$
$$R = \begin{bmatrix} 108 & 72 & 108 & 180 \\ 0 & 500 & 700 & 300 \\ 0 & 0 & 36 & 72 \\ 0 & 0 & 0 & 900 \end{bmatrix}.$$

The tendency of fraction-free methods to introduce spurious factors is described in [28], where it is shown that the last column of the Θ matrix is always divisible by $\det A = -162000$. Using this simplification, we obtain the decomposition

$$\Theta = \begin{bmatrix} 3 & 8100 & 4050000 & -900 \\ 3 & -1620 & -810000 & -5220 \\ 9 & 11340 & -2430000 & 540 \\ 15 & -8100 & 810000 & 900 \end{bmatrix},$$
$$D = \begin{bmatrix} 108 & 0 & 0 & 0 \\ 0 & 162000 & 0 & 0 \\ 0 & 0 & 29160000 & 0 \\ 0 & 0 & 0 & -162000 \end{bmatrix},$$
$$R = \begin{bmatrix} 108 & 72 & 108 & 180 \\ 0 & 500 & 700 & 300 \\ 0 & 0 & 36 & 72 \\ 0 & 0 & 0 & 900 \end{bmatrix}.$$

This goes a small way in the direction of reducing the spurious factors, but further GCD calculations lead to

$$\Theta = \begin{bmatrix} 1 & 5 & 5 & -5 \\ 1 & -1 & -1 & -29 \\ 3 & 7 & -3 & 3 \\ 5 & -5 & 1 & 5 \end{bmatrix},$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix},$$

$$R = \begin{bmatrix} 3 & 2 & 3 & 5 \\ 0 & 5 & 7 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

It may be that this example unfairly makes the FFQR look worse than it deserves, but it seems that unless GCD reductions are added to the algorithm, it is likely that it will continue to deliver unsatisfactory results.

If we make a distinction between orthonormal matrices, being defined by $Q^T Q = QQ^T = I$, and orthogonal matrices, being defined by $Q^T Q = QQ^T = D$, where $D = dI$ is diagonal, then Θ matrices are in general only left orthogonal, meaning $\Theta^T \Theta = D$, where D is not in general dI, and $\Theta\Theta^T$ is not diagonal.

4.2.3 Pythagorean magic squares

Related to searches for orthogonality is a paper by Euler [13]. The paper is in Latin, and the description here is based on an understanding of the equations. Euler in 1771 embarked on a mission to construct a 4-by-4 square (not magic), but which becomes a magic square of squares. He started off with the square in table 4.1 which uses 8 parameters a, b, c, d, p, q, r, s,

ap+bq+cr+ds	aq-bp+cs-dr	ar-bs-cp+dq	as+br-cq-dp
aq-bp-cs+dr	-ap-bq+cr+ds	-as-br-cq-dp	ar-bs+cp-dq
ar+bs-cp-dq	as-br-cq+dp	-ap+bq-cr+ds	-aq-bp-cs-dr
as-br+cq-dp	-ar-bs-cp-dq	aq+bp-cs-dr	-ap+bq+cr-ds

Table 4.1: Euler's first square

where the sum of the squares in each column, either horizontal or vertical is

$$= (aa + bb + cc + dd)(pp + qq + rr + ss). \quad (4.13)$$

Then for the sums to be equal, he squared the expression and divided each of the number by its root. Euler had an example table 4.2 on the left and its square on the right table 4.3, whose column and row sums was 1530 but the diagonal sum was 2516 and 541.

37	4	1	12
-6	33	-18	9
11	8	-7	-36
-2	19	34	-3

Table 4.2: Example based on table 4.1.

1369	16	1	144	1530
36	1089	324	81	1530
121	69	49	1296	1530
4	361	1156	9	1530
1530	1530	1530	1530	sum

Table 4.3: Example obtained by squaring each element in table 4.2

So, Euler posed another square table 4.4 whose vertical and horizontal sums is eq.4.13.

ap+bq+cr+ds	ar-bs-cp+dq	-as-br+cq+dp	aq-bp+cs-dr
-aq+bp+cs-dr	as+br+cq+dp	ar-bs+cp-dq	ap+bq-cr-ds
ar+bs-cp-dq	-ap+bq-cr+ds	aq+bp+cs+dr	as-br-cq+dp
-as+br-cq+dp	-aq-bp+cs+dr	-ap+bq+cr-ds	ar+bs+cp+dq

Table 4.4: Euler's magic square of squares

For the diagonal sums to be equal, eq.4.14 and eq.4.15 was made

$$abpq+abrs+acpr+dcqs+adps+adqr+bcqr+bcps+bdqs+bdqr+cdrs+cdpq=0 \tag{4.14}$$

$$-abpq-abrs+acpr+acqs-adps-adqr-bcqr-bcps+bdqs+bdpr-cdrs-cdpq=0 \tag{4.15}$$

from which these two are derived

$$(ac+bd)(pr+qs)=0 \tag{4.16}$$

$$(ab+cd)(pq+rs)+(ad+bc)(ps+qr)=0. \tag{4.17}$$

For table 4.4 to be valid the following conditions must hold

$$pr+qs=0 \tag{4.18}$$

$$a/c=(-d(pq+rs)-b(ps+qr))/(b(pq+rs)+d(ps+qr)). \tag{4.19}$$

Using Euler's values, $p=6$ $q=3$ $r=1$ $s=-2$ $a=9$ $b=1$ $c=16$ $d=0$, which satisfies the conditions eq.(4.18) and eq.(4.19) to produce the square, table 4.5 whose squares of square has horizontal, vertical and diagonal sum of 16900.

73	-85	65	11
-53	31	101	41
-89	-67	1	-67
-29	-65	-35	103

Table 4.5: Example 2

4.2.4 A Depository of Rational Orthonormal Q matrices

A repository containing $3X3$ rational orthonormal matrices and $4X4$ rational orthonormal matrices which are either dense or sparse can be found on the ORCCA projects website page. And can also be accessed with the link http://orcca.on.ca/projects.html.

Chapter 5

Conclusion and Further works

5.1 Conclusion

"Do not worry about your difficulties in mathematics, I can assure you mine are still greater."
Albert Einstein

While the struggles of Einstein cannot be compared to the struggles of mathematics students, especially linear algebra students for the purpose of this research, it is quite real and telling. As mentioned in the research purpose, solving problems in mathematics requires patience and the ability to make mistakes and learn from them. While this ability is encouraged, there is also an advantage in making some aspects a little easier for both the students and the instructors, in this case square root free.

In the course of this research, it is evident that there are various methods of generating Pythagorean n-tuples, although only a few are mentioned and described in the paper. A lot more research work are being carried out on Pythagorean n-tuples and its various application to various fields such as in the case of Pythagorean theorem in surveying, navigation and facial recognition to mention a few.

The application of Pythagorean n-tuples to generate rational orthonormal matrices[22] shows an interesting observation such as in the 3×3 case where the last column doesn't need to be found because it is unique and rational. As shown with the Gram-Schmidt method all the quantities and entries at each reduction step are rational which makes a less complicated teaching and learning encounter.

This thesis has some conjectures which were not resolved. They are listed here.

5.1.1 All 3×3 circulant-like matrices follow (4.7)

5.1.2 All 4×4 circulant-like matrices follow (4.9)

5.1.3 Repeated elements lead to degeneracy

One aim of this thesis is the generation of linear algebra exercises. Mostly one prefers examples that do not have lucky simplifications that cause steps to be skipped. If one constructs QR problems from matrices in which the columns have repeated elements, then Householder transformations will lead to trivial submatrices. For example, using a matrix from §4.2.1, we have

$$A = \frac{1}{9}\begin{bmatrix}2 & 4 & 5 & 6\\ 4 & -2 & -6 & 5\\ 5 & 6 & -2 & -4\\ 6 & -5 & 4 & -2\end{bmatrix}\begin{bmatrix}9 & 9 & -9 & 3\\ 0 & 9 & 9 & 3\\ 0 & 0 & 9 & -3\\ 0 & 0 & 0 & -3\end{bmatrix} = \begin{bmatrix}2 & 6 & 7 & -\frac{5}{3}\\ 4 & 2 & -12 & 1\\ 5 & 11 & -1 & \frac{17}{3}\\ 6 & 1 & -7 & -\frac{1}{3}\end{bmatrix}$$

The intermediate matrices in a Householder decomposition are

$$\begin{bmatrix}9 & 9 & -9 & 3\\ 0 & \frac{2}{7} & -\frac{20}{7} & -\frac{5}{3}\\ 0 & \frac{62}{7} & \frac{73}{7} & \frac{7}{3}\\ 0 & -\frac{11}{7} & \frac{47}{7} & -\frac{13}{3}\end{bmatrix}, \quad \begin{bmatrix}9 & 9 & -9 & 3\\ 0 & 9 & 9 & 3\\ 0 & 0 & -\frac{99}{61} & -\frac{147}{61}\\ 0 & 0 & \frac{546}{61} & -\frac{213}{61}\end{bmatrix}, \quad \begin{bmatrix}9 & 9 & -9 & 3\\ 0 & 9 & 9 & -3\\ 0 & 0 & 9 & -3\\ 0 & 0 & 0 & -3\end{bmatrix}$$

At each step a new Householder matrix must be computed. If now a matrix is used with columns containing repeated elements

$$A = \frac{1}{5}\begin{bmatrix}1 & 2 & 2 & 4\\ 2 & -1 & 4 & -2\\ 2 & 4 & -1 & -2\\ 4 & -2 & -2 & 1\end{bmatrix}\begin{bmatrix}5 & 5 & -5 & 1\\ 0 & 5 & 5 & 1\\ 0 & 0 & 1 & -1\\ 0 & 0 & 0 & -1\end{bmatrix} = \begin{bmatrix}1 & 3 & \frac{7}{5} & -\frac{3}{5}\\ 2 & 1 & -\frac{11}{5} & -\frac{1}{5}\\ 2 & 6 & \frac{9}{5} & \frac{9}{5}\\ 4 & 2 & -\frac{32}{5} & \frac{3}{5}\end{bmatrix}$$

The intermediate matrices in a Householder decomposition are

$$\begin{bmatrix}5 & 5 & -5 & 1\\ 0 & 0 & 1 & -1\\ 0 & 5 & 5 & 1\\ 0 & 0 & 0 & -1\end{bmatrix}, \quad \begin{bmatrix}5 & 5 & -5 & 1\\ 0 & 5 & 5 & 1\\ 0 & 0 & 1 & -1\\ 0 & 0 & 0 & -1\end{bmatrix}.$$

Thus the reduction requires a second step that is essentially just rearranging, and then the third step is not needed. Its value as a didactic exercise is reduced. Pity the instructor who builds an exercise using (4.10): the Householder reduction terminates after one step! The conjecture is that the degeneracy is the result of the repeated elements in the columns of the matrix. Notice, it is not a result of the circulant-like or permutation structures, as the first example shows.

The implied snag is that many of the 4×4 matrices that have been found in our systematic search contain columns with repeated elements. This is already apparent when looking at lists of Pythagorean quintuples (dimension 4 vectors). For vectors of higher dimensions, the possibility of repeated elements 1 and 2 means that vectors containing strings of small integers can be adjusted to become Pythagorean. This effect is restricted to matrices in which Pythagorean vectors are required. The matrices used in numerical linear algebra are subject to less temptations to emphasize these structures. More work can be done to find more quintuples that can be used to build $4x4$ rational matrices without repeated elements.

Further research can also be made on the computation of Pythagorean n-tuples of higher dimensions and their application to the method described above. Another research idea could be in the realm of LU decomposition such as the application of the Bareiss method to rational matrices generated with Pythagorean n-tuples.

Bibliography

[1] H. Anton. *Elementary Linear Algebra*. Wiley, 11 edition, 2013.

[2] Edward J. Barbeau. *Power Play*. Mathematical Association of America, 1997.

[3] Raymond A. Beauregard and E. R. Suryanarayan. Proof without words: Parametric representation of primitive Pythagorean triples. *Mathematics Magazine*, 69(3):189, Jun., 1996.

[4] Carl B. Boyer. *A History of Mathematics*. John Wiley and Sons Inc., Hoboken, New Jersey, 3rd edition, 2011.

[5] R. D. Carmichael. *Diophantine Analysis*. John Wiley and Sons, New York, 1915.

[6] Byungchul Cha, Emily Nguyen, and Brandon Tauber. Quadratic forms and their Berggren trees. *Journal of Number Theory*, 185:218—-256, 2018.

[7] R.M. Corless and N. Fillion. *A graduate introduction to numerical methods*. Springer, 2013.

[8] Robert M. Corless. What can we learn from bohemian matrices. *Maple Transactions*, 1(1), 2021.

[9] A. C. Camargos Couto and D. J. Jeffrey. Rational Householder transformations. In *SYNASC 2018*, pages 61–64. IEEE, 2018.

[10] Richard Earl, James R. Nicholson, and Christopher Clapham. *The concise Oxford dictionary of mathematics*. Oxford: Oxford University Press, sixth edition, 2021.

[11] Robson Eleanor. Words and Pictures: New Light on Plimpton 322. *Mathematical Association of America Monthly*, 109(2):105–120, 2002.

[12] Úlfar Erlingsson, Erich Kaltofen, and David Musser. Generic Gram–Schmidt orthogonalization by exact division. In *ISSAC 1996*, pages 275–282. ACM Press, 1996.

[13] Leonhard Euler. Problema algebraicum ob affectiones prorsus singulares memorabile. *Novi Commentarii academiae scientiarum Petropolitanae*, 15:75–106, 1771.

[14] L. Fibonacci and L. E. Sigler. *The book of squares*. Academic Press, Boston, 1987.

[15] Sophie Frisch and L. N. Vaserstein. Parametrization of pythagorean triples by a single triple of polynomials. *J. Pure Appl. Algebra*, 212(1):271–274, 2008.

[16] Sophie Frisch and Leonid N. Vaserstein. Polynomial parametrization of pythagorean quadruples, quintuples and sextuples. *Journal of Pure and Applied Algebra*, 216(1):184–191, 2012.

[17] F. R. Gantmacher. *The Theory of Matrices*, volume I. Chelsea, 1960.

[18] R.C. Gilbert. Companion matrices with integer entries and integer eigenvalues and eigenvectors. *American Math. Monthly*, 95(10):947–950, 1988.

[19] N. G. Heo. Proof without words: Pythagorean theorem. *The College Mathematics Journal*, 46, 2015.

[20] Anthony E. Hoffman. The Gram-Schmidt process is not so bad! *Mathematics Magazine*, 43(5):261–263, Nov., 1970.

[21] Linda Kaufman. The genaralized Householder transformation and sparse matrices. *Linear Algebra and Its Applications*, 90:221–234, 1987.

[22] Nasir Khattak and David J. Jeffrey. Rational orthonormal matrices. *SYNASC*, 2017.

[23] Edward G. Landauer. A method of generating Pythagorean n-tuples. *Int. J. Math. Edu. Sci. Technol.*, 10[2]:293–294, 1979.

[24] Price H. Lee. The Pythagorean tree: A new species. *https://arxiv.org/abs/0809.4324*, 2008.

[25] Hans Liebeck and Anthony Osborne. The generation of all rational orthogonal matrices. *The American Mathematical Monthly*, Vol.98, No.2,:pp.131–133, (Feb.,1991).

[26] James W. Longley and Roger D. Longley. Accuracy of gram-schmidt orthogonalization and householder transformation for the solution of linear least squares problems. *Numerical Linear Algebra with Applications*, 4(4):295–303, 1997.

[27] D.Steven Mackey, Niloufer Mackey, and Francoise Tisseur. G-reflectors: analogues of Householder transformations in scalar product spaces. *Linear Algebra and Its Applications*, 385:187–213, 2004.

[28] Johannes Middeke, David J. Jeffrey, and Christoph Koutschan. Common factors in fraction-free matrix decompositions. *Mathematics in Computer Science*, 15:589–608, 2020.

[29] Roger Nelson. Proof without words: Pythagorean quadruples. *The College Mathematics Journal*, 45(3), May,2014.

[30] J.H. Noble, M. Lubasch, and U.D. Jentschura. Genaralized householder transformations for the complex symmetric eigenvalue problem. *The European Physical Journal Plus*, 128:93, 2013.

[31] R. L. Schwaller. A method of generating Pythagorean triples. *Int. J. Math. Edu. Sci. Technol.*, 10[1]:75–77, 1979.

[32] Robert Spira. The Diophantine equation $x^2 + y^2 + z^2 = m^2$. *The American Mathematical Monthly*, 69:360—-365, 1962.

[33] John H. Staib. An alternative to the Gram-Schmidt process. *Mathematics Magazine*, 42(4):203–205, Sep., 1969.

[34] Llyod N. Trefethen and David Bau III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

[35] Wenqin Zhou and David J. Jeffrey. Fraction-free matrix factors: new forms for *LU* and *QR* factors. *Frontiers of Computer Science in China*, 2(1):67–80, 2008.

Appendix A

Matrices

- Bergrren Matrices:

$$M_1 = \begin{bmatrix} -1 & 2 & 2 \\ -2 & 1 & 2 \\ -2 & 2 & 3 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \end{bmatrix}, \quad M_3 = \begin{bmatrix} 1 & -2 & 2 \\ 2 & -1 & 2 \\ 2 & -2 & 3 \end{bmatrix}$$

- Price Matrices:

$$P_1 = \begin{bmatrix} 2 & 1 & -1 \\ -2 & 2 & 2 \\ -2 & 1 & 3 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 2 & 1 & 1 \\ 2 & -2 & 2 \\ 2 & -1 & 3 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ 2 & 1 & 3 \end{bmatrix}$$

Appendix B

Maple codes

- **Euclid's formula**

```
euclid:=proc(max) local outlist, m, n;
outlist:=[];
for m from 2 to max do
  for n from 1 + modp(m,2) by 2 to m do
    if igcd(m,n) =1 then
      outlist:= [op(outlist),[m^2 - n^2, 2nm, m^2+n^2]];
    end if;
  end do;
end do;
end proc:
```

- **Fibonacci's method**

```
fibtrip:= proc(k) local n, a, i, b, c;
if type(sqrt(k),odd) = true then
  n := k/2+1/2;
  a := sqrt(k);
  b := sqrt(add(seq(i, i = 1 .. k-1, 2)));
  c := sqrt(add(seq(i = 1.. k,2)));
  if a^2 +b^2 =c^2 then
     return c,a,b;
  end if;
else
  print("not an odd square integer:);
end if;
end proc:
```

- **Bergrren matrices**

```
bmatrices := proc(x, y, z) local a, b, c, A, PT1, PT2, PT3;
a := <<-1, -2, -2> | <2, 1, 2> | <2, 2, 3>>;
b := <<1, 2, 2> | <2, 1, 2> | <2, 2, 3>>;
c := <<1, 2, 2> | <-2, -1, -2> | <2, 2, 3>>;
A := <x, y, z>;
PT1 := a . A;
PT2 := b . A;
PT3 := c . A;
return PT1, PT2, PT3;
end proc:
```

- **Price matrices**

```
pmatrices := proc(x, y, z) local a, b, c, A, PT1, PT2, PT3;
a := <<2, -2, -2> | <1, 2, 1> | <-1, 2, 3>>;
b := <<2, 2, 2> | <1, -2, -1> | <1 2, 3>>;
c := <<2, 2, 2> | <-1, 2, 1> | <1, 2, 3>>;
A := <x, y, z>;
PT1 := a . A;
PT2 := b . A;
PT3 := c . A;
return PT1, PT2, PT3;
end proc:
```

- **Polynomial Parametrization of Pythagorean triples**

```
soph := proc(w,x,y,z) local a,b,c;
if x>0 and y>0 and z>0 and w>=0 then
  a := (x+(1-w)^(2)*x)*((y+(1+w)*z)^(2)- y^2)/2;
  b := (x+(1-w)^(2)*x)*(y+(1+w)*z)*y;
  c := (x+(1-w)^(2)*x)*((y+(1+w)*z)^(2)+ y^2)/2;
  if a^(2) + b^(2) = c^(2) then
     return a,b,c;
  end if;
end if;
end proc:
```

- **Polynomial Parametrization of Pythagorean Quintuples**

  This is the theorem presented in [16]. The authors state that there are 14 variables, but they also multiply the output by `y0` which is not included in the parameter list. In order for this code to run in Maple, `y0` must be added to the parameter list. Also, the output is `f1,f2,f3,f5,f6` and `f4` is not mentioned in the paper.

```
sophie5:=proc(w0,w12,w13,w14,w23,w24,w34,t1,t2,t3,d1,d2,d3,w4)
local f1,f2,f3,f5,f6,y0,y1,y2,y3,y4,y5,y6,y7,y8,z0,z1,z2,z3,z4,z12,
z13,z14,z23,z24,z34;

z0 := w0 + t1*w0 + t2*w0 - 2*t1*t2*w0 + t3*w0 - 2*t1*t3*w0
- t2*t3*w0 + 2*t1*t2*t3*w0 + t1*w12 - t1*t2*w12 - t1*t3*w12
+ t2*t3*w12 + t2*w13 - t1*t2*w13 + t3*w14 - t1*t3*w14
+ t1*w23 + t2*w23 - 2*t1*t2*w23 - t1*t3*w23 - t2*t3*w23
+ 2*t1*t2*t3*w23 + t1*w24 - t1*t2*w24 + t3*w24 - 2*t1*t3*w24
- t2*t3*w24 + 2*t1*t2*t3*w24 + t2*w34 - t1*t2*w34 + t3*w34
-t1*t3*w34 - 2*t2*t3*w34 + 2*t1*t2*t3*w34;
z1 :=2*d1 + t1*t2 + t3 - 2*t1*t2*t3 + w4;
z2 :=2*d2 + t1 - t1*t2 + t3 - t1*t3 - t2*t3 + 2*t1*t2*t3 + w4;
z3 :=2*d3 + t2 + t3 - t1*t3 - 2*t2*t3 + 2*t1*t2*t3 + w4;
z4 :=w4;
z12 :=w12 + t1*t2*w12 - t1*t2*t3*w12 + t1*t2*w14 - t1*t2*t3*w14
+ t1*t2*w23 - t1*t2*t3*w23 + t1*t2*w34 - t1*t2*t3*w34;
z13 :=w13 + t1*t3*w13 - t1*t2*t3*w13 + t1*t3*w14 - t1*t2*t3*w14
+ t1*t3*w23 - t1*t2*t3*w23 + t1*t3*w24 - t1*t2*t3*w24;
z14 :=w14;
z23 :=w23;
z24 :=t1*t2*t3*w12 + t1*t2*t3*w13 + w24 + t1*t2*t3*w24
+ t1*t2*t3*w34;
z34 :=w34;

y1 :=z0*z1;
y2 :=z0*z2;
y3 :=z0*z3;
y4 :=z0*z4;
y5 :=-z14*z1 - z24*z2 - z34*z3;
y6 :=z13*z1 + z23*z2 - z34*z4;
y7 :=-z12*z1 + z23*z3 + z24*z4;
y8 :=-z12*z2 - z13*z3 - z14*z4;

f1 :=2*y0*(y1*y5 + y2*y6 + y3*y7 + y4*y8);
f2 :=2*y0*(-y1*y6 + y2*y5 + y3*y8 - y4*y7);
f3 :=2*y0*(-y1*y7 - y2*y8 + y3*y5 + y4*y6);
f5 :=y0*(y1^2 + y2^2 + y3^2 + y4^2 - y5^2 - y6^2 - y7^2 - y8^2)/2;
f6 :=y0*(y1^2 + y2^2 + y3^2 + y4^2 + y5^2 + y6^2 + y7^2 + y8^2)/2;

if f1^2+f2^2+f3^2=f5*f6 then
   return f1,f2,f3,f5,f6;
end if;
end proc:
```

- **Pythagoran n-tuples or vectors**

  Calculates Pythagorean vectors of dimension d up to a maximum of m. The vectors are returned as a list of lists, with each list being a vector in one of three forms, selected by an optional argument. Option 1: $< Norm, vector >$; option 2: $< vector >$; option 3: $< vector, Norm >$. Option 3 corresponds to the usual presentation of a Pythagorean n-tuple. Option 1 is convenient for sorting the vectors by their lengths.

```
PythagVecs:=proc(d::posint,m::posint,opt::posint := 1)
local v,inc,outlist,t,vinit;
outlist := [];
v := Vector(d);
inc := 1;
do
  vinit := v[inc]+1;
  v[inc..d] := Vector(d - inc+1,fill=vinit);
  inc := d;
  do
    t := LinearAlgebra:-Norm(v,2);
    if type(t,integer) then
      if igcd(entries(v,'nolist'))=1 then
        if opt=1 then
          outlist := [op(outlist),convert(<Norm(v,2),v>,list)];
        elif opt=2 then
          outlist := [op(outlist),convert(v,list)];
        else
          outlist := [op(outlist),convert(<v,Norm(v,2)>,list)];
        end if;
      end if;
    end if;
    v[inc] := v[inc]+1;
  until m<v[inc];
  inc := inc - 1;
  while 0<inc and m<=v[inc] do
    inc := inc - 1;
  end do;
until inc=0;
return outlist;
end proc
```

- **QR matrices**

```
erichqr := proc(B) local d, sig, mut, i, j, l, m, n, a, Bt;
 m, n := LinearAlgebra:-Dimension(B);
 d := Array(0 .. n);
```

```
 mut := Array(1 .. n, 1 .. n);
 Bt := Matrix(m, n);
 d[0] := 1;
 for i to n do
   for j to i - 1 do
     sig := 0;
     for l to j - 1 do
       sig := (d[l]*sig + mut[i, l]*mut[j, l])/d[l - 1];
     end do;
     mut[i, j] := d[j-1]*((B[()..(),i]).(B[()..(),j]))- sig;
  end do;
  sig := 0;
  for l to i - 1 do
    sig := (d[l]*sig + mut[i, l]^2)/d[l - 1];
  end do;
  d[i] := d[i - 1]*((B[()..(), i]).(B[()..(), i])) - sig;
  mut[i, i] := d[i];
  a := d[1]*B[() .. (), i] - mut[i, 1]*B[() .. (), 1];
  for l to i - 2 do
    a := (d[l + 1]*a - mut[i,l + 1]*Bt[()..(),l + 1])/d[l];
  end do;
  Bt[() .. (), i] := a;
end do;
Bt[() .. (), 1] := B[() .. (), 1];
return Bt;
end proc:
```

Curriculum Vitae

Name:	Aishat Olagunju
Post-Secondary Education and Degrees:	Bowen University Iwo, Osun State, Nigeria 2014 - 2018 B.Sc.
	The University of Western Ontario London, ON, Canada 2020 - 2022 M.Sc.
Related Work Experience:	Teaching Assistant The University of Western Ontario 2021 - 2022