Maximum-Likelihood Decoding and Integer Least-Squares: The Expected Complexity

BABAK HASSIBI HARIS VIKALO

Department of Electrical Engineering
California Institute of Technology, Pasadena, CA 91125
e-mail: hassibi,hvikalo@systems.caltech.edu

April 7, 2003

Abstract

The problem of finding the least-squares solution to a system of linear equations where the unknown vector is comprised of integers, but the matrix coefficient and given vector are comprised of real numbers, arises in many applications: communications, cryptography, GPS, to name a few. The problem is equivalent to finding the closest lattice point to a given point and is known to be NP-hard. In communications applications, however, the given vector is not arbitrary, but rather is an unknown lattice point that has been perturbed by an additive noise vector whose statistical properties are known. Therefore in this paper, rather than dwell on the worst-case complexity of the integer-least-squares problem, we study its expected complexity, averaged over the noise and over the lattice. For the "sphere decoding" algorithm of Fincke and Pohst we find a closed-form expression for the expected complexity and show that, for a wide range of noise variances and dimensions, the expected complexity is polynomial, in fact often roughly cubic. Since many communications systems operate at noise levels for which the expected complexity turns out to be polynomial, this suggests that maximum-likelihood decoding, which was hitherto thought to be computationally intractable, can in fact be implemented in real-time—a result with many practical implications.

Index Terms—Integer least-squares problem, sphere decoding, wireless communications, multiple-antenna systems, lattice problems, NP hard, expected complexity, polynomial-time complexity

1 Introduction and Problem Statement

In this paper we shall be concerned with the following so-called *integer least-squares problem*

$$\min_{s \in \mathcal{Z}^m} \|x - Hs\|_2 , \tag{1}$$

where $x \in \mathcal{R}^n$, $H \in \mathcal{R}^{n\times m}$, and $\mathcal{Z}^m$ denotes the m-dimensional integer lattice, i.e., s is an m-dimensional vector with integer entries. Often, the search space is a (finite) subset of the infinite lattice, $\mathcal{D} \subset \mathcal{Z}^n$, in which case we have

$$\min_{s \in \mathcal{D} \subset \mathcal{Z}^m} \|x - Hs\|_2 . \tag{2}$$

The integer least-squares problem has a simple geometric interpretation. As the entries of s run over the integers, s spans the "rectangular" m-dimensional lattice, $\mathcal{Z}^m$. However, for any given *lattice-generating matrix* H, the n-dimensional vector Hs spans a "skewed" lattice. (When $n > m$, this skewed lattice lives in an m-dimensional subspace of $\mathcal{R}^n$.) Therefore, given the skewed lattice Hs, and given a vector $x \in \mathcal{R}^n$, the integer least-squares problem is to find the "closest" lattice point (in a Euclidean sense) to x—see Figure 1.

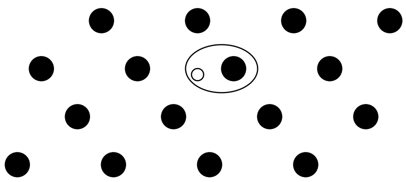

Figure 1: *Geometrical interpretation of the integer least-squares problem.*

Compared to the standard least-squares problem where the unknown vector s is an arbitrary vector in $\mathcal{R}^m$, and the solution is obtained via a simple pseudo-inverse, it is much more difficult to find the solution to (1) or (2). The main reason is that the search space is discrete (whether it is finite or infinite). In fact, it is well known that problems (1) and (2) are, for a general H, NP hard, both in a worst-case sense [1], as well as in an average sense [2].

Integer least-squares problems appear in a host of applications. In communications, when the channel is linear and the noise i.i.d. Gaussian, maximum-likelihood decoding leads to a least-squares cost. When the transmitted symbols are from a finite set, this can be often cast as an integer least-squares problem. Applications iclude lattice codes [3, 4], CDMA systems [5, 6], multi-antenna systems [7, 8, 9], etc. In all these applications, the unknown vector s represents the transmitted signal, the matrix H represents the channel, and the vector x represents the received signal. For example, in the multi-antenna context of V-BLAST [7] where we have M transmit and N receive antennas, H is the $(m = 2M) \times (n = 2N)$ real channel matrix, and for linear space-time codes (such as those in [9]) it is the equivalent channel matrix. The integer least-squares problem also arises when we have a frequency selective FIR channel $H(z) =$

$h_0 + h_1 z^{-1} + \ldots h_L z^{-L}$, in which case the channel matrix takes the form of a Toeplitz matrix

$$H = \begin{bmatrix} h_0 & & & \\ h_1 & h_0 & & \\ \vdots & \ddots & \ddots & \\ h_L & \ddots & \ddots & h_0 \\ & h_L & \ddots & h_1 \\ & & \ddots & \vdots \\ & & & h_L \end{bmatrix}. \tag{3}$$

Other applications include global positioning systems (GPS) [10] and cryptography. In fact, there is a whole family of public-key cryptosystems based on the NP-hardness of the integer least-squares problem [11, 12, 13].

The remainder of this paper is organized as follows. In section 2 we give an overview of some heuristic and exact methods to solve the integer least-squares problem. We show that the exact methods can offer substantial gains over the heuristic ones. However, they generally rqeuire an exponential worst-case complexity, whereas the heuristic methods require only cubic, $O(m^3)$, computations. Section 3 introduces the sphere decoding algorithm of Fincke and Pohst and argues that, if the received point is arbitrary, then the expected complexity of the algorithm is exponential. Section 4 introduces a random model for the integer least-squares problem, where the received point is assumed to be a lattice point perturbed by an additive Gaussian noise vector with known statistical properties. It then proceeds to compute the expected complexity of the sphere decoding algorithm, averaged over both the noise and the lattice, as a function of the noise variance, or SNR. This is done both for the infinite lattice, as well as some finite lattices encountered in communications problems. Simulations are presented in section 5, where it is shown that over a wide range of rates, SNRs and dimensions (in fact, those that are typically encountered in communications problems), the expected complexity of the sphere decoding algorithm is polynomial, often cubic. Section 6 mentions some generalizations of the Fincke-Pohst algorithm and section 7 provides the conclusion. The appendices give some mathematical background for the problems encountered in this paper.

2 Overview of Methods

Since the integer least-squares problem arises in many applications and finding the exact solution is, in general, NP hard, all practical systems employ some approximations, heuristics or combinations thereof. In communications applications, these approximations can be broadly categorized into three classes.

1. Solve the unconstrained least-squares problem to obtain $\hat{s} = H^{\dagger}x$, where $H^{\dagger}$ denotes the pseudo-inverse of H. Since the entries of $\hat{s}$ will not necessarily be integers, round them off to the closest integer (a process referred to as slicing) to obtain

$$\hat{s}_B = \left[H^{\dagger}x\right]_{\mathcal{Z}}. \tag{4}$$

   The above $\hat{s}_B$ is often called the Babai estimate [1]. In communications parlance, this procedure is referred to as *zero-forcing equalization*.

2. *Nulling and cancelling.* In this method, the Babai estimate is used for only one of the entries of s, say the first entry s_1. s_1 is then assumed to be known and its effect is cancelled out to obtain a reduced-order integer least-squares problem with $m-1$ unknowns. The process is then repeated to find s_2, etc. In communications parlance this is known as *decision-feedback equalization*.

3. *Nulling and cancelling with optimal ordering.* Nulling and cancelling can suffer from "error-propagation": if s_1 is estimated incorrectly it can have an adverse effect on the estimation of the remaining unknowns s_2, s_3, etc. To minimize the effect of error propagation, it is advantageous to perform nulling and cancelling from the "strongest" to the "weakest" signal. This is the method proposed for V-BLAST [7]—see also [14].

The above heuristic methods all require $O(m^3)$ computations, essentially because they all first solve the unconstrained least-squares problem.

2.1 Lattice Reduction

The aforementioned heuristic methods are exact only if the columns of H are orthogonal. In this case H can be diagonalized by a unitary transformation on the left, and so slicing the unconstrained least-squares solution yields the exact solution.

In practice, however, the columns of H are rarely orthogonal. Orthogonalizing the columns of H via a QR decomposition, or otherwise, generally destroys the lattice structure. (The reason being that, if s has integer entries, Rs need not have integer entries.) One method that attempts to alleviate this is *lattice reduction*. In these methods, one attempts to find an invertible $M \times M$ matrix T, such that T and T^{-1} have integer entries (thereby preserving the lattice structure), and such that the matrix $G = HT$ is as "orthogonal as possible". Having found such a T, rather than solve (1), one can solve the integer least-squares problem

$$\min_{t \in \mathcal{Z}^m} \|x - Gt\|_2 , \tag{5}$$

using the earlier meantioned heuristics and set $s = T^{-1}t$. Of course, lattice reduction is itself NP-hard. A common heuristic is the LLL (Lenstra, Lenstra and Lovasz [15]) algorithm which, permitting a gross oversimplification, can be regarded as Gram-Schmidt over integers.

While lattice reduction may lead to some improvement in the solution of (1), the integer least-squares problem over the infinite lattice, it is not useful for (2), which is over a subset of the lattice. The reason is that the lattice transmforming matrix T often destroys the properties of the subset $\mathcal{D} \subset \mathcal{Z}^n$. Since in communications applications, we are always interested in a *finite* subset of the integer lattice, we shall therefore not consider lattice reduction methods in this paper.

2.2 Exact Methods

With the abundance of heursitic methods, it is natural to ask what their performance is, and how close they come to the optimal solution? In [8] this question is studied in the context of V-BLAST where it is shown that the exact solution significantly outperforms even the best heuristics. We also give an example here in the context of space-time codes from [9], which is shown in Figure 2. The example is a rate $R = 16$ space-time code for a system with $M = 8$ transmit and $N = 4$ receive antennas,. The resulting integer least-squares problem corresponds to dimension $m = 64$ and the entries of s each take on 4 integer values, say $\{-3, -1, 1, 3\}$. Therefore the number of lattice points in $\mathcal{D}$ is $4^{64} = 2^{128} \approx 3.4 \times 10^{38}$. As can be seen from Figure 2, the BER performance of the exact is integer least-squares solution is far superior to that of the best heuristic, which in this case is nulling and cancelling with optimal ordering.[1]

The above discussion shows that there is merit in studying exact solutions. The most obvious one is

[1] Of course, at this point it may appear surprising that one can even generate Figure 2, since it requires finding the exact solution among a set of size 10^{38}—more on this later.

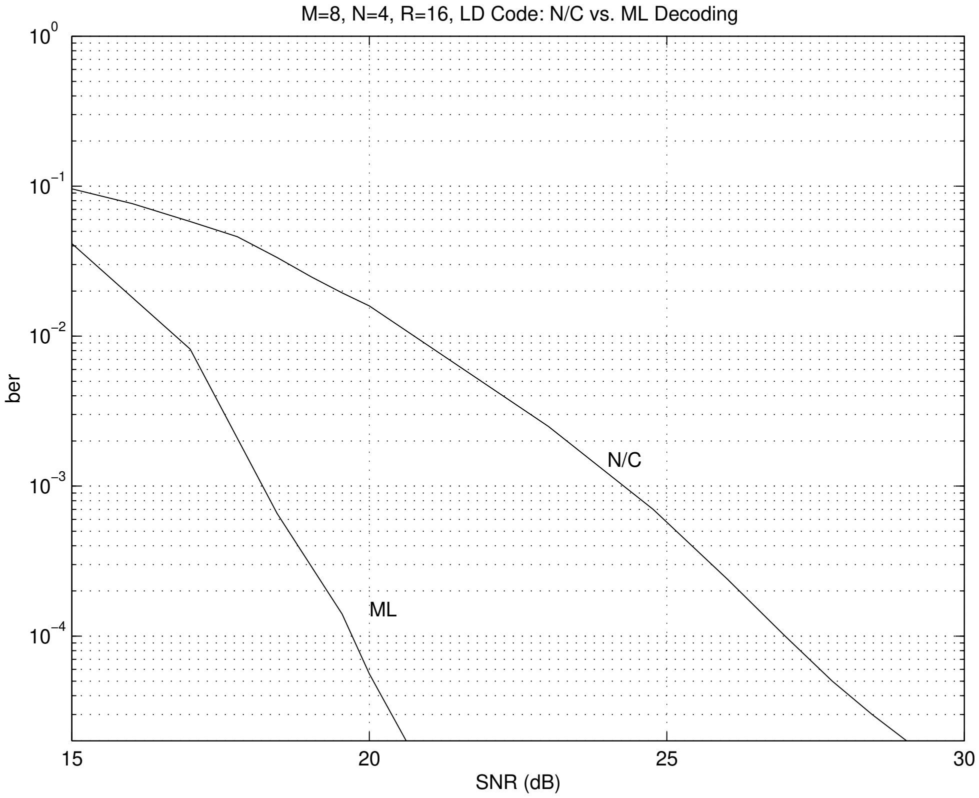

Figure 2: *Bit error performance of a rate* 16 *linear space-time code, corresponding to* $m = 64$*. Exact ML solution vs. nulling/cancelling with optimal ordering. (No. of lattice points=* $2^{128} \approx 3.4 \times 10^{38}$*).*

to form a search over the entire lattice which, although theoretically feasible for finite lattices, invariably requires an exponential search. There do, however, exist exact methods that are a bit more sophisticated than the above full search. These include Kannan's algorithm [16] (which searches only over restricted parallelograms), the KZ algorithm [17] (based on the Korkin-Zolotarev reduced basis [18]) and the sphere decoding algorithm of Fincke and Pohst [19]. Since the work of Fincke and Pohst the sphere decoding algorithm has been rediscovered in several contexts (see, e.g., [10] in the context of GPS systems) and is the algorithm we will be considering in this paper.

3 Sphere Decoding

The basic premise in sphere decoding is rather simple: we attempt to search over only lattice points that lie in a certain hypersphere of radius r around the given vector x, thereby reducing the search space and hence the required computations (see Figure 3). Clearly, the closest lattice point inside the hypersphere will also be the closest lattice point for the whole lattice. However, close scrutiny of this basic idea leads to two key questions.

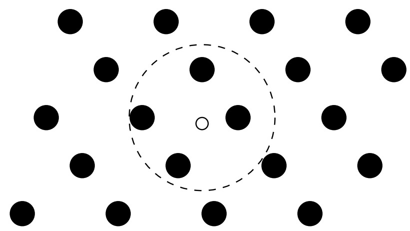

Figure 3: *Idea behind the sphere decoder*

1. *How to choose r?* Clearly, if r is too large, we obtain too many points and the search remains exponential in size, whereas if r is too small, we obtain no points inside the hypersphere.

   A natural candidate for r is the *covering radius* of the lattice, defined to be the smallest radius of spheres centered at the lattice points that cover the entire space. This is clearly the smallest radius that guarantees the existence of a point inside the hypersphere for any vector x. The problem with this choice of r is that determining the covering radius for a given lattice is itself NP hard [20].

   Another choice is to use r as the distance between the Babai estimate and the vector x, i.e., $r = \|x - H\hat{s}_B\|$, since this radius guarantees the existence of at least one lattice point (here the Babai estimate) inside the hypersphere. However, it is again not clear in general whether this choice of radius leads to too many lattice point lying inside the hypersphere.

2. *How can we tell which lattice points are inside the hypersphere?* If this requires testing the distance of each lattice point from x (to determine whether it is less than r), then there is no point in sphere decoding as we will still need an exhaustive search.

Sphere decoding does not really address the first question. However, it does propose an efficient way to answer the second, and more pressing, one. The basic observation is the following. Although it is difficult to determine the lattice points inside a general m-dimensional hypersphere, it is trivial to do so in the (one-dimensional) case of $m = 1$. The reason is that a one-dimensional hypersphere is simply an interval and so the desired lattice points will be the integer values that lie in this interval. We can use this observation to go from dimension k to $k + 1$. Suppose we have determined all k-dimensional lattice points that lie in a hypersphere of radius r. Then for any such k-dimensional point, the set of admissible values of the $k + 1$-th dimensional coordinate that lie in the higher dimensional sphere of the *same* radius r forms an interval.

The above means that we can determine all lattice points in a hypersphere of dimension m and radius r by successively determining all lattice points in hyperspheres of lower dimensions $1, 2, \ldots, m$ and the same radius r. Such an algorithm for determining the lattice points in an m-dimensional hypersphere essentially constructs a tree where the branches in the k-th level of the tree correspond to the lattice points inside the hypersphere of radius r and dimension k—see Figure 4. Moreover, the complexity of such an algorithm will depend on the *size* of the tree, i.e., on the number of lattice points visited by the algorithm in different dimensions.

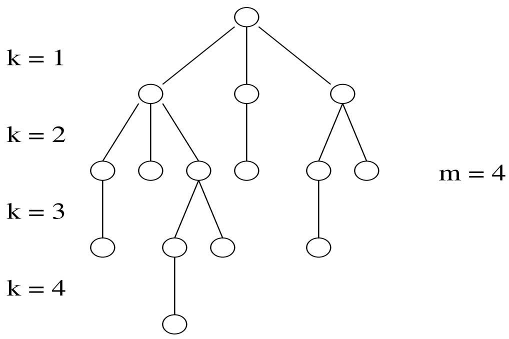

Figure 4: *Sample tree generated to determine lattice points in a 4-dimensional hypersphere.*

With this brief discussion we can now be more specific about the problem at hand. To this end, we shall assume that $n \geq m$, i.e., that there are at least as many equations as unknowns in $x \approx Hs$ (the case $n < m$

is considered further below). Note that the lattice point Hs lies in a hypersphere of radius r if, and only if,

$$r^2 \;\geq\; \|x - Hs\|^2 \,. \tag{6}$$

In order to break the problem into the subproblems described above, it is useful to consider the QR factorization of the matrix H

$$H = Q \begin{bmatrix} R \\ 0_{(n-m)\times m} \end{bmatrix}, \tag{7}$$

where R is an $m \times m$ upper triangular matrix and $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ is an $n \times n$ orthogonal matrix. The matrices Q_1 and Q_2 represent the first m and last $n - m$ orthonormal columns of Q, respectively. The condition (6) can therefore be written as

$$r^2 \;\geq\; \left\| x - \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} s \right\|^2 = \left\| \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix} x - \begin{bmatrix} R \\ 0 \end{bmatrix} s \right\|^2 = \|Q_1^* x - Rs\|^2 + \|Q_2^* x\|^2 \,.$$

Or in other words,

$$r^2 - \|Q_2^* x\|^2 \;\geq\; \|Q_1^* x - Rs\|^2 \,. \tag{8}$$

Defining $y = Q_1^* x$ and $r'^2 = r^2 - \|Q_2^* x\|^2$ allows us to rewrite this as

$$r'^2 \geq \sum_{i=1}^{m} \left(y_i - \sum_{j=i}^{m} r_{ij} s_j \right)^2 . \tag{9}$$

Here is where the upper triangular property of R comes in handy. The RHS of the above inequality can be expanded as

$$r'^2 \geq (y_m - r_{mm} s_m)^2 + (y_{m-1} - r_{m-1,m} s_m - r_{m-1,m-1} s_{m-1})^2 + \ldots \tag{10}$$

where the first term depends only on s_m, the second term on $\{s_m, s_{m-1}\}$ and so on. Therefore a necessary condition for Hs to lie inside the hypersphere is that $r'^2 \geq (y_m - r_{mm} s_m)^2$. This condition is equivalent to s_m belonging to the interval

$$\left\lceil \frac{-r' + y_m}{r_{mm}} \right\rceil \leq s_m \leq \left\lfloor \frac{r' + y_m}{r_{mm}} \right\rfloor . \tag{11}$$

Of course, (11) is by no means sufficient. For every s_m satisfying (11), defining $r'^2_{m-1} = r'^2 - (y_m - r_{mm}s_m)^2$ and $y_{m-1|m} = y_{m-1} - r_{m-1,m}s_m$, a stronger necessary condition can be found by looking at the first two terms in (10), which leads to s_{m-1} belonging to the interval

$$\left\lceil \frac{-r'_{m-1} + y_{m-1|m}}{r_{m-1,m-1}} \right\rceil \leq s_{m-1} \leq \left\lfloor \frac{r'_{m-1} + y_{m-1|m}}{r_{m-1,m-1}} \right\rfloor. \tag{12}$$

One can continue in a similar fashion for s_{m-2}, and so on until s_1, thereby obtaining all lattice points belonging to (6).

3.1 The Sphere Decoding Algorithm

We can now formalize the algorithm.

Input: $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}, R, x, y = Q_1^* x, r.$

1. Set $k = m$, $r'^2_m = r^2 - \|Q_2^* x\|^2$, $y_{m|m+1} = y_m$
2. (Bounds for s_k) Set $UB(s_k) = \lfloor \frac{r'_k + y_{k|k+1}}{r_{k-1,k-1}} \rfloor$, $s_k = \lceil \frac{-r'_k + y_{k|k+1}}{r_{k-1,k-1}} \rceil - 1$
3. (Increase s_k) $s_k = s_k + 1$. If $s_k \leq UB(s_k)$ go to 5, else to 4.
4. (Increase k) $k = k + 1$ and go to 3.
5. (Decrease k) If $k = 1$ go to 6. Else $k = k - 1$, $y_{k|k-1} = y_k + \sum_{j=k+1}^{m} r_{kj}s_j$, $r'^2_k = r'^2_{k+1} - (y_{k+1} - r_{k+1,k+1}s_{k+1})^2$.
6. Solution found. Save s_k and go to 3.

We should mention that the original paper of Fincke and Pohst [19] used slightly different notation to the one we have used. For completeness, we shall include it here. The paper [19] makes use of the unconstrained least-squares solution $\hat{s} = H^\dagger x = R^{-1}Q_1^* x$. In this case, it follows that $\|Q_2^* x\|^2 = \|x\|^2 - \|H\hat{s}\|^2$ and so inequality (8) becomes

$$r^2 - \|x\|^2 + \|H\hat{s}\|^2 \geq \|R(s - \hat{s}\|^2. \tag{13}$$

The expansion (10) becomes

$$r'^2 \geq r^2_{m-1,m-1}\left(s_{m-1} - \hat{s}_{m-1} + \frac{r_{m-1,m}}{r_{m-1,m-1}}(s_m - \hat{s}_m)\right)^2 + \dots \tag{14}$$

and the intervals (11) and (12)

$$\left\lceil \hat{s}_m - \frac{r'}{r_{mm}} \right\rceil \le s_m \le \left\lfloor \hat{s}_m + \frac{r'}{r_{mm}} \right\rfloor \tag{15}$$

and

$$\left\lceil \hat{s}_{m-1|m} - \frac{r'_{m-1}}{r_{m-1,m-1}} \right\rceil \le s_{m-1} \le \left\lfloor \hat{s}_{m-1|m} + \frac{r'_{m-1}}{r_{m-1,m-1}} \right\rfloor \tag{16}$$

respectively, where we have defined $\hat{s}_{m-1|m} = \hat{s}_{m-1} + \frac{r_{m-1,m}}{r_{m-1,m-1}}(s_m - \hat{s}_m)$. We can now alternatively write the algorithm as

Input: R, x, $\hat{s}$, r.

1a. Set $k = m$, $r'^2_m = r^2 - \|x\|^2 + \|H\hat{s}\|^2$, $\hat{s}_{m|m+1} = \hat{s}_m$

2a. (Bounds for s_k) Set $z = \frac{r'_k}{r_{kk}}$, $UB(s_k) = \lfloor z + \hat{s}_{k|k+1} \rfloor$, $s_k = \lceil -z + \hat{s}_{k|k+1} \rceil - 1$

3a. (Increase s_k) $s_k = s_k + 1$. If $s_k \le UB(s_k)$ go to 5a, else to 4a.

4a. (Increase k) $k = k + 1$ and go to 3a.

5a. (Decrease k) If $k = 1$ go to 6a. Else $k = k - 1$, $\hat{s}_{k|k-1} = \hat{s}_k + \sum_{j=k+1}^{m} \frac{r_{kj}}{r_k k}(s_j - \hat{s}_j)$, $r'^2_k = r'^2_{k+1} - r^2_{k+1,k+1}(s_{k+1} - \hat{s}_{k+1|k+2})^2$.

6a. Solution found. Save s_k and go to 3a.

3.2 A First Look at Complexity

The paper [19] gives a complexity analysis of the above algorithm. The main result is that the number of arithmetic operations of the aforementioned algorithms (excluding Steps 1, 2, 3) is at most

$$\frac{1}{6}(2m^3 + 3m^2 - 5m) + \frac{1}{2}\left((2\lfloor\sqrt{r^2 d}\rfloor + 1) \begin{pmatrix} \lfloor 4r^2 d \rfloor + m - 1 \\ \lfloor 4r^2 d \rfloor \end{pmatrix} + 1 \right), \tag{17}$$

where $d = \max(r^2_{11}, \ldots, r^2_{mm})$. In practice d grows proportional to n (r^2_{11}, for example, is simply the squared norm of the first column of H, which has n entries) and r^2 grows proportional to m (for more on this see below) and so the upper bound on the number of computations in (17) can be quite large. Our experience with numerical implementations of the algorithm shows that the bound is quite loose. Moreover, although it does depend on the lattice-generating matrix H (through the quantity d), it offers little insight into the complexity of the algorithm. We will therefore not further consider it.

In this paper we propose to study the complexity of the sphere decoding algorithm using the geometric interpretation we have developed so far. As mentioned earlier, the complexity of the sphere decoding algorithm depends on the size of the generated tree in Fig. 4, which is equal to the sum of the number of lattice points in hyperspheres of radius r and dimensions $k = 1, \ldots, m$. The size of this tree depends on the matrix H, as well as on the vector x. Therefore, unlike the complexity of solving the unconstrained least-squares problem which only depends on m and n and *not* on the specific H and x the complexity of the sphere decoding algorithm is *data-dependent*.

3.2.1 Expected Complexity

Of course since the integer least-squares problem is NP hard, the worst-case complexity of sphere decoding is exponential. However, if we assume that the matrix H and vector x are generated randomly (according to some known distributions), then the complexity of the algorithm will itself be a random variable. In this case, it is meaningful to study the expected (or average) complexity of sphere decoding, and perhaps even some of its higher order moments.[2]

In what follows we will give a rough argument for the expected complexity of sphere decoding, although it is not too difficult to make it rigorous. (For a rigorous treatment, albeit using a different approach, see [2].) For an arbitrary point x, and an arbitrary lattice H, it is not too difficult to show that the expected number of lattice points inside the k-dimensional sphere of radius r is proportional to its volume,

$$\frac{\pi^{k/2}}{\Gamma(k/2+1)} r^k.$$

Therefore the expected total number of points is

$$\sum_{k=1}^{m} \frac{\pi^{k/2}}{\Gamma(k/2+1)} r^k > \sum_{k=1}^{\frac{m}{2}} \frac{\pi^k}{\Gamma(k+1)} r^{2k} \approx e^{\pi r^2}, \text{ for large } m.$$

[2]In passing, we should mention that there is recent interest in studying the expected, rather than worst-case, complexity of various algorithms in the computer science literature. The reader may wish to refer to the survey paper [21] and the references therein, as well as the influential papers [22, 23]. In these works, a uniform distribution on the underlying problem is often (artificially) assumed and complexity issues such as NP-completeness, etc., are re-visited. However, as we shall see below, our problem allows for a very natural stochastic model.

To have a nonvanishing probability of finding a point in the m-dimensional sphere, its volume must be

$$\frac{\pi^{m/2}}{(m/2)!} r^m = O(1).$$

But from Stirling's formula this implies that $r^2 = O(m)$ and that the expected complexity of the algorithm is exponential, $e^{O(m)}$.

4 A Random Model

Although not unexpected, the above is a discouraging result. In communications applications, however, the vector x is not arbitrary, but rather is a lattice point perturbed by additive noise with known statistical properties. Thus, we will assume

$$x = Hs + v, \tag{18}$$

where the entries of v are independent $N(0, \sigma^2)$ random variables.

4.1 Choice of the Radius

The first by-product of this assumption is a method to determine the desired radius r. Note that $\frac{1}{2\sigma^2} \cdot \|v\|^2 = \frac{1}{2\sigma^2} \cdot \|x - Hs\|^2$ is a Ξ^2 random variable with $n/2$ degrees of freedom. Thus we may choose the radius to be a scaled variance of the noise,

$$r^2 = \alpha n \sigma^2,$$

in such a way that with a high probability p_{fp} we find a lattice point inside the sphere,

$$\int_0^{\alpha n/2} \frac{\lambda^{n/2-1}}{\Gamma(n/2)} e^{-\lambda} d\lambda = p_{\text{fp}},$$

where p_{fp} is set to a value close to 1, say, $p_{\text{fp}} = 0.99$. [If the point is not found, we can increase the probability p_{fp}, adjust the radius, and search again.]

The important point is that the radius r is chosen based on the statistics of the noise, and *not* based on the lattice H. Making the choice based on H quickly leads us to NP hard problems (such as determing the covering radius). Moreover, choosing the radius based on the noise has a benificial effect on the computational complexity.

4.2 Implications for Complexity

Clearly, when $\sigma^2 = 0$, i.e., when there is no noise, the exact solution can be found in $O(m^3)$ time. (The pseudo-inverse does the trick). On the other hand, when $\sigma^2 \to \infty$, the received vector x becomes arbitrary, for which we argued in section 3.2 that the expected complexity is exponential. What we are interested in is what happens at intermediate noise levels. In other words, how do we transition from cubic-time to exponential complexity?

In our analysis we shall compute the expected complexity averaged over both the noise v, as well as over the lattice-generating matrix H. Thus, we need a random model for H and will assume that it is comprised of independent $N(0,1)$ entries. This assumption is made for two reasons:

1. It makes the problem analytically tractable.
2. It is also a very reasonable assumption for large, unstructured, matrices H. (There exist many results in random matrix theory, such as Wigner's semi-circle law, mixing conditions, etc. that are not very sensitive to Gaussian assumptions–see e.g., [24].)

Of course, if H possesses special structure, such as the Toeplitz structure (3), then this is not a reasonable assumption and the structure must be explicitly taken into account. However, this merits a separate analysis and is beyond the scope of the current paper.

Now, as argued in the previous section, the complexity of sphere decoding algorithm is proportional to the number of nodes visited on the tree in Figure 4 and, consequently, to the number of points visited in the spheres of radius r and dimensions $k = 1, 2, \ldots, m$. Hence the expected complexity is proportional to the number of points in such spheres that the algorithm visits on average. Thus the expected complexity of sphere decoding algorithm is given by

$$C(m, \sigma^2) = \sum_{k=1}^{m} \underbrace{(\text{expected \# of points in } k\text{-sphere of radius } r)}_{\triangleq E_p(k, r^2 = \alpha m \sigma^2)} \cdot \underbrace{(\text{flops/point})}_{2k+17}. \tag{19}$$

The summation in (19) goes over the dimensions $k = 1$ through $k = m$. The coefficient $2k + 17$ is the number of elementary operations (additions, subtractions, and multiplications) that the Fincke-Pohst algorithm performs per each visited point in dimension k.

We need to compute $E_p(k, r^2)$, the expected number of points inside the k-dimensional hypersphere of radius r. Let us first begin with the highest dimension, i.e., $k = m$.

4.2.1 $k = m$

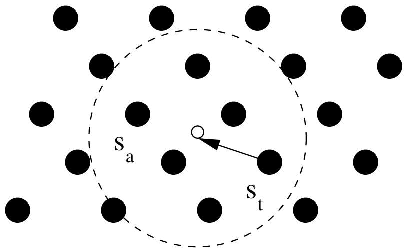

Figure 5: s_t *transmitted and* x *received. We are interested whether an arbitrary point* s_a *lies in a hypersphere of radius* r *centered around* x.

If the lattice point s_t was transmitted and the vector $x = Hs_t + v$ received, we are interested in the number of arbitrary lattice points s_a such that

$$\|x - Hs_a\|^2 \leq r^2.$$

(See Figure 5.) But, since $x = Hs_t + v$, this is just

$$\|v + H(s_t - s_a)\|^2 \leq r^2. \tag{20}$$

Now the vector $w = v + H(s_t - s_a)$ is clearly a zero-mean Gaussian random vector, since its entries are the sums of zero-mean Gaussian random variables. Now the covariance matrix has (i, j) entry

$$\begin{aligned} Ew_iw_j &= E\left(v_i + \sum_{k=1}^{m} h_{ik}(s_{t,k} - s_{a,k})\right)\left(v_j + \sum_{l=1}^{m} h_{jl}(s_{t,l} - s_{a,l})\right) \\ &= \sigma^2\delta_{ij} + \sum_{k=1}^{m}\sum_{l=1}^{m} \delta_{ij}\delta_{kl}(s_{t,k} - s_{a,k})(s_{t,l} - s_{a,l}) \\ &= \delta_{ij}(\sigma^2 + \|s_t - s_a\|^2), \quad \text{where } \|s\|^2 = \textstyle\sum_{k=1}^{m} s_k^2. \end{aligned}$$

Thus, $w = v + H(s_t - s_a)$ is an n-dimensional vector of zero-mean iid Gaussian random variables with variance $\sigma^2 + \|s_t - s_a\|^2$. This implies that $\frac{\|w\|^2}{2(\sigma^2+\|s_t-s_a\|^2)} = \frac{\|v+H(s_t-s_a)\|^2}{2(\sigma^2+\|s_t-s_a\|^2)}$ is a Ξ^2 random variable with $n/2$ degrees of freedom. Thus, the probability that the lattice point s_a lies in a hypersphere of radius r around x is

$$\gamma\left(\frac{r^2}{2(\sigma^2 + \|s_a - s_t\|^2)}, \frac{n}{2}\right) = \int_0^{\frac{r^2}{2(\sigma^2+\|s_a-s_t\|^2)}} \frac{\lambda^{n/2-1}}{\Gamma(n/2)} e^{-\lambda} d\lambda. \tag{21}$$

Now that we have computed this probability, the expected number of points in the m-dimensional hypersphere can be evaluated. However, before doing so, let us turn to the $k < m$ case.

4.2.2 $k < m$

Referring back to (10), we are interested in all k-dimensional lattice points s such that

$$r'^2 \geq \sum_{i=m-k+1}^{m} \left(y_i - \sum_{j=i}^{m} r_{ij} s_j \right)^2 . \tag{22}$$

To better understand this set, let us again consider the QR decomposition of (7) to write

$$\begin{aligned} \|x - Hs_a\|^2 = \|v + H(s_t - s_a)\|^2 &= \left\| v + Q \begin{bmatrix} R \\ 0 \end{bmatrix} (s_t - s_a) \right\|^2 \\ &= \left\| Q^* v + \begin{bmatrix} R \\ 0 \end{bmatrix} (s_t - s_a) \right\|^2 . \end{aligned}$$

Now if we partition the upper triangular matrix R and the vector $u \triangleq Q^* v$ as

$$R = \begin{bmatrix} R_{m-k,m-k} & R_{m-k,k} \\ 0 & R_{k,k} \end{bmatrix} \quad \text{and} \quad u = \begin{bmatrix} u^{m-k} \\ u^k \\ u^{n-m} \end{bmatrix} \tag{23}$$

where the block matrices $R_{m-k,m-k}$, $R_{m-k,k}$ and $R_{k,k}$ are $(m-k) \times (m-k)$, $(m-k) \times k$ and $k \times k$, respectively, and the vectors u^{m-k}, u^k and u^{n-m} are $m-k$, k and $n-m$ dimensional, respectively, then we can write

$$\begin{aligned} \|x - Hs_a\|^2 &= \|u^{m-k} + R_{m-k,m-k}(s_t^{m-k} - s_a^{m-k}) + R_{m-k,k}(s_t^k - s_a^k)\|^2 + \\ &\quad \|u^k + R_{k,k}(s_t^k - s_a^k)\|^2 + \|u^{n-m}\|^2. \end{aligned}$$

It is now straightforward to see that $r'^2 = r^2 - \|u^{n-m}\|^2$ and that $\|u^k + R_{k,k}(s_t^k - s_a^k)\|^2$ is simply the last k terms in the sum (9). Thus, we may rewrite the inequality (22) as

$$r^2 \geq \|u^k + R_{k,k}(s_t^k - s_a^k)\|^2 + \|u^{n-m}\|^2 = \left\| \begin{bmatrix} u^k \\ u^{n-m} \end{bmatrix} + \begin{bmatrix} R_{k,k} \\ 0 \end{bmatrix} (s_t^k - s_a^k) \right\|^2. \tag{24}$$

Thus, to compute the expected number of k-dimensional lattice points that satisfy (22), we need to determine the probability distribution of the RHS of (24). For this we need the following result.

Lemma 1. *Let H be an $n \times m$ (with $n \geq m$) random matrix with iid columns such that each column has a distribution that is rotationally-invariant from the left. In other words, for any $n \times n$ unitary matrix Θ, the distribution $\underline{h}_i$, the i-the column of H, satisfies*

$$p_{\underline{\mathbf{h}}}(\underline{h}_i) = p_{\underline{\mathbf{h}}}(\Theta \underline{h}_i).$$

Consider now the QR decomposition

$$H = Q \begin{bmatrix} R \\ 0 \end{bmatrix},$$

with Q $n \times n$ and unitary, and R $m \times m$ and upper triangular with non-negative diagonal entries. Then Q and R are independent random matrices, where

1. *Q has an isotropic distribution, i.e., one that is invariant under pre-multiplication by any $n \times n$ unitary matrix:*

$$p_{\mathbf{Q}}(Q) = p_{\mathbf{Q}}(\Theta Q), \quad \forall \Theta\Theta^* = \Theta^*\Theta = I.$$

2. *Consider the partitioning of R according to (23) and further partition H as*

$$H = \begin{bmatrix} H_{m-k,m-k} & H_{m-k,k} \\ H_{n-m+k,m-k} & H_{n-m+k,k} \end{bmatrix},$$

where the subscripts indicate the dimensions of the sub-matrices. Then $R_{k,k}$ has the same distribution as the R obtained from the QR decomposition of the $(n-m+k) \times k$ matrix $H_{n-m+k,k}$.

Proof: See Appendix A.

□

Remarks:

1. What is interesting about the above Lemma is that even though the $(n-m+k)\times k$ submatrix $\begin{bmatrix} R_{k,k} \\ 0 \end{bmatrix}$ is *not* the R of the QR decomposition of the $(n-m+k)\times k$ submatrix $H_{n-m+k,k}$ it has the *same* distribution.

2. Lemma 1 clearly holds for an H with iid zero-mean unit-variance Gaussian entries. In this case, one can be explicit about the distribution of R: the entries are all independent with the i-th diagonal having a Ξ^2 distribution with $(n-i+1)/2$ degrees of freedom and the strictly upper triangular entries having iid zero-mean unit-variance Gaussian distributions.

Let us now apply Lemma 1 to the problem at hand. First, since v has iid zero-mean σ^2-variance Gaussian entries and Q is unitary, the same is true of $u = Q^* v$ and also of the sub-vectors u^{m-k}, u^k and u^{n-m}. Moreover, since Q is indepedent of R, the same is true of u. Returning to the inequality (24) let us multiply the vector inside the norm by an isotropically-random unitary matrix Θ. Since this does not change norms, we have

$$r^2 \geq \left\| \Theta \begin{bmatrix} u^k \\ u^{n-m} \end{bmatrix} + \Theta \begin{bmatrix} R_{k,k} \\ 0 \end{bmatrix} (s_t^k - s_a^k) \right\|^2 .$$

Now clearly, the $n-m+k$-dimensional vector $\bar{v} = \Theta \begin{bmatrix} u^k \\ u^{n-m} \end{bmatrix}$ has iid zero-mean σ^2-variance Gaussian entries. Also, from Lemma 1 part 2, the $(n-m+k)\times k$ matrix $\bar{H} = \Theta \begin{bmatrix} R_{k,k} \\ 0 \end{bmatrix}$ has iid zero-mean unit-variance Gaussian entries. Thus, we may write (24) as

$$r^2 \geq \|\bar{v} + \bar{H}(s_t^k - s_a^k)\|^2, \tag{25}$$

which is precisely (20), except that the dimensions have changed from n and m to $n-m+k$ and k. Thus, using the same argument as presented after (20), we conclude that the probability that the k-dimensional lattice point s_a^k lies in a hypersphere of radius r around x is

$$\gamma\left(\frac{r^2}{2(\sigma^2 + \|s_a^k - s_t^k\|^2)}, \frac{n-m+k}{2}\right) = \int_0^{\frac{r^2}{2(\sigma^2+\|s_a^k - s_t^k\|^2)}} \frac{\lambda^{(n-m+k)/2-1}}{\Gamma((n-m+k)/2)} e^{-\lambda} d\lambda. \tag{26}$$

Given this probability and the one in (21), one could in principle proceed by finding the argument of the gamma function in (21) and (26) for each pair of points (s_a, s_t), and sum their contributions; however, even for a finite lattice this would clearly be a computationally formidable task (and not doable at all in the infinite lattice case). Therefore, we shall find it useful to enumerate the lattice, i.e., count the number of points with the same argument of the gamma function in (21) and (26). Enumeration of infinite and finite lattices is treated separately.

4.3 The Infinite Lattice Case

The above probability (26) depends only on the Euclidean distance between the points s_t^k and s_a^k, that is, on $\|s_a - s_t\|^2 . \|s_a^k - s_t^k\|^2 = \|s\|^2$. Now, since in an infinite integer lattice $s_a - s_t = s$ is just another lattice point, we conclude that the probability in (26) depends only on the squared norm of an arbitrary lattice point in the k-dimensional lattice. It is thus straightforward to see that the expected number of of lattice points inside a k-dimensional hypersphere of radius r is given by:

$$E_p(k, r^2) = \sum_{l=0}^{\infty} \gamma\left(\frac{r^2}{2(\sigma^2+n)}, \frac{n-m+k}{2}\right) \cdot (\text{\# of } k\text{-dimensional lattice points with } \|s^k\|^2 = l). \quad (27)$$

Since $\|s\|^2 = s_1^2 + \ldots + s_k^2$, we basically need to figure out how many ways a non-negative integer l can be represented as the sum of k squared integers. This is a classic problem in number theory and the solution is denoted by $r_k(l)$ [25]. There exist a plethora of results on how to compute $r_k(l)$. We only mention one here due to Euler: $r_k(l)$ is given by the coefficient of x^n in the expansion

$$\left(1 + 2\sum_{m=1}^{\infty} x^{m^2}\right)^k = 1 + \sum_{l=1}^{\infty} r_k(l) x^l. \quad (28)$$

(For more on the problem of representing integers as the sum of squares see Appendix B.)

The above arguments lead to the following result.

Theorem 1 (Expected complexity of sphere decoding over infinite lattice). *Consider the model*

$$x = Hs + v,$$

where $v \in \mathcal{R}^{n\times 1}$ is comprised of i.i.d. $\mathcal{N}(0, \sigma^2)$ entries, $H \in \mathcal{R}^{n\times m}$ is comprised of i.i.d. $\mathcal{N}(0, 1)$ entries, and $s \in \mathcal{Z}^m$ is an m-dimensional vector whose entries are integer numbers. Then the expected complexity of the sphere decoding algorithm of section 3.1 with a search radius r for solving the integer least-squares *problem,*

$$\min_{s \in \mathcal{Z}^m} \|x - Hs\|^2,$$

is given by

$$C(m, \sigma^2, r) = \sum_{k=1}^{m} (2k + 17) \sum_{l=0}^{\infty} \gamma\left(\frac{r^2}{2(\sigma^2 + l)}, \frac{n - m + k}{2}\right) r_k(l). \tag{29}$$

Proof: Follows from the earlier discussions.

$\square$

We should remark that, for any given search radius r, there always exists a probability that no lattice point is found. Therefore, to obtain the optimal solution, it is necessary to increase the search radius. One plasuible way of doing this is to start with a radius for which the probability of finding a point is $1 - \epsilon$, then if no point is found to increase the search radius to a value such that the probability of finding no point is $1 - \epsilon^2$, and so on. For such a strategy, we have the following result.

Corollary 1 (Expected complexity for finding the optimal solution). *Consider the setting of Theorem 1. Given any $0 < \epsilon \ll 1$, consider a strategy where we first choose a radius such that we find a lattice point with probability $1 - \epsilon$, and then increase it to a probability of $1 - \epsilon^2$, and so on, if no point is found. Then the expected complexity of the sphere decoding algorithm to find the optimal solution is given by*

$$C(m, \sigma^2, \epsilon) = \sum_{i=1}^{\infty} (1 - \epsilon)\epsilon^{i-1} \sum_{k=1}^{m} (2k + 17) \sum_{l=0}^{\infty} \gamma\left(\frac{\alpha_i m \sigma^2}{2(\sigma^2 + l)}, \frac{n - m + k}{2}\right) r_k(l), \tag{30}$$

where α_i is chosen such that

$$\gamma\left(\frac{\alpha_i n}{2}, \frac{n}{2}\right) = 1 - \epsilon^i, \quad i = 1, 2, \ldots \tag{31}$$

4.3.1 The Complex Case

In many applications, one is confronted with a complex version of the integer least-squares problem. In this case, we may assume that the model is

$$x = Hs + v, \tag{32}$$

where now $v \in \mathcal{C}^{n \times 1}$ is comprised of i.i.d. $\mathcal{CN}(0, \sigma^2)$ (circularly-symmetric complex normal) entries, $H \in \mathcal{C}^{n \times m}$ is comprised of i.i.d. $\mathcal{CN}(0, 1)$ entries, and $s \in \mathcal{CZ}^m$ is an m-dimensional complex vector whose entries have real and imaginary parts that are integers. As before, we are interested in the problem:

$$\min_{s \in \mathcal{CZ}^m} \|x - Hs\|^2. \tag{33}$$

The sphere decoding algorithm of section 3.1 can again be applied, provided we use the complex QR decomposition, and replace with real operations with appropriate complex ones. In this case, it can be shown (we are omitting the details for brevity and because they closely parallel the real case) that (29) is now replaced by

$$C(m, \sigma^2, r) = \sum_{k=1}^{m} (8k + 17) \sum_{l=0}^{\infty} \gamma \left(\frac{r^2}{\sigma^2 + l}, n - m + k \right) r_{2k}(l). \tag{34}$$

Moreover, (30) is replaced by

$$C(m, \sigma^2, \epsilon) = \sum_{i=1}^{\infty} (1 - \epsilon)\epsilon^{i-1} \sum_{k=1}^{m} (8k + 17) \sum_{l=0}^{\infty} \gamma \left(\frac{\alpha_i m \sigma^2}{(\sigma^2 + l}, n - m + k \right) r_{2k}(l), \tag{35}$$

where α_i is chosen such that

$$\gamma(\alpha_i n, n) = 1 - \epsilon^i, \quad i = 1, 2, \ldots \tag{36}$$

4.3.2 Simulation Results

As a measure of complexity, instead of the complexity itself, it is often useful to look at the *complexity exponent*, defined as

$$e_c = \frac{\log C(m, \sigma^2)}{\log m}. \tag{37}$$

When plotted, e_c is more visually appealing since complexity exponent approaches a constant if the expected complexity is polynomial, and grows like $\frac{m}{\log m}$ if $C(m, \sigma^2)$ is exponential.

The complexity exponent is plotted as a function of m for different values of σ^2 in Figure 6. As can be seen from the figure, for small enough noise the expected complexity is polynomial, indicated by the constant e_c over a wide range of m. On the other hand, for large noise e_c clearly exhibits the $\frac{m}{\log m}$ behavior and the computational complexity of the algorithm is exponential. We thus see the transition from polynomial-time to exponential complexity that we were seeking.

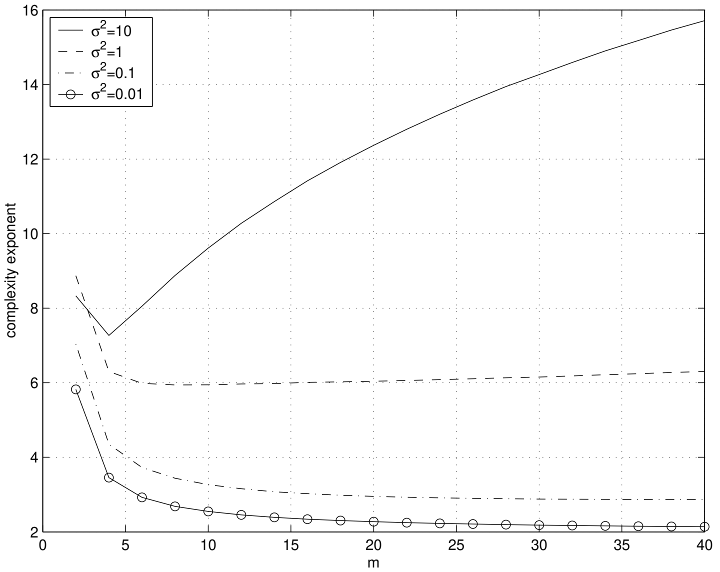

Figure 6: *The complexity exponent as a function of* m *for* $\sigma^2 = 0.01, 0.1, 1, 10$ *with* $\epsilon = .1$ *chosen for the sphere decoder.*

4.4 Finite Lattice Case

In communications problems, rather than being unbounded integers, the entries in the unknown m-dimensional vector s are often points that belong to an L-PAM constellations,

$$\mathcal{D}_L = \left\{-\frac{L-1}{2}, -\frac{L-3}{2}, \ldots, \frac{L-3}{2}, \frac{L-1}{2}\right\}. \tag{38}$$

In fact, L is often taken as power of 2. We say that the point s then belongs to the lattice $\mathcal{D}_L^m$,

$$\mathcal{D}_L^m = \underbrace{\mathcal{D}_L \times \mathcal{D}_L \times \ldots \times \mathcal{D}_L}_{m\text{-times}},$$

where $\times$-operation denotes the Cartesian product.

Furthermore, in this case, rather than the noise variance σ^2, one is interested in the signal-to-noise ratio ρ,

$$\rho = \frac{m(L^2-1)}{12\sigma^2}. \tag{39}$$

The probability expression (26) for finding an arbitrary lattice point s_a^k inside a sphere around the given point x when the point lattice s_t^k was transmitted , holds for the finite lattice case as well. However, counting lattice points which have the same argument of the gamma incomplete function in (26) is not as easy. The reason is that unlike in the infinite lattice case, the difference between two lattice points, $s_a^k - s_t^k$, is not necessarily another lattice point. Thus, the lattice enumeration that we used in the previous section needs to be performed over pairs of points, (s_a^k, s_t^k).

More formally, the number of subset lattice points in the k-dimensional sphere is given by

$$\frac{1}{L^k}\sum_l \sum_{s_t^k, s_a^k \in \mathcal{D}_L^k, \|s_t^k - s_a^k\|^2 = l} \gamma\left(\frac{\alpha m \sigma^2}{2(\sigma^2+n)}, \frac{n-m+k}{2}\right),$$

and enumerating the set

$$\left\{(s_t^k, s_a^k) \mid s_t^k, s_a^k \in \mathcal{D}_L^k, \|s_t^k - s_a^k\|^2 = l\right\},$$

appears to be complicated.

For this, we propose a modification of Euler's generating function technique. In particular, for various finite lattices we will define generating polynomials that, when combined appropriately, perform the counting operations for us.

Let us do the case study for various values of L:

1. $\mathcal{D}_2^k$: The constellation $\mathcal{D}_2^k$ consists of the corners of a k-dim hypercube, as illustrated in Figure 7. Due to symmetry, all points in the hypercube are essentially equivalent. Therefore, without loss

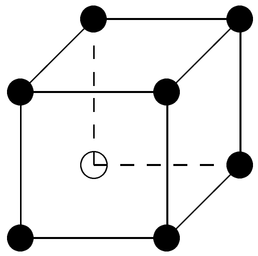

Figure 7: *Counting for* $\mathcal{D}_2^k$

of generality, we can assume that the "lower-left-corner" point $s_t^k = \left[-\frac{1}{2}, -\frac{1}{2}, \ldots, -\frac{1}{2}\right]^T$ has been transmitted. Then, depending on whether the corresponding entry of s_t^k is $-\frac{1}{2}$ or $\frac{1}{2}$, the vector $s_a^k - s_t^k$ is comprised of zero and one entries. The number of such vectors whose squared norm is l is clearly given by the number of vectors that have l entries at one, which is $\begin{pmatrix} k \\ l \end{pmatrix}$. This gives the number of points in $\mathcal{D}_2^k$ at distance l from s_t^k.

2. $\mathcal{D}_4^k$: In this case, all points in the constellation are *not* the same. For each entry of s_t^k, we can distinguish between the "corner" and the "center" points, as illustrated in Figure 8. Extending

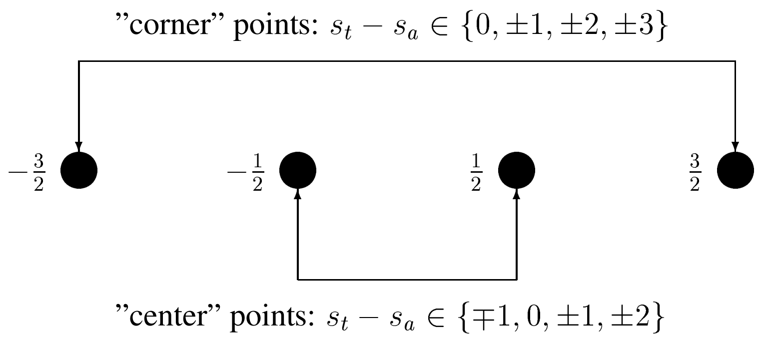

Figure 8: *Counting for* $\mathcal{D}_4^k$

Euler's idea, for the corner points we identify the generating polynomial

$$\theta_0(x) = 1 + x + x^4 + x^9, \tag{40}$$

and for the center points the polynomial

$$\theta_1(x) = 1 + 2x + x^4. \tag{41}$$

Essentially, the powers in the polynomials $\theta_0(x)$ and $\theta_1(x)$ contain information about possible squared distances between an arbitrary point s_a^k and the transmitted point s_t^k. For instance, if an entry in the transmitted vector s_t^k, say $s_{t,1}$, is a corner point, then $s_{a,1} - s_{t,1} \in \{0, \pm 1, \pm 2, \pm 3\}$, depending on $s_{a,1} \in \mathcal{D}_4$. Thus the squared norm of their difference $|s_{a,1} - s_{t,1}|^2$ can be either 0, 1, 4, or 9, as described by the powers of $\theta_0(x)$. On the other hand, if $s_{t,1}$, is a center point, then $s_{a,1} - s_{t,1} \in \{0, \pm 1, \mp 1, \pm 2\}$ (which explains coefficient 2 in front of term x in $\theta_1(x)$). Now, if among the k entries of s_t^k, we choose a corner point j times, the number of ways $\|s_t^k - s_a^k\|^2$ can add up to l is given by the coefficient of x^l in the polynomial

$$\begin{pmatrix} k \\ j \end{pmatrix} \phi_0^j(x) \phi_1^{k-j}(x). \tag{42}$$

3. $\mathcal{D}_8^k$: Note that

$$\mathcal{D}_8^k = \{-\frac{7}{2}, -\frac{5}{2}, -\frac{3}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2}\}^k.$$

Let us define the following subsets of $\mathcal{D}_8$:

$$\begin{array}{ll} \mathcal{S}_1 = \{-\frac{7}{2}, \frac{7}{2}\} & \mathcal{S}_2 = \{-\frac{5}{2}, \frac{5}{2}\} \\ \mathcal{S}_3 = \{-\frac{3}{2}, \frac{3}{2}\} & \mathcal{S}_4 = \{-\frac{1}{2}, \frac{1}{2}\} \end{array}. \tag{43}$$

Similar to the $L = 4$ case, we can identify the following polynomials for counting $s_a - s_t$ in $\mathcal{D}_8^k$ lattice:

$$\begin{aligned} \psi_0(x) &= 1 + x + x^4 + x^9 + x^{16} + x^{25} + x^{36} + x^{49}, \\ \psi_1(x) &= 1 + 2x + x^4 + x^9 + x^{16} + x^{25} + x^{36}, \\ \psi_2(x) &= 1 + 2x + 2x^4 + x^9 + x^{16} + x^{25}, \\ \psi_3(x) &= 1 + 2x + 2x^4 + 2x^9 + x^{16}. \end{aligned} \tag{44}$$

Therefore, if among k entries of s_t^k, we choose j_i points from $\mathcal{S}_i$, $i \in \{1, 2, 3, 4\}$, then the number of ways $\|s_t^k - s_a^k\|^2$ can add up to l is given by the coefficient of x^l in the polynomial

$$\begin{pmatrix} k \\ j_1, j_2, j_3, j_4 \end{pmatrix} \psi_0^{j_1}(x)\psi_1^{j_2}(x)\psi_0^{j_3}(x)\psi_1^{j_4}(x), \tag{45}$$

where $j_1 + j_2 + j_3 + j_4 = k$ and $\begin{pmatrix} k \\ j_1, j_2, j_3, j_4 \end{pmatrix} = \frac{k!}{j_1!j_2!j_3!j_4!}$.

4. Counting for $\mathcal{D}_{16}^k$ and higher order lattices is done similarly.

We can now summarize the above results for the expected computational complexity of the Fincke-Pohst algorithm for finite lattices in the following theorem.

Theorem 2. [Expected complexity of the sphere decoding over a finite lattice] *Consider the model*

$$x = Hs + v,$$

where $v \in \mathcal{R}^{n\times 1}$ is comprised of i.i.d. $\mathcal{N}(0, 1)$ entries, $H \in \mathcal{R}^{n\times m}$ is comprised of i.i.d. $\mathcal{N}(0, \rho/m)$ entries, and $s \in \mathcal{D}_L^m$ is an m-dimensional vector whose entries are elements of an L-PAM constellation. Then the expected complexity of the sphere decoding algorithm of section (3.1) with a search radius of r for *solving the integer least-squares problem*

$$\min_{s \in \mathcal{D}_L^m} \|x - Hs\|^2,$$

1. *For a 2-PAM constellation is*

$$C(m,\rho,r) = \sum_{k=1}^{m} (2k+17) \sum_{l=0}^{k} \left(\begin{array}{c} k \\ l \end{array} \right) \gamma \left(\frac{\alpha m}{2(1+\frac{12\rho l}{m(L^2-1)})}, \frac{n-m+k}{2} \right). \tag{46}$$

2. *For a 4-PAM constellation is*

$$C(m,\rho,r) = \sum_{k=1}^{m} (2k+17) \sum_{q} \frac{1}{2^k} \sum_{l=0}^{k} \left(\begin{array}{c} k \\ l \end{array} \right) g_{kl}(q) \gamma \left(\frac{\alpha m}{2(1+\frac{12\rho q}{m(L^2-1)})}, \frac{n-m+k}{2} \right), \tag{47}$$

   where $g_{kl}(q)$ is the coefficient of x^q in the polynomial

$$(1+x+x^4+x^9)^l (1+2x+x^4)^{k-l}.$$

3. *For a 8-PAM constellation is*

$$C(m,\rho,r) = \sum_{k=1}^{m} (2k+17) \sum_{q} \frac{1}{4^k} \sum_{l=0}^{k} g_{kj_1j_2j_3j_4}(q) \gamma \left(\frac{\alpha m}{2(1+\frac{12\rho q}{m(L^2-1)})}, \frac{n-m+k}{2} \right), \tag{48}$$

   where $g_{kj_1j_2j_3j_4}(q)$ is the coefficient of x^q in the polynomial (45).

4. *Similar expressions can be obtained for 16-PAM, etc., constellations.*

Proof: Follows from the above discussions.

□

We remark that to obtain the optimal solution to the integer least-squares problem we will occasionally need to increase the search radius r, and so we can obtain a result similar to that of Corollary 1, which we omit for brevity.

4.4.1 The Complex Case

When confronted with a complex integer least-squares problem, results similar to Theorem 2 hold. For complex 2-PAM (i.e., 4-QAM) constellations we have

$$C(m,\rho,r) = \sum_{k=1}^{m}(8k+17)\sum_{l=0}^{2k}\begin{pmatrix} 2k \\ l \end{pmatrix}\gamma\left(\frac{\alpha m}{1+\frac{12\rho l}{m(L^2-1)}}, n-m+k\right), \tag{49}$$

for complex 4-PAM (i.e., 16-QAM),

$$C(m,\rho,r) = \sum_{k=1}^{m}(8k+17)\sum_{q}\frac{1}{2^{2k}}\sum_{l=0}^{2k}\begin{pmatrix} 2k \\ l \end{pmatrix} g_{2kl}(q)\gamma\left(\frac{\alpha m}{1+\frac{12\rho q}{m(L^2-1)}}, n-m+k\right), \tag{50}$$

and for complex 8-PAM (i.e., 64-QAM) we have

$$C(m,\rho,r) = \sum_{k=1}^{m}(8k+17)\sum_{q}\frac{1}{4^{2k}}\sum_{l=0}^{2k} g_{2kj_1j_2j_3j_4}(q)\gamma\left(\frac{\alpha m}{1+\frac{12\rho q}{m(L^2-1)}}, n-m+k\right). \tag{51}$$

5 Simulation Results

We shall illustrate the complexity calculations with a communications example. Figure 9 shows the multiple antenna system with M-transmit and N-receive antennas.

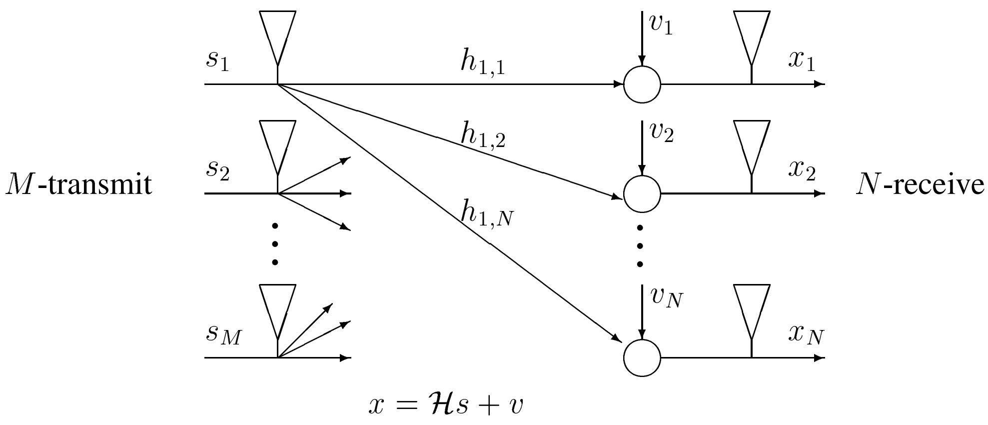

Figure 9: *Multiple antenna system*

The received signal x is related to the transmitted symbol s via

$$x = \sqrt{\frac{\rho}{M}} Hs + v, \tag{52}$$

where $H \in \mathcal{C}^{N \times M}$ is the known channel matrix, and $v \in \mathcal{C}^{N \times 1}$ is the additive noise vector, both comprised of independent, identically distributed complex-Gaussian entries $\mathcal{C}(0,1)$. Furthermore, entries in the symbol vector s are chosen from a complex-valued QAM constellation. If we assume that the entries of s and H have, on the average, unit variance, then ρ is the expected received signal-to-noise ratio (SNR) at each receive antenna. To find a real-valued equivalent to the model (9), we let $m = 2M$, $n = 2N$, and let s denote an m-dimensional real vector obtained from the M-dimensional complex vector s,

$$s = \begin{bmatrix} \mathcal{R}(s)^T & \mathcal{I}(s)^T \end{bmatrix}^T.$$

Furthermore, let $x \in \mathcal{R}^{n \times 1}$ denotes

$$x = \begin{bmatrix} \mathcal{R}(x)^T & \mathcal{I}(x)^T \end{bmatrix}^T,$$

let $v \in \mathcal{R}^{n \times 1}$ denotes

$$v = \begin{bmatrix} \mathcal{R}(v)^T & \mathcal{I}(v)^T \end{bmatrix}^T,$$

and let $H \in \mathcal{R}^{n \times m}$ be given by

$$H = \sqrt{\frac{\rho}{M}} \begin{bmatrix} \mathcal{R}(H) & \mathcal{I}(H) \\ -\mathcal{I}(H) & \mathcal{R}(H) \end{bmatrix}.$$

Then the real-valued equivalent of (9) is given by

$$x = Hs + v.$$

We consider the expected complexity of sphere decoding algorithm for signal detection in the system shown in Figure 9 for various QAM modulation schemes. The expected complexity $C(\rho, m)$ is a function of both the symbol vector size m and the SNR ρ.[3] We shall consider "snapshots" in each dimension, i.e., we keep

[3]In all the simulations presented, the complexities are for $\epsilon = .1$. In other words, our initial radius is determined so that we find a lattice point with probability .9. If no lattice point is found, we increase the radius so that this probability increases to .99, and so on.

either m or ρ variable fixed and plot the complexity as a function of the other variable.

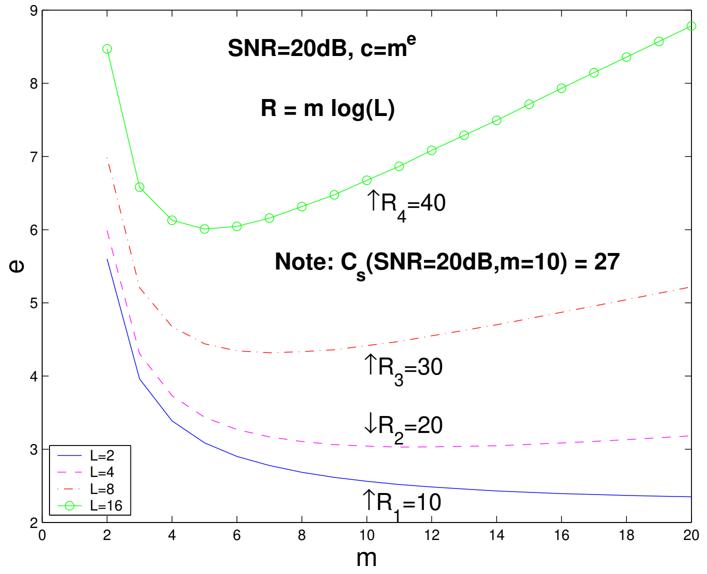

Figure 10: *The complexity exponent as a function of m for $\rho = 20db$ and $L = 2, 4, 8, 16$.*

Figure 10 shows the complexity exponent as a function of m for a fixed SNR $\rho = 20$db and L-PAM constellations with $L = 2, 4, 8, 16$. For low rates (i.e., small constellations) the expected complexity is polynomial, whereas for high rates (i.e., large constellations) it is exponential. Simulation results suggest that the complexity is polynomial as long as the rate is sufficiently, but not necessarily all that much, below the Shannon capacity corresponding to the SNR. Since this is the regime at which most communication systems operate, it suggests that ML decoding can be feasible. For instance, the complexity exponents curves in Figure 10 that correspond to $L = 8$ and $L = 16$ modulation schemes appear to be in the exponential regime. However, as is illustrated in Figure 10 for $m = 10$, the data rates corresponding to the points on those two curves are larger than the corresponding ergodic capacity,

$$C_s = E\left[\log\det\left(I_m + \frac{\rho}{M}HH^T\right)\right].$$

For instance, when $m = 10$ (and SNR $20dB$), ergodic capacity is $C_{\text{erg}} = 27$. For the same system parameters, only the rates provided by the modulation schemes corresponding to $L = 2$ and $L = 4$ ($R = 10$ and $R = 20$, respectively) can be supported by the channel. The other two modulation schemes cannot be employed (we assume uncoded transmission). Note that expected complexity exponent in the data transmission regime that is supportable by the channel complexity is roughly cubic – which, in fact, is the complexity of the heuristic techniques.

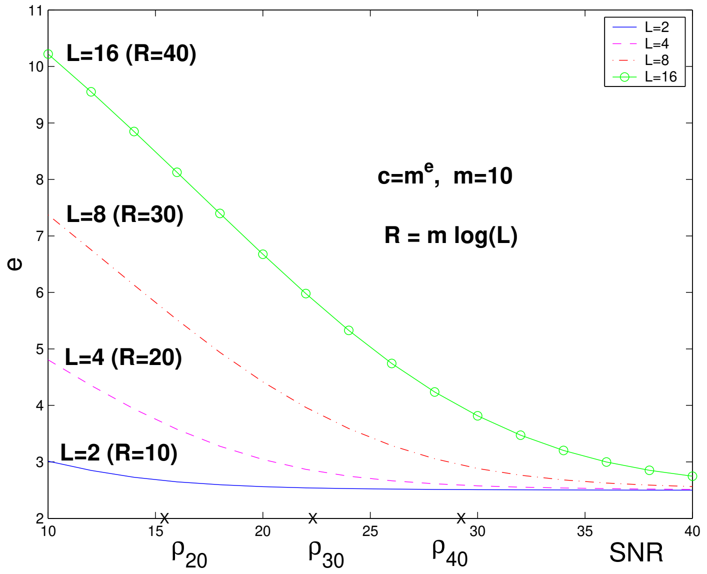

Figure 11: *The complexity exponent as a function of ρ for $m = 10$ and $L = 2, 4, 8, 16$.*

Figure 11 shows the complexity as a function of SNR for a fixed $m = 10$ (i.e., $M = N = 5$ transmit and receive antennas) and L-PAM constellations with $L = 2, 4, 8, 16$. A particular modulation scheme can be used only in the range of SNRs that supports transmission at the rate corresponding to that modulation scheme. We note that in such a range, the complexity exponent is roughly cubic. For instance, although the complexity for $L = 16$ appears to be high over a wide range of SNR, it is only for $\rho > \rho_{40}$ that this modulation scheme can be employed (ρ_{40} is the SNR for which the capacity $C_{\text{erg}} = 40 = R(L = 16)$). The complexity exponent at ρ_{40} and $L = 16$ is $c_e \approx 4$. The other SNRs marked on Figure 11, ρ_{30}, and ρ_{20}, have similar meanings (only for $L = 8$ and $L = 4$, respectively).

Figures 10-11 show the expected complexity, that is, the first-order statistics. In Figure 12, the empirical distribution of the complexity exponent is shown for $M = N = 5$ transmit and receive antennas, 16-QAM modulation scheme, and for 4 different SNR values. From Figure 11, we see that the lowest SNR in Figure 12 (16dB) roughly corresponds to the minimum SNR required for transmission on the particular system with the modulation scheme of choice. The outer dashed lines in each graph of Figure 12 correspond to three standard deviations of the corresponding distribution. The middle dashed line denotes the mean, i.e.,

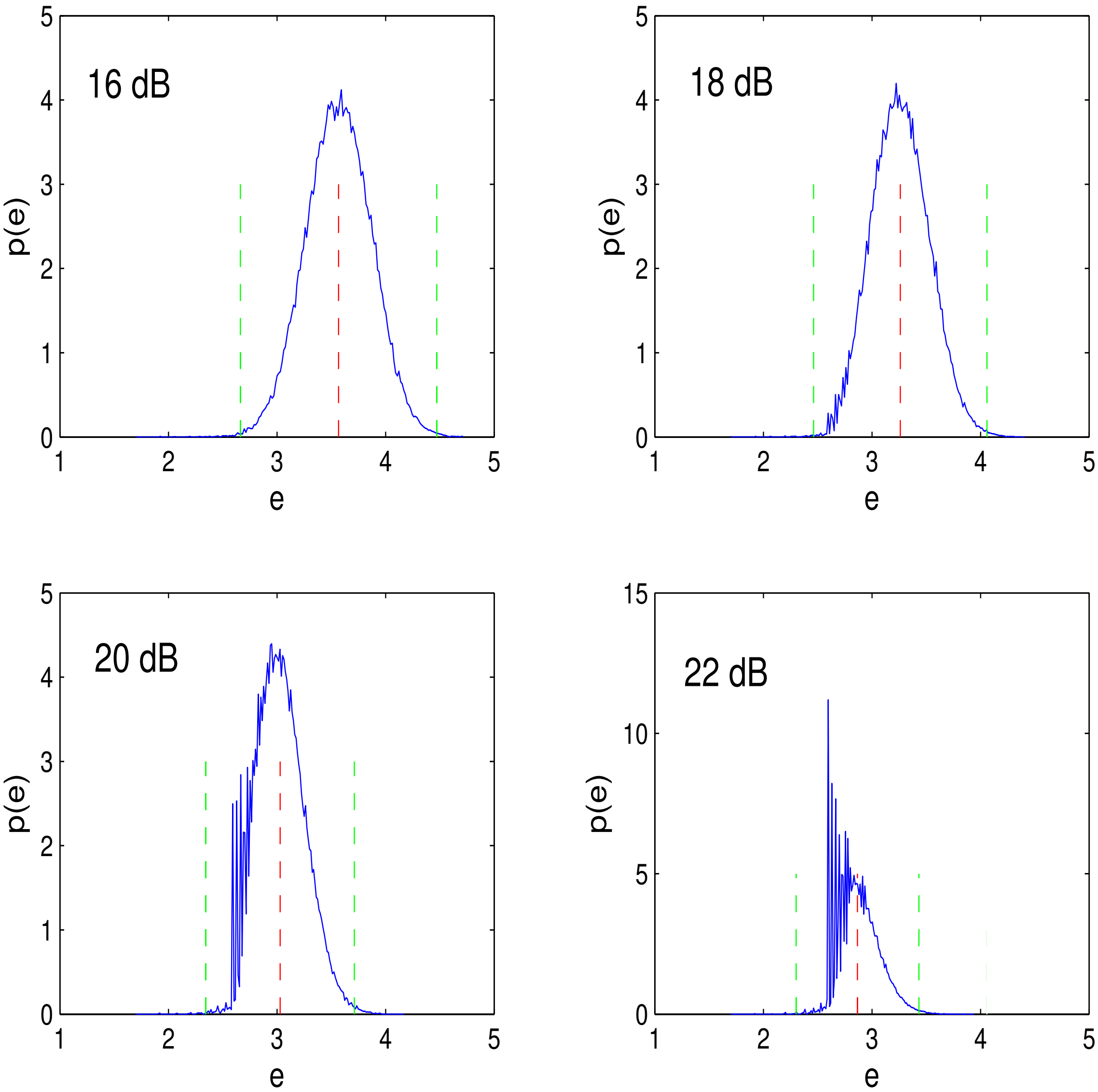

Figure 12: *The complexity exponent distribution for* $M = N = 5$*,* $L = 4$*, and* $SNR = 16, 18, 20, 22dB$*.*

the expected complexity that we previously obtained analytically. We can make the following observations in relation to the distributions as a function of the SNR:

- The expected complexity decreases, which was already anticipated from the results illustrated in Figure 11.
- The variance of the complexity decreases, as illustrated with tightening of the standard deviation.
- As the SNR increases, some "point-mass" like segments occur in the distribution. This is expected: for large SNRs, the radius of the sphere will be small and only a small (discrete) number of lattice points are found inside.

For comparison, exhaustive search in $M = N = 5$, 16-QAM system requires examining $k = 4^{10} \approx 10^6$ points, which is of the $O(m^{\log_{10} 10^6}) = O(m^6)$ order.

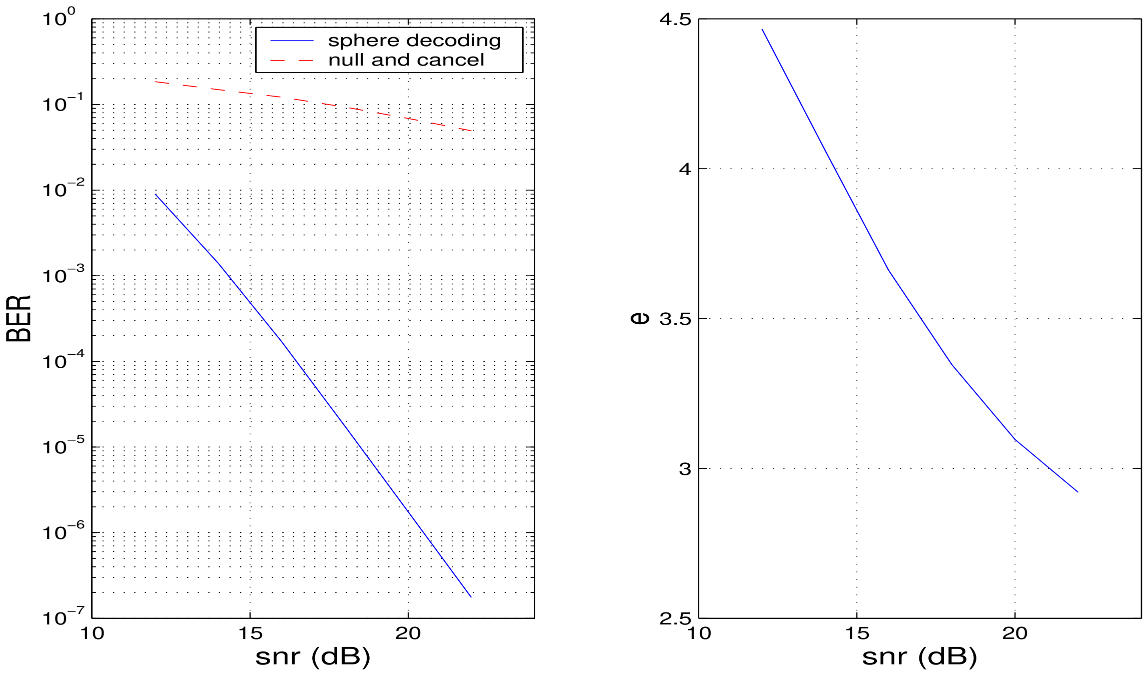

Figure 13: *Sphere decoder vs. nulling and cancelling,* $M = N = 5$, $L = 4$.

Figure 13 shows the improvement in performance of sphere decoding over nulling and cancelling for a multi-antenna system employing $M = N = 5$ transmit and receive antennas and 16-QAM modulation scheme. The complexity of ML decoding via sphere decoding here is comparable to that of nulling and cancelling, whereas the performance improvement is significant. The range of signal-to-noise ratios in Figure 13 is typical for indoor applications ([7]).

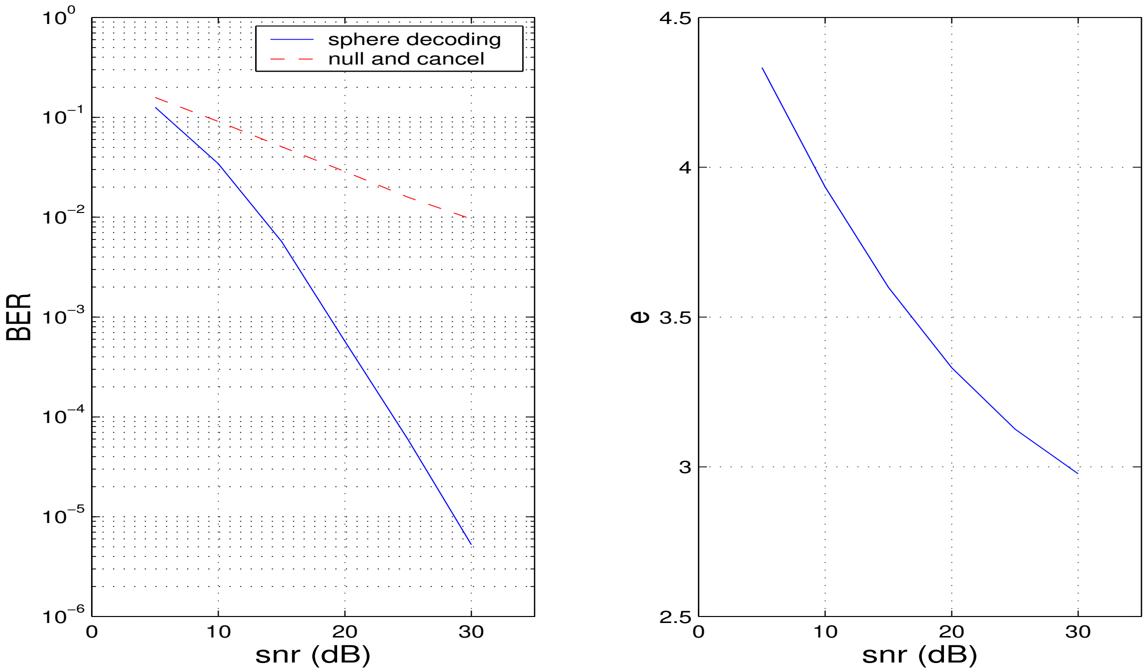

Figure 14: *Sphere decoder vs. nulling and cancelling,* $M = N = 2$*,* $L = 4$.

Figure 14 compares the performance of sphere decoding and nulling and cancelling for a multi-antenna system with $M = N = 2$ antennas, employing 16-QAM. The expected complexity exponent is also shown in Figure 14. In the SNR range of interest, the expected complexity of sphere decoding is again comparable to that of nulling and cancelling.

Comparing Figure 13 and Figure 14 we note that the performance gap between ML detection (provided by sphere decoding) and nulling and cancelling increases as the system employs more antennas. The analytical expression for the average probability of error appears difficult to derive. As an alternative, a pairwise error probability, as in [26], may be considered instead. The pairwise error probability that the vector $s^{(j)}$ was detected while $s^{(i)}$ was transmitted can be upper bounded at high SNRs as [26]

$$P(s^{(i)} \to s^{(j)}) \leq \frac{1}{\left(\frac{\rho}{4M}\right)^N \|d_{i,j}\|_F^2}, \tag{53}$$

where $d_{i,j} = s^{(i)} - s^{(j)}$. The bound (53) indicates that BER decreases exponentially with *receive diversity* –

the number of receive antennas. ML decoder, as evident from Figure 13, fully exploits the receive diversity – the slope of the BER curve implies improvement by $N = 4$ orders of magnitude per SNR decade. On the other hand, nulling and cancelling (assuming no error propagation) converts the channel into a set of parallel channels with increasing diversity [27]. However, due to the error propagation, the performance is dominated by the first stream decoded by the receiver. Thus to improve the performance of nulling and cancelling, the decoding is often ordered according to the signal-to-interference-and-noise ratio of the incoming data streams.

Finally, Figure 15 illustrates the symbol error rate performance comparison of the sphere decoding and nulling and cancelling for the system employing $M = 8$ transmit and $N = 12$ receive antennas and 16-QAM modulation, the system specifications of V-BLAST [7]. The corresponding expected complexity is sub-cubic over the entire SNR range of interest.

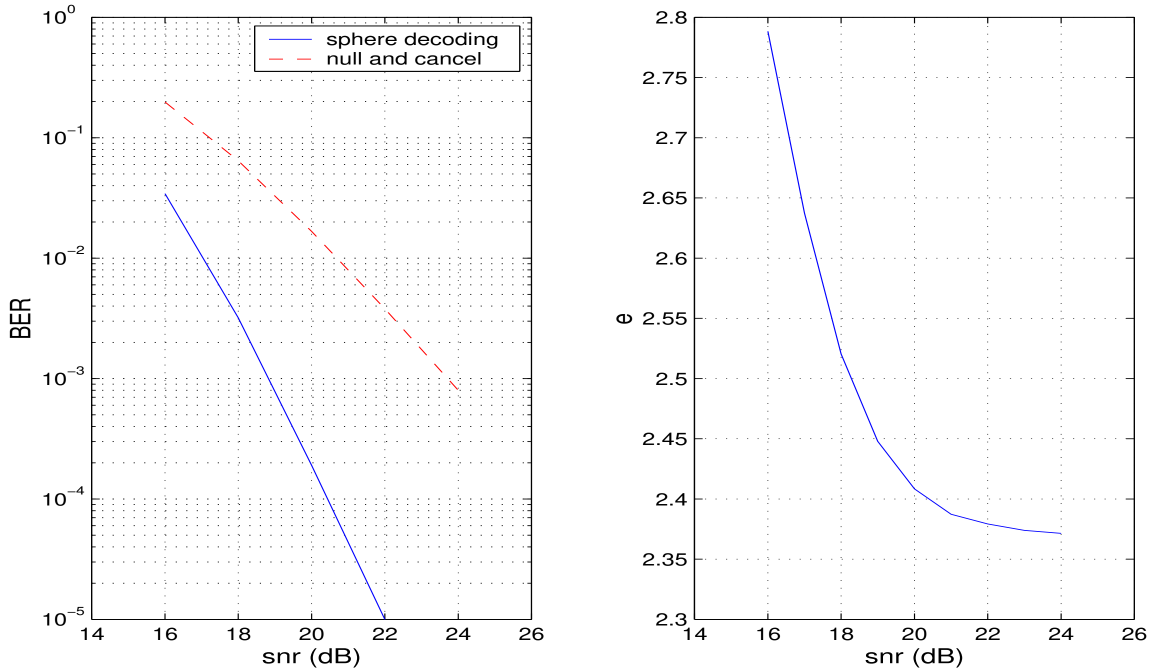

Figure 15: *Sphere decoder vs. nulling and cancelling,* $M = 8$*,* $N = 12$*, 16-QAM.*

6 Some Remarks

The expected complexity that we discussed in this paper accounts for finding all the lattice points in the sphere. The point among those found ones that is closest to x is the solution to the maximum-likelihood problem. There are some variations on the basic sphere decoding algorithm which we briefly mention here.

- *Sphere decoding with radius update.*

  Every time a point s_i in the sphere is found, we set the new radius of the sphere $r^2 = \|x - Hs_i\|^2$ and restart the algorithm. The radius update may be particularly useful at lower SNRs, where the number of points in the initial sphere is relatively large. However, it may not be benificial at high SNR, since restarting the sphere decoder may be costly. In any event, computing the expected complexity for this algorithm appears to be complicated, since it requires the calculation of the distribution of the radii that are updated.

- *Schnorr-Euchner version of the sphere decoding.*

  This strategy was proposed in [28]. The likelihood that the point will be found early is maximized if the search at each dimension k is performed in the order

  $$[\hat{s}_d], [\hat{s}_d] - 1, [\hat{s}_d] + 1, [\hat{s}_d] - 2, \ldots$$

  and if radius update (as described above) is used. The expected complexity of the Schnorr-Euchner version of the sphere decoding algorithm is no greater than the expected complexity of the basic algorithm that we derived in this chapter. However, computing the expected complexity of this algorithm appears to be formidable.

7 Conclusion

In many communication problems, maximum-likelihood detection reduces to solving an integer least-squares problem. In such applications ML detection is rarely performed, on the grounds that it requires exponential complexity and is therefore computationally intractable. In this paper we obtained a closed-form expression for the expected complexity of sphere decoding in terms of the noise variance, the dimension of the lattice, and (for subsets of lattices) the constellation. It turns out that over a wide range of noise variances and dimensions the expected complexity is often cubic or sub-cubic. Since many communications systems operate at noise levels for which this is the case, this suggests that maximum-likelihood decoding, which was hitherto thought to be computationally intractable, can in fact be implemented with complexity similar to heuristic methods, but with siginificant performance gains—a result with many practical implications.

A Proof of Lemma 1

Let us start with part 1. Since H is rotationally-invariant from the left, the matrix ΘH has the same distribution as H, for any unitary matrix Θ. Since this distribution is independent of Θ, we conclude that the same is true for any random unitary Θ that is independent of H (since the distribution of Θ is readily integrated out as $p(\Theta H)$ does not depend on Θ). In other words, for any random unitary matrix Θ, the matrix ΘH has the same distribution as H. This is true, in particular, for an isotropically random random unitary Θ.[4] Now, for such a Θ, we have

$$\Theta H = \Theta Q R,$$

from which, due to the uniqueness of the QR decomposition when R has positive diagonal entries (see, e.g., [31, 32]), we conclude that ΘQ is the Q and R remains the R in the QR decompositon of ΘH. Now, since Θ is isotropically random, ΘQ is also isotropically random and, moreover, it is independent of Q. Therefore ΘQ must be independent of R, as well. Since ΘH and H have the same distribution, the Q's in their QR decompositions must have the same distribution, from which we conclude that Q must be an isotropically random unitary matrix, independent of R.

This concludes the proof of part 1.

We remark that the prrof of part 1 only required that H be rotationally-invariant. We did not require the independence of the columns of H. This independence is required for the proof of part 2, to which we now turn our attention.

Consider the partitioning

$$H = \left[\begin{array}{cc} H_{m-k,m-k} & H_{m-k,k} \\ H_{n-m+k,m-k} & H_{n-m+k,k} \end{array} \right],$$

where the subscripts indicate the dimensions of the sub-matrices. Now consider the QR decomposition of of the leading $m-k$ columns of H, i.e.,

$$\left[\begin{array}{c} H_{m-k,m-k} \\ H_{n-m+k,m-k} \end{array} \right] = Q_1 \left[\begin{array}{c} R_{m-k,m-k} \\ 0 \end{array} \right],$$

where Q_1 is unitary and $R_{m-k,m-k}$ is upper triangular with non-negative diagonal entries. Now since $R^*_{m-k,m-k} R_{m-k,m-k} = H^*_{m-k,m-k} H_{m-k,m-k} + H^*_{m-k,k} H_{m-k,k}$, which is the leading $(m-k) \times (m-k)$

[4] For some of the properties of isotropically random unitary matrices the reader may refer to [29, 30].

submatrix of H^*H, we conclude that, as the notation suggests, $R_{m-k,m-k}$ is indeed the leading $(m-k) \times (m-k)$ submatrix of R in the partitioning (23).

Applying the unitary matrix Q_1^* to the full H (and not just its leading $m-k$ columns) we have

$$Q_1^* H = \begin{bmatrix} R_{m-k,m-k} & \bar{H}_{m-k,k} \\ 0 & \bar{H}_{n-m+k,k} \end{bmatrix}. \tag{A.1}$$

Now, since Q_1 depends only on the first $m-k$ columns of H and these are independent of the remaining k columns, by the rotational-invariance of the columns of H, we conclude that $\bar{H}_{n-m+k,k}$ has the *same* distribution as $H_{n-m+k,k}$.[5] Now if we consider the QR decomposition of $\bar{H}_{k,k}$:

$$\bar{H}_{n-m+k,k} = Q_2 \begin{bmatrix} R_{k,k} \\ 0 \end{bmatrix},$$

combining this with (A.1), we have

$$Q_1^* H = \begin{bmatrix} I & 0 \\ 0 & Q_2 \end{bmatrix} \begin{bmatrix} R_{m-k,m-k} & \bar{H}_{m-k,k} \\ 0 & R_{k,k} \\ 0 & 0 \end{bmatrix},$$

and so

$$H = Q_1 \begin{bmatrix} I & 0 \\ 0 & Q_2 \end{bmatrix} \begin{bmatrix} R_{m-k,m-k} & \bar{H}_{m-k,k} \\ 0 & R_{k,k} \\ 0 & 0 \end{bmatrix}. \tag{A.2}$$

Since $Q_1 \begin{bmatrix} I & 0 \\ 0 & Q_2 \end{bmatrix}$ is unitary and since the diagonal entries of $R_{m-k,m-k}$ and $R_{k,k}$ are non-negative, we conclude that this is indeed the QR decomposition of H (which justifies our use of the notation $R_{k,k}$ for the R in the QR of $\bar{H}_{k,k}$).[6] Since $\bar{H}_{n-m+k,k}$ and $H_{n-m+k,k}$ have the same distribution, we conclude that $R_{k,k}$ has the *same* distribution as the R obtained from the QR decomposition of $H_{n-m+k,k}$.

This concludes the proof of part 2.

[5]This is also true of $\bar{H}_{m-k,k}$ and $H_{m-k,k}$, though we shall not need this fact.

[6]Thus, for example, $\bar{H}_{m-k,k} = R_{m-k,k}$.

B Representing Integers as a Sum of Squares

The problem of representing a positive integer as the sum of squares has a long history in mathematics and number theory. The problem of determining the number of ways that a non-negative integer l can be represented as k squares was first posed by Waring in 1770 and is denoted by $r_k(l)$. [7] The first known result in this direction is due to Diophantus of Alexandria (325-409 A.D.) who showed that no integer of the form $4m + 3$ can be represented as the sum of two squares. In other words, $r_2(4m + 3) = 0$. In 1632, Girard conjectured that l is the sum of two squares if the prime divisors of l of the form $4m + 3$ occur in l in an even power. (For example, $l = 3^2 \cdot 5 = 3^2 + 6^2$, while $l = 3^3 \cdot 5$ cannot be represented as the sum of two squares.) Euler proved this conjecture in 1749. However, he did not give an explicit formula for $r_2(l)$. This was done by Legendre in 1798 and Gauss in 1801, who showed that

$$r_2(l) = 4(d_1(l) - d_3(l)), \tag{B.1}$$

where $d_1(l)$ and $d_3(l)$ are the number of divisors of l congruent to 1 and 3 mod 4, respectively.

In 1770, Lagrange proved his famous Four Squares Theorem, which states that every positive integer can be represented as the sum of four squares. This essentially establishes that $r_4(l) > 0$ for all positive integers l; however, Lagrange did not give an explicit formula for $r_4(l)$.

In terms of computing the value of $r_k(l)$, the first result is due to Euler who introduced (what is now known as) the Jacobi theta function

$$\theta(x) = \sum_{m=-\infty}^{\infty} x^{m^2} = 1 + 2\sum_{m=1}^{\infty} x^{m^2} \tag{B.2}$$

and established the following.

Theorem 3. *Let $\theta(x)$ be given by (B.2). Then*

$$\theta^k(x) = 1 + \sum_{l=1}^{\infty} r_k(l)x^l. \tag{B.3}$$

In other words, the number of ways a non-negative integer l can be represented as the sum of k squares is

[7] In fact, Waring considered the much more general problem of determining the number of ways am integer can be represented as the sum of k integers raised to the power q. In this sense, the number of ways an integer can be represented as the sum of k squares is essentially the $q = 2$ Waring problem.

given by the coefficient of x^l in the expansion of $\theta^k(x)$.

This can be illustrated as follows: $\theta^k(x)$ is clearly a series in which each term has an exponent that is obtained as the sum of k squares; since the summation in (B.2) goes over all integers, the coefficients in front of each term in the series $\theta^k(x)$ must be equal to the number of ways that the exponent in that same term can be represented as a sum of two squares.

Using the connection between the above theta function and elliptic functions, Jacobi in 1829 obtained closed-form expressions for $r_k(l)$ when $k = 2, 4, 6, 8$ (see [25], chapter 9). His formula for $k = 4$ immediately yields Lagrange's Four Squares Theorem. Solutions for $k = 10$ and $k = 12$ were found by Liouville and Eisenstein. Later, Ramanujan, Hardy, and Littlewood obtained formulas for even $k \leq 24$. For odd k, the only results are due to Dirichlet, who found $r_3(l)$, and Eisenstein, Smith, and Minkowski, who found $r_5(l)$ and $r_7(l)$.

For a long time, these were the only known explicit formulas for $r_k(l)$. Indeed, results by Glaisher, and by R. Rankin (1965), using the theory of modular forms, discouraged many researchers from obtaining further closed-form expressions. The subject was therefore presumed to be "dead" until very recently. In 1994, as a consequence of their study of certain affine super-algebras, V. Kac and M. Wakimoto conjectured formulas for $\theta^k(x)$ when $k = 4m^2$ and $k = 4m(m + 1)$ [33]. In 1996, these conjectures were proved by S. Milne using Jacobi's elliptic functions, Hankel determinants and continued fractions [34]. For an expository review of this, and subsequent results, the interested reader is referred to [35].

This exhausts known closed-form solutions for $r_k(l)$. There exist many asymptotic results (in both k and l)—see e.g., [36], chapter 5. In anycase, for any given k and l, the value of $r_k(l)$ can be numerically computed using Euler's formula (B.3). Moreover, $r_{k,l}$ is also a built-in function in *Mathematica*, `SumOf-SquaresR[k,l]` [37].

References

[1] M. Grotschel, L. Lovasz, and A. Schriver, *Geometric Algorithms and Combinatorial Optimization*. Springer Verlag, 2nd ed., 1993.

[2] M. Ajtai, "The shortest vector problem in L_2 is NP-hard for randomized reductions," *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 10–19, 1998.

[3] A. Banihashemi and A. Khandani, "On the complexity of decoding lattices using the Korkin-Zolotarev reduced basis," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 162–171, 1998.

[4] E. Agrell, A. Vardy, and K. Zeger, "Closest point search in lattices," *submitted to IEEE Transactions on Information Theory*, 2002.

[5] C. Brutel and J. Boutros, "Euclidean space lattice decoding for joint detection in CDMA systems," *Proc. of the 1999 IEEE Info. Thy. and Comm. Workshop*, p. 129, 1999.

[6] E. Viterbo and J. Boutros, "A universal lattice decoder for fading channels," *IEEE Transacstions on Information Theory*, vol. 45, pp. 1639–1642, July 2000.

[7] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs. Tech. J.*, vol. 1, no. 2, pp. 41–59, 1996.

[8] M. O. Damen, A. Chkeif, and J.-C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Comm. Let.*, pp. 161–163, May 2000.

[9] B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Info. Theory*, vol. 48, pp. 1804–1824, July 2002.

[10] A. Hassibi and S. Boyd, "Integer parameter estimation in linear models with applications to GPS," *IEEE Transactions on Signal Processing*, vol. 46, pp. 2938–52, November 1998.

[11] M. Ajtai, "Generating hard instances of lattice problems," *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 99–108, 1996.

[12] O. Goldreich, S. Goldwasser, and S. Halevi, "Public-key cryptosystems from lattice reduction problems," *Advances in Cryptology - CRYPTO97. 17th Ann. Int. Crypto. Conf.*, pp. 112–31, 1997.

[13] R. Fischlin and J. Seifert, "Tensor-based trapdoors for cvp and their application to public key cryptography," *Cryptography and Coding. 7th IMA International Conference*, pp. 244–57, 1999.

[14] B. Hassibi, "An efficient square-root algorithm for BLAST," *submitted to IEEE Trans. Sig. Proc.*, 2000. Download available at `http://mars.bell-labs.com`.

[15] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, pp. 515–534, 1982.

[16] R. Kannan, "Improved algorithms on integer programming and related lattice problems," *Proc. 15th Annu. ACM Symp. on Theory of Computing*, pp. 193–206, 1983.

[17] J. Lagarias, H. Lenstra, and C. Schnorr, "Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal," *Combinatorica*, vol. 10, pp. 333–348, 1990.

[18] A. Korkin and G. Zolotaerv, "Sur les formes quadratiques," *Math. Ann.*, vol. 6, pp. 366–389, 1873.

[19] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, pp. 463–471, April 1985.

[20] J. Conway and N. Sloane, *Sphere Packings, Lattices and Graphs*. Springer-Verlag, 1993.

[21] J. Wang, "Average-case computational complexity theory," *Complexity Theory Retrospective II*, pp. 295–328, 1997.

[22] L. Levin, "Average case complete problems," *SIAM Journal on Computing*, vol. 15, pp. 285–86, 1986.

[23] Y. Gurevich, "Average case completeness," *Journal of Computer and System Sciences*, vol. 42, no. 3, pp. 346–398, 1991.

[24] M. L. Mehta, *Random Matrices*. Academic Press, 2nd ed., 1991.

[25] G. Hardy, *Ramanujan: Twelve Lectures*. Chelsea Publishing, 1940.

[26] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Info. Theory*, vol. 44, pp. 744–765, 1998.

[27] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-time Wireless Communications*. Combiridge University Press, to appear, 2003.

[28] C. P. Schnorr and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset sum problems," *Mathematical Programming*, vol. 66, pp. 181–191, 1994.

[29] B. M. Hochwald and T. L. Marzetta, "Unitary space-time modulation for multiple-antenna communication in Rayleigh flat-fading," *IEEE Trans. Info. Theory*, vol. 46, pp. 543–564, Mar. 2000.

[30] B. Hassibi and T. L. Marzetta, "Block-fading channels and isotropically-random unitary inputs: The received signal density in closed-form," *IEEE Trans. on Info. Thy.*, vol. 48, pp. 1473–84, June 2002.

[31] R. Horn and C. Johnson, *Topics in Matrix Analysis*. Cambridge: University Press, 1991.

[32] G. Golub and C. V. Loan, *Matrix Computations*. Balitmore: Johns Hopkins University Press, 2nd ed., 1989.

[33] V. Kac and M. Wakimoto, "Integrable highest weight modules over affine super-algebras and Appell's function," *Comm. Math. Phys.*, vol. 215, no. 3, pp. 631–682, 2001.

[34] S. Milne, "Infinite families of exact sums of squares formulas, Jacobi elliptic functions, continued fractions, and Schur functions," *Ramanujan*, vol. 6, no. 1, pp. 7–149, 2002.

[35] I. Peterson, "Surprisingly square," *Science News*, vol. 159, June 2001.

[36] M. Knopp, *Modular Functions in Analytic Number Theory*. Chicago, IL: Markham Publishing Company, 1970.

[37] S. Wolfram, *The Mathematica Book*. Cambridge University Press, 4th ed., 1999.