An Algorithm for Finding Optimal Integration Lattices of Composite Order[*]

J. N. Lyness   T. Sørevik[†]
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439

Abstract

In this paper we describe a search algorithm that can be used to determine good lattices of order N when $N = N_L N_R$ has two nontrivial integer factors. This algorithm is based on relationships between an integer lattice Λ_R of order N_R and its various sublattices all of order N. Using this we have determined all the four dimensional good lattices of order 599 or less. Our list has 23 entries.

Subject Classification AMS(MOS); 65D30, 65D32
Keywords Multidimensional Quadrature, Lattice Rules, Good Lattices, Number Theoretic rules, Figure of merit, Hypercube, Periodic Functions.

1 Introduction

1.1 Background

A lattice rule is an s-dimensional quadrature rule for integration over an s-dimensional hypercube. The lattice rule is particularly useful for those smooth periodic integrands $f(x)$ whose period coincides with the dimensions of the hypercube and which have a high degree of continuity. Considerable effort has been applied to the task of constructing cost effective lattice rules. Some of the methods employed involve large scale computer searches. In this paper we shall describe a special technique which under some circumstances can shorten this task. This is based on carrying out the search in a manner in which the sublattice structure of the lattices can be

[*] This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract W-31-109-Eng-38, and by the Norwegian Council for Humanities and Science.

[†] Present address: Dept. of Informatics, University of Bergen, 5020 Bergen, Norway

exploited. In this introduction, we shall provide a very brief theoretical background to lattices, together with some of the properties deemed desirable in a cost-effective lattice rule. We also describe in simple terms the general nature of our standard search algorithm [LySø91]. This enables us, in Section 2 to highlight the underlying idea of the new algorithm in a simplified setting, and to provide an elementary cost analysis. In Section 3 we show that the symmetry of *the figure of merit* (see (2) below) can be exploited in the new algorithm in the same way as in the previous one. Section 4 written for the practitioner, deals with some of the less elegant aspects of using these algorithms. It will appear that both algorithms are useful in complementary contexts. In Section 5, we provide a table of four dimensional lattices found mainly using these techniques. All but three of these 23 lattices are new, and are improvements on presently known lattices of the corresponding order. Without loss in generality, we shall follow convention and treat integration over the unit hypercube $[0,1)^s$.

An s-dimensional lattice is an array of points in R^s. It is conveniently described by its *generator matrix* A. This is a nonsingular $s \times s$ matrix, and the elements of the lattice are integer linear combinations of the rows of A. The generator matrix of a given lattice is not unique. In fact, premultiplication of A by any unimodular matrix leaves the lattice unchanged.

A particular lattice, known as the *unit lattice* Λ_0, comprises all points all of whose components are integers. An *integration lattice* is one which contains the unit lattice as a subset. The abscissa set of a lattice rule, Q, includes all points of an integration lattice (denoted here by Λ_Q) which lie in $[0,1)^s$. The number of these points is

$$N = \mid detA \mid^{-1} \tag{1}$$

which is an integer. The rule applies the same weight $1/N$ to each of these abscissae.

The product trapezoidal rule is itself a lattice rule. A lattice rule of order N uses a subset of the N^s points which would be required by the product N-panel trapezoidal rule. Other well known lattice rules include the *number theoretic rules* (Good Lattice Points) introduced by Korobov [Kor59]. There is a large literature devoted to these rules. Niederreiter [Nie78] provides an excellent summary of the present state of the theory. The theory of lattice rules is a development of the theory of number theoretic rules [Slo85] [Nie88] and many of the definitions and properties are analogous. It appears that the discretisation error made by a lattice rule is closely related to a lattice Λ, the *reciprocal lattice* of Λ_Q whose generator matrix is $B = (A^T)^{-1}$.

It is important not to confuse the integration lattice Λ_Q whose elements are abscissae of the lattice rule with its reciprocal lattice Λ. The former, whose generator matrix has been denoted by A, contains the unit lattice Λ_0 as a sublattice. Its reciprocal lattice Λ is a sublattice of the unit lattice Λ_0 having as generator matrix an integer valued matrix $B = (A^T)^{-1}$.

In point of fact the elements of the reciprocal lattice other than the origin represent those elements of the multidimensional fourier expansion of $f(x)$ which are not integrated exactly and so contribute to the fourier expansion of the quadrature error. This circumstance can be used to justify a standard measure of the accuracy of a lattice rule. This is the *figure of merit* or *Zaremba rho index*, $\rho(\Lambda)$, of the reciprocal lattice Λ, [Zar66], defined as

$$\rho(\Lambda) = \min_{\mathbf{r}\in\Lambda,\, \mathbf{r}\neq 0} \rho(\mathbf{r}) = \min_{\mathbf{r}\in\Lambda,\, \mathbf{r}\neq 0} \prod_{i=1}^{s} \max(1, \mid r_i \mid), \tag{2}$$

where $\mathbf{r} = (r_1, r_2, ..., r_s)$. Clearly the value of $\rho(\Lambda)$ depends on the 'nearest' element of Λ which corresponds to the 'lowest' order term in the fourier expantion which is not integrated exactly. Thus the figure of merit, $\rho(\Lambda)$ is a measure of the accuracy of the quadrature rule, when applied to a naturally periodic function.

Associated with each lattice Λ is its order N. This is the inverse point density of Λ and is also the number of abscissas employed by the associated quadrature rule. We denote the set of s-dimensional integer lattices of order N by $\mathcal{L}_s(N)$. Of interest are the optimal lattices of order N. These have figures of merit

$$\rho(N) = \max_{\Lambda\in\mathcal{L}_s(N)} \rho(\Lambda). \tag{3}$$

In general, to find these, a computer aided search is required. One of the difficulties encountered immediately is that, $\nu_s(N)$, the number of lattices in $\mathcal{L}_s(N)$, is large. This is a multiplicative number theoretic function defined by

$$\nu_s(p^t) = (p^{s+t-1} - 1)/(p^t - 1); \quad p\ prime \tag{4}$$

and satisfying

$$\nu_s(MN) \leq \nu_s(M)\nu_s(N), \tag{5}$$

equality occurring if and only if $(M, N) = 1$ (i.e., M and N are mutually prime). It is shown in [LySø89] that there exists $\kappa(s)$ such that

$$(N^s - 1)/(N - 1) \leq \nu_s(N) < \kappa(s)N^{s-1}\log\log N \ \ \forall N, s > 1 \tag{6}$$

and valid values for $\kappa(s)$ include $\kappa(3) = 2.93$ and $\kappa(4) = 3.53$. $\nu_s(N)$ may be very large. For example, $\nu_4(400) = 239,834,166$ and $\nu_4(401) = 64,642,404$. In [LySø91] we have described at length a search for optimal lattices.

This search is based on representing each lattice by a generator matrix in upper triangular lattice form (*utlf*). This is essentially the Hermite normal form and is unique to the lattice.

Definition 1 *An $s \times s$ integer matrix B is in upper triangular lattice form (utlf) if and only if*

$$\begin{array}{ll} b_{ii} \geq 1 & i = 1, 2, ..., s \\ b_{ij} = 0 & 1 \leq j < i \leq s \\ 0 \leq b_{ij} < b_{ii} & otherwise\ . \end{array}$$

There is a one-to-one correspondence between the $\nu_s(N)$ distinct lattices of $\mathcal{L}_s(N)$ and the matrices B in *utlf* having $\det(B) = N$. We shall refer to a search which actually treats each of these lattices individually as a *naive lattice search*. An implementation of such a simple search is described in Section 2.2 of [LySø91]. In fact virtually all papers of practical import on this topic are devoted to accelerating the search for an optimum lattice in one way or another. One way to reduce the scope of the search is to take advantage of the symmetry of the function $\rho(\Lambda)$. Lattices obtained from one another by permuting the coordinate system have the same figure of merit. (See also Section 3 below.) A major part of [LySø91] is devoted to showing how this is exploited in our search program. It also describes a method by which the program can in some cases recognise *a priori* that the figure of merit, $\rho(\Lambda)$ of a lattice just about to be treated is less than a currently available lower bound $\rho_L(N)$ and hence can be disregarded. In the present paper, we shall show that when N can be factorised, further economies may be available.

1.2 Sublattices

In Section 2 we shall outline an algorithm for determining $\rho(N)$ based on the factorisation of N. When N can be factorised, $N = N_L N_R$, this method can significantly outperform our standard program described in [LySø91]. When N is prime, this method reduces to a somewhat ineffficent version of the standard program.

The method is based on the relationship between generator matrices of Λ and its various superlattices. This is treated in detail in [LySøKe91]. Only the part of the theory required for this algorithm is given here.

In what follows B (or R) is a generator matrix of the lattice Λ_B (or Λ_R) of order N_B (or N_R); unless otherwise stated, we are interested in evaluating $\rho(N)$ with $N = N_B = N_L N_R$ and, in general, when Λ_B is a sublattice of Λ_R.

Theorem 2 *Λ_B is a sublattice of Λ_R if and only if $L = BR^{-1}$ is an integer matrix.*

This fundamental and simple theorem is of course quite independent of which of the many choices for generator matrices B and R are used for Λ_B and Λ_R, respectively.

Theorem 3 *Let R_j; $j = 1, 2, ..., \nu_s(N_R)$ be generator matrices of the $\nu_s(N_R)$ distinct lattices of order N_R. Let L_i; $i = 1, 2, ..., \nu_s(N_L)$ be a complete set of distinct $s \times s$ matrices of order N_L in utlf. Then the set of $\nu_s(N_L)\nu_s(N_R)$ $s \times s$ matrices*

$$B_{i,j} = L_i R_j; \;\; 1 \leq i \leq \nu_s(N_L)\,, 1 \leq j \leq \nu_s(N_R) \tag{7}$$

includes at least one generator matrix of every lattice Λ_B of order $N_B = N_L N_R$.

This theorem is proved in [LySøKe91]. Note that the generator matrices of Λ_R need not be in *utlf*, but we shall usually use that set. The matrices L_i have no relevant interpretation in terms of lattices. When both L_i and R_j are in *utlf*, then

B_{ij} is upper triangular, but not necessarily in *utlf.* That is, it need not satisfy the third item of Definition 1 above.

Note also that, when $(N_L, N_R) = 1$, the set B_{ij} includes exactly one generator matrix corresponding to each of the $\nu_s(N_B)$ distinct lattices of order N_B. Otherwise there is duplication of a somewhat unpredictable nature, discussed in detail in [LySøKe91].

2 The Factorization Algorithm

In this section we describe the underlying mechanism of the new algorithm. Then we compare the cost, $(C_{O1} + C_{I1})$ below, of a pedestrian implementation with the cost C_{S1} of a correspondingly pedestrian implementation of the standard algorithm of [LySø91]. More realistic cost comparisons will be presented later in this paper; and, of course, in practice efficient implementations of both algorithms are used. Depending on the factors of N, one algorithm may on occasion be far more efficient than the other.

Our new algorithm is based on the following premise. Since Λ_B is a sublattice of Λ_R, every point $\mathbf{y}$ of Λ_B is also a lattice point of Λ_R. If we have available a list of points of Λ_R, then in order to find lattice points of Λ_B we need search only among the lattice points of Λ_R. If our list of lattice points of Λ_R includes a sufficiently large region, this search will find a critical point $\mathbf{y}$ of Λ_B, that is, one for which $\rho(\mathbf{y})$ (see (2) above) is least, and so provide the value of $\rho(\Lambda_B)$.

As a stand-alone procedure to find $\rho(\Lambda_B)$ for a single lattice, this approach would be grossly inefficient. But as part of an overall search for $\rho(N_B)$ it is very effective. This is because we may apply it to each B_{ij}, R_j pair. Work is required to make a list of lattice points of R_j. However, once available, this same list is used by each of $\nu_s(N_L)$ different lattices of order N_B. It appears from estimates given below, and from the timing results of our implementation, that a significant overall saving of effort may ensue.

Schematically this new search comprises an outer loop over lattices Λ_R of order N_R and, for each lattice Λ_R, an inner loop over all $\nu_s(N_L)$ lattices having generator matrices $B_{i,j} = L_i R$, where L_i includes all integer matrices in *utlf* of order N_L. For each lattice Λ_R treated in the outer loop, a set of lattice points $\mathbf{x}$ of Λ_R is constructed; this list, arranged in order of increasing $\rho(\mathbf{x})$, includes all those points for which $\rho(\mathbf{x}) < \rho_u(N_B)$; here $\rho_u(N_B)$ is an upper bound on $\rho(N_B)$. Algorithm 3 described in [LySø91] may be used here, and the length of this list is expected to be roughly $2^s N_B/N_R = 2^s N_L$. Once this list is available, the inner loop can be executed. Every lattice Λ_B treated in this inner loop is a sublattice of Λ_R. Thus every point in the lattice Λ_B is also in Λ_R. To find $\rho(\Lambda_B)$, we check each point of Λ_R to see whether it is in Λ_B. Since this list is in order of increasing $\rho(\mathbf{x})$, the first one encountered has $\rho(\mathbf{x}) = \rho(\Lambda_B)$.

To understand why one might expect this approach to be less costly than the standard approach, we first compare cost analyses of equivalently simplified implementations of both. In each, we suppose that we use the same naive lattice search of Section 2.2 of [LySø91]. This simply produces all lattices of the required order without regard to symmetric copy duplication. (See Section 3 below.) In each we suppose that we use the same pedestrian algorithm 1 of [LySø91]. This simply checks each indicated grid point in turn to see if it lies on the lattice under consideration. A crude cost analysis of this pair of simplified implementations follows.

(1) *The Standard Algorithm.* In this one examines a total of $\nu_s(N_B)$ lattices. One checks the unit lattice, point by point, and finds $\rho(\Lambda)$ after an average of N_B attempts. The total number of these attempts is then approximately $N_B\nu_s(N_B)$ and the overall cost is

$$C_{S1} = m_c \nu_s(N_B) N_B, \tag{8}$$

where m_c is the average cost of a check, that is, an attempt to see whether $\mathbf{x} \in \Lambda$.

(2) *The Factorization Algorithm.* This treats first the $\nu_s(N_R)$ lattices Λ_R. For each lattice, one has to check $2^s N_B$ points in order to provide a list of $2^s N_L$ points. This "up front" cost, the cost of the outer loop only, is

$$C_{O1} = m_c \nu_s(N_R) 2^s N_B, \tag{9}$$

which is generally much smaller than (8) above. After this, for each lattice Λ_B, one checks this list of $2^s N_L$ points in order to locate the first one to lie in Λ_B. (If N_R were unity so we were looking at each point, we would expect to check N_B points; as it is, these points are less dense, and we expect to check only $N_B/N_R = N_L$ points.) This is the major element of the cost and is

$$C_{I1} = m_c \nu_s(N_R) \nu_s(N_L) N_L. \tag{10}$$

This also is generally much smaller than (8). When $(N_L, N_R) = 1$, it is smaller by a factor of $N_L/N_B = 1/N_R$.

All other aspects of the calculations being taken to be equivalent in cost, we find

$$\frac{C_{O1} + C_{I1}}{C_{S1}} = \left(\frac{2^s}{\nu_s(N_L)} + \frac{1}{N_R} \right) \frac{\nu_s(N_L)\nu_s(N_R)}{\nu_s(N_B)}. \tag{11}$$

Note that when $(N_L, N_R) = 1$ the cost ratio in (11) is simply the expression in curved parentheses. This is usually much smaller than 1. For example, with $s = 4$ and $N = 360 = N_L N_R = 8 \cdot 45$ the ratio is $16/1395 + 1/45 \approx 1/30$.

We have to emphasise once more that these estimates are based on a crude model of the true state of affairs. However, estimates of this sort cannot be ignored. It will appear that the more sophisticated estimates of Section 4 confirm the overall picture. And this is confirmed once more in practice by examination of actual running times.