A COMPUTER SEARCH OF RANK-2 LATTICE RULES
FOR MULTIDIMENSIONAL QUADRATURE

IAN H. SLOAN AND LINDA WALSH

Abstract. For certain lattice rules of 'rank 2' it has been shown, in a recent paper, that a unique representation exists in a form suitable for computer evaluation. The present paper describes computer searches of such rules, reports results and identifies rules that appear promising for the numerical evaluation of practical multidimensional integrals.

1. Introduction

There is a continuing effort to find nonrandom sets of points in the multidimensional unit cube that are good abscissas for equal-weight multiple integration rules. This effort is roughly divided into two parts: the quasi-Monte Carlo method and the lattice method. Both methods aim to achieve faster convergence than the standard Monte Carlo method. Whereas the classical Monte Carlo method converges with order $1/\sqrt{N}$, where $N$ is the number of points, the quasi-Monte Carlo method can achieve an order $(\log N)^\gamma/N$ for some $\gamma > 0$; a full description of the quasi-Monte Carlo method can be found in [9].

Lattice methods aim to achieve still faster rates of convergence, for integrands that are suitably well behaved. The study of lattice methods was initiated by Korobov [3] with the number-theoretic good lattice method. See also [9, 1] and references therein. Recently, a much wider class of lattice rules have been defined for the integration of smooth functions over the unit $s$-dimensional cube [10, 11]. In essence, a lattice rule is a rule whose abscissas are taken from a geometrical 'lattice' which includes the integer vectors as a sublattice. Subsequent work by Sloan and Lyness [12] has established a classification of lattice rules, which introduces the concept of 'rank'. The rank takes the value 1 for the number-theoretic rules of Korobov, and the value $s$ for the $s$-dimensional trapezoidal rule, and more generally for any lattice rule that is an $n^s$ copy of another rule.

The purpose of this paper is to consider in detail certain rules of rank 2. Some preliminary work has been done by Newman and Lyness [8] in a computer study of certain rank-2 and rank-3 rules in three dimensions. We now carry out computer searches in higher dimensions, and find the 'best' such rules according to a criterion introduced for number-theoretic rules by Korobov [4]. At the same time we carry out searches of rank-1 (i.e., number-theoretic) rules, so that the relative performance of the new rules can be assessed.

Received December 8, 1988.
In §2 we first restate the main results of [12] for lattice rules, and of [5] for the particular case of rules of rank 2. We then show how to eliminate from the computer search rules that differ only by uninteresting permutations or reflections. In §3 the full details of our computer implementation of the searching procedures are given. Section 4 contains numerical results, and §5 discusses the implications of these results. A preliminary report of this work has appeared in [13].

2. Theory

2.1. Lattice rules for computer evaluation. Let \( I f \) denote the integral of \( f \) over the unit \( s \)-dimensional cube \( C^s = [0, 1]^s \),

\[
I f = \int_{C^s} f(x) \, dx.
\]

We consider only functions \( f \) which are continuous on \( C^s \), and moreover have a continuous \( 1 \)-periodic extension with respect to each variable \( x^{(1)}, \ldots, x^{(s)} \); that is, \( f \) is assumed to have the same values at points on opposite faces of the unit cube,

\[
f(x)|_{x^{(i)} = 0} = f(x)|_{x^{(i)} = 1}, \quad i = 1, \ldots, s.
\]

We may then define \( \overline{f} \), the periodic extension of \( f \), by

\[
\overline{f}(x) = f(\{x\}), \quad x \in \mathbb{R}^s,
\]

where \( \{x\} \) is the \( s \)-vector whose \( i \)-th component is the fractional part \( \{x^{(i)}\} = x^{(i)} - \lfloor x^{(i)} \rfloor \) of \( x^{(i)} \). The extension to functions without the property (2) is discussed in [12]. However, for practical applications we recommend that \( f \) be at least continuous, and preferably have a continuous first derivative. A preliminary coordinate transformation is usually needed to force \( f \) to have the desired property.

The general definition of a lattice rule, as given in [10, 11], is

\[
Qf = \frac{1}{N} \sum_{j=1}^{N} f(x_j),
\]

where \( x_1, \ldots, x_N \) are all the points of an infinite lattice that lie in the half-open unit cube. For our present purposes it is more convenient to begin with a representation established in [12]: there, it is shown that every lattice rule can be represented as a nonrepetitive expression of the form

\[
Qf = \frac{1}{n_1 \cdots n_m} \sum_{j_1 = 1}^{n_1} \cdots \sum_{j_m = 1}^{n_m} \overline{f} \left( j_1 \frac{z_1}{n_1} + \cdots + j_m \frac{z_m}{n_m} \right),
\]

where \( z_1, \ldots, z_m \) are integer vectors, and

\[
n_{i+1} \text{ divides } n_i \text{ for } i = 1, \ldots, m - 1, \quad n_m > 1.
\]

Conversely, every expression of the form (4) and (5) is a lattice rule. A key feature of nonrepetitive rules of this form is that the numbers \( m \) (the rank)
A COMPUTER SEARCH OF RANK-2 LATTICE RULES

and \( n_1, \ldots, n_m \) (the invariants) are uniquely determined. The simplest case is that of the number-theoretic rules [3],

\[
Qf = \frac{1}{N} \sum_{j=1}^{N} f \left( j \frac{p}{N} \right),
\]

for which the rank is clearly 1, and the (sole) invariant is \( N \), provided \( p \) and \( N \) have no nontrivial common factor.

Though the form (4) is convenient for computer evaluation, we do not yet have a satisfactory foundation for an efficient computer search, for two reasons:

(i) A rule of this form with no restrictions on \( z_1, \ldots, z_m \) can be repetitive, that is, there can be fewer than \( n_1 \cdots n_m \) distinct abscissas.

(ii) The vectors \( z_1, \ldots, z_m \) are far from unique, the same rule often appearing in many different guises.

In order to find an easily computable representation that leads to an efficient computer search procedure, we restrict attention to rules of rank 2.

\[ 2.2. \text{Lattice rules of rank 2.} \]

We have established already that a lattice rule of rank 2 can be written as a nonrepetitive expression of the form

\[
Qf = \frac{1}{n^2 r} \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} f \left( j_1 \frac{z_1}{nr} + j_2 \frac{z_2}{n} + k_1 \frac{y_1}{n} + k_2 \frac{y_2}{n} \right),
\]

where \( n > 1 \) and \( r \geq 1 \), and the two invariants are written as \( nr \) and \( n \) to satisfy (5). We shall assume throughout that \( n \) and \( r \) in (7) have no common prime factor, i.e., that their greatest common divisor \( (n, r) \) equals 1. Under this assumption the rank-2 rule can be rewritten in the form

\[
Qf = \frac{1}{n^2 r} \sum_{j=1}^{r} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} f \left( j \frac{z}{nr} + k_1 \frac{y_1}{n} + k_2 \frac{y_2}{n} \right),
\]

where \( z, y_1 \) and \( y_2 \) are integer vectors: for it may easily be verified that (8) becomes equivalent to (7) if \( z = z_1 \), \( y_1 = z_1 \) and \( y_2 = z_2 \). Conversely, it is shown in [5] that every nonrepetitive expression of the form (8) (with \( n > 1 \) and \( (n, r) = 1 \)) is a rank-2 rule, and hence is expressible in the form (7): for example, we may choose in (7) \( z_1 = nz + ry_1 \), \( z_2 = y_2 \).

If the components of vectors in (8) are suitably restricted, then this representation becomes unique: specifically, it is shown in [5] that the representation (8) is unique if \( z, y_1 \) and \( y_2 \) satisfy

\[
z^{(1)} = 1, \quad \begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ y_1^{(2)} \\ y_2^{(2)} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},
\]

and

\[
0 \leq z^{(i)} < r, \quad 0 \leq y_j^{(i)} < n, \quad \text{for } j \neq 1, 2, \quad i = 1, \ldots, s.
\]

Further, (9) ensures that the expression (8) is nonrepetitive.

The representation (8) is a convenient starting point for our rank-2 searches. It should be said, however, that not all rank-2 rules can be written in the form
In fact, it is shown in [5] that a rank-2 rule can be written in this way if and only if its one- and two-dimensional projections onto the \(x^{(1)}\) and \(x^{(2)}\) subspaces have invariants \(nr\) and \(nr, n\), respectively; in other words, if and only if the leading one- and two-dimensional projections have as many abscissas as possible (namely \(nr\) and \(n^2r\))—or in the language of [5], if and only if the rule has ‘full principal projections’.

There is still a further point to consider before we begin our computer search, namely that we would wish to eliminate from the list of rules to be searched any repetitions of rules that differ from each other only by uninteresting geometrical transformations.

2.3. The elimination of geometrically equivalent rules. We shall say that two rules are ‘geometrically equivalent’ if they differ only by a permutation of the variables, or by a reflection in one or more mid-planes, or by a combination of these. Since the search criterion to be described below does not distinguish between geometrically equivalent rules, it is clearly wasteful to include two or more geometrically equivalent rules in the list to be searched. We therefore seek an effective strategy for the recognition and elimination of this kind of redundancy.

The representation (8) is convenient for this purpose, as it allows the problem of geometrical equivalence to be reduced to that of the simpler rank-2 rule, with invariants \(n, n\),

\[
\tilde{Q}_f = \frac{1}{n^2} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \tilde{f}\left(k_1 \frac{y_1}{n} + k_2 \frac{y_2}{n}\right).
\]

For if \(y_1, y_2\) and \(y'_1, y'_2\) generate two nonequivalent rules \(\tilde{Q}_f\) and \(\tilde{Q}'_f\), then the corresponding rules \(Qf\) and \(Q'f\), where \((8')\) is the analogue of (8) with \(z, y_1, y_2\) replaced by \(z', y'_1, y'_2\), will also be nonequivalent. On the other hand, if \(y_1, y_2\) and \(y'_1, y'_2\) generate equivalent rules \(\tilde{Q}_f\) and \(\tilde{Q}'_f\), then it will be sufficient to include only one of these pairs in the list to be searched, provided we search over all allowed values of \(z\) in (8).

The strategy of using only the nonequivalent \(y_1, y_2\) pairs from (11) and then searching over all allowed \(z\) vectors in (8) may not eliminate all geometrical equivalences of \(Qf\). For example, if \(n = 2\), then (11) is symmetric under reflection in every mid-plane. As a result, the replacement of \(z^{(i)}\) by \(r - z^{(i)}\) in the formula (8) for \(Qf\) leads to a geometrically equivalent rule. In cases such as this, where the rule (11) is symmetric in the mid-plane \(x^{(i)} = \frac{1}{2}\), it is clearly sufficient to limit the component \(z^{(i)}\) to the range \(0 \leq z^{(i)} \leq [r/2]\). It is easy to find other situations in which the proposed strategy still leaves some geometrically equivalent pairs, but beyond some point the effort of eliminating them may not seem worthwhile.

The problem now reduces to the recognition of geometrical equivalences in the \(n^2\)-point rule (11), with the first two components of \(y_1, y_2\) fixed by (9). In the present work a computational, rather than algebraic, method was employed for this purpose. The computational procedure is based on applying the
rule (11), with the first two components of $y_1, y_2$ fixed by (9) and the other components running from 0 to $n - 1$, to the function

$$f(x) = \left( \sum_{i=1}^{s} h(x^{(i)}) \right)^s,$$

where

$$h(x) = |x - \frac{1}{2}|.$$

As $f$ is fully symmetric under permutations and reflections in mid-planes, two geometrically equivalent rules will clearly give the same value of $Qf$; and we believe (but cannot yet prove) that two nonequivalent rules will yield different values. Thus we need retain in the list of $y_1, y_2$ pairs only those that yield a value of $Qf$ not previously encountered. Table 1 gives the surviving $y_1, y_2$ pairs for some small values of $n$ and $s$.

<table>
<thead>
<tr>
<th>$s$</th>
<th>$n$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>100</td>
<td>010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>101</td>
<td>011</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1000</td>
<td>0100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>0101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1001</td>
<td>0101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>0111</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1001</td>
<td>0110</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1001</td>
<td>0111</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>100</td>
<td>010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>101</td>
<td>011</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1000</td>
<td>0100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>0101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1001</td>
<td>0101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>0111</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1001</td>
<td>0110</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1001</td>
<td>0111</td>
</tr>
</tbody>
</table>

In practice, the above procedure was modified to avoid rounding error problems arising from the use of real arithmetic, by evaluating the related integer quantity

$$n^2 \tilde{Q}F \equiv \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} F \left( k_1 \frac{y_1}{n} + k_2 \frac{y_2}{n} \right).$$
where $F(x) = (\sum_{i=1}^{s} H(x^{(i)}))^s$, and $H(x) = |2nx - n|$. The possibility of integer overflow was ignored, so that the integers in (14) were actually calculated modulo some large number (on a Vax 11/750 computer), a procedure that seemed safe enough in practice. Table 2 lists the number $N_y$ of surviving $y_1, y_2$ pairs for the cases $n = 2$ and $n = 3$ in 3 to 8 dimensions.

### Table 2
The number of geometrically nonequivalent $y_1, y_2$ pairs.

<table>
<thead>
<tr>
<th>$s$</th>
<th>$n = 2$</th>
<th>$n = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>23</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>44</td>
</tr>
</tbody>
</table>

2.4. The search criterion. A general discussion of possible criteria for assessing lattice and other rules is given in [6]. In the present work we use a standard criterion of Korobov and the number theorists (criterion (f') of [6]), namely: for $\alpha$ a fixed even positive integer (e.g., $\alpha = 2$), minimize

$$P_n = Qf_n - I f_n = Qf_n - 1,$$

where

$$f_n(x) = \phi_n(x^{(1)})\phi_n(x^{(2)})\ldots\phi_n(x^{(s)}),$$

and

$$\phi_n(x) = 1 - (-1)^{\alpha/2}(2\pi)^\alpha B_n(x)/\alpha!.$$ 

Here $B_n$ is the Bernoulli polynomial of degree $\alpha$.

The motivation lies in the form of the error expression for a lattice rule: it is shown in [11] that if the periodic extension $\overline{f}$ of the integrand in (1) has the absolutely convergent Fourier series expansion

$$\overline{f}(x) = \sum_{m \in \mathbb{Z}} a(m) e^{2\pi i m \cdot x},$$

then the lattice rule (3) corresponding to the lattice $L$ has the error

$$Qf - If = \sum_{m \in L^\perp} a(m),$$

where $L^\perp$ is the ‘dual’ lattice,

$$L^\perp = \{ m \in \mathbb{Z}^s : m \cdot x \in \mathbb{Z} \forall x \in L \},$$

and the prime indicates that the term $m = 0$ is to be omitted from the sum.
Since the particular function \( f_n \) defined by (16) has the simple Fourier series expansion
\[
\tilde{f}_n(x) = \sum_{m \in \mathbb{Z}} \frac{1}{(m_1 m_2 \cdots m_3)^r} e^{2\pi i m \cdot x},
\]
where
\[
\bar{m} = \begin{cases} 1 & \text{if } m = 0, \\ |m| & \text{if } m \neq 0, \end{cases}
\]
it follows from (15) and (19) that
\[
P_\alpha = \sum_{m \in L^\perp} \frac{1}{(m_1 m_2 \cdots m_3)^\alpha}.
\]
The interesting point is that \( P_\alpha \) is the maximum value of \( |Qf - If| \) over the set of functions \( f \) whose Fourier coefficients satisfy \( |a(m)| \leq 1/(m_1 m_2 \cdots m_3)^\alpha \): for, with \( f \) in that set, (19) gives
\[
|Qf - If| \leq \sum_{m \in L^\perp} |a(m)| \leq \sum_{m \in L^\perp} (m_1 m_2 \cdots m_3)^{-\alpha} = P_\alpha,
\]
with equality being achieved if \( f = f_n \).

In essence, the point of using \( P_\alpha \) as the criterion is that every lattice rule finds the function \( f_n \) a difficult one to integrate, because the error expression (20) for the error in \( Qf_n \) involves no cancellation. As pointed out in [6], \( P_\alpha \) may be a completely inappropriate measure for nonlattice rules, because many such rules find \( f_n \) trivially easy to integrate.

For the case of the rank-1 number-theoretic rules (6), it is known (see, for example, [9]) that there exist a sequence of prime numbers \( N \) and corresponding vectors \( p \) such that
\[
P_\alpha \leq c(s, \alpha) \frac{(\log N)^{\beta(s)}}{N^\alpha},
\]
where \( c(s, \alpha) \) and \( \beta(s) \) are independent of \( N \). Such sequences are often called ‘good’ lattice rules.

The following very easy result establishes the same property for a sequence of rank-2 rules having fixed smaller invariant \( n \). Thus the judgement between rank-1 and rank-2 rules cannot be based on the notion that rank-1 rules necessarily have a better order of convergence—they do not.

**Theorem.** Given \( n > 1, s \geq 1 \) and \( \alpha > 1 \), there exists a sequence of lattice rules with rank 2 and invariants \( nr, n \), with \( (n, r) = 1 \), such that
\[
P_\alpha \leq d(n, s, \alpha) \frac{(\log N)^{\beta(s)}}{N^\alpha},
\]
where \( N = n^2 r \), and \( d(n, s, \alpha) \) and \( \beta(s) \) are independent of \( N \).

**Proof.** Corresponding to the rank-2 rule \( Qf \) in the form (8), there exists a corresponding \( r \)-point rule of rank 1,
\[
\tilde{Qf} = \frac{1}{r} \sum_{j=1}^r f\left(j \frac{Z}{r}\right).
\]
Assume \( r \) is prime and \( r > n \), implying \((n, r) = 1\). By the result (21) quoted above, there exists a sequence of choices of \( r \) and \( z \) such that

\[
\bar{P}_\alpha = Q \bar{f}_\alpha - I \bar{f}_\alpha \leq c(s, \alpha) \frac{(\log r)^\beta(s)}{r^\alpha}.
\]

Now the lattice \( L(\hat{Q}) \) corresponding to the rule \( \hat{Q} \) is clearly a subset of the lattice \( L(Q) \) corresponding to the rule \( Q \), from which it follows that \( L(Q)^\perp \subset L(\hat{Q})^\perp \). Hence, for any choice of \( y_1, y_2 \) in (8),

\[
P_\alpha = \sum_{m \in L(Q)^\perp} \frac{1}{(m_1 \cdots m_s)^\alpha} \leq \sum_{m \in L(\hat{Q})^\perp} \frac{1}{(m_1 \cdots m_s)^\alpha} = \bar{P}_\alpha
\]

\[
\leq c(s, \alpha) \frac{(\log r)^\beta(s)}{r^\alpha} \leq d(n, s, \alpha) \frac{(\log N)^\beta(s)}{N^\alpha},
\]

where \( d(n, s, \alpha) = n^{2\alpha} c(s, \alpha) \). ⊠

It may be remarked that the proof yields a poor value for the constant in (22), compared to that in (21). However, the numerical calculations later in the paper strongly suggest that this is merely an artifact of the proof, at least for small values of \( n \).

### 3. The computer search

The object is to search over geometrically nonequivalent rank-2 rules with invariants \( nr, n \) and \((n, r) = 1\), which have full principal projections. This is achieved in practice by searching over all \( y_1, y_2 \) pairs from the restricted list as described in subsection 2.3, and a set of allowed \( z \) vectors satisfying (9), (10). In each case we applied the criterion in subsection 2.4 to obtain the rules giving the smallest \( P_\alpha \) values. In practice we minimized \( P_2 \) and \( P_6 \) separately, and also performed comparable searches on rank-1 rules of the form (6), so that the relative performances of rank-1 and rank-2 rules could be quantified.

The search procedure described in this work differs from previous searches [1, 2, 7] in two main respects: first, that rank-2 rules are being searched for the first time, and second (due to the ever increasing power and improving architecture of computers) that search programs have been written specifically to take advantage of the Cyber 205 vector capabilities. A full statement of the search procedure now follows.

For the rank-2 quadrature rule in the form (8), the ingredients to be specified are the dimension \( s \), the positive integers \( n \) and \( r \) (with \((n, r) = 1\)), the pair of vectors \( y_1, y_2 \), and the vector \( z \). Given \( s \) and \( n \), the pairs \( y_1, y_2 \) are chosen as in subsection 2.3 (some examples, for small \( s \), being given in Table 1). Recall that there are \( N_\gamma \) such pairs, with \( N_\gamma \) given in Table 2. In a particular search we compute, for fixed \( s, n \) and \( r \), the values of \( P_2 \) and \( P_6 \) for all \( y_1, y_2 \) pairs, and for a set of \( z \) vectors as specified below. We retain from the search only the rules which minimize \( P_2 \) or \( P_6 \) (or both).

Since it is impossible in practice to search over all possible \( z \) vectors, we set up, for given \( s \) and \( r \), two separate search sets of \( z \) vectors, one with
components of $z$ chosen systematically, as below, and another (of equal size) with components chosen at random. The first set consists of vectors of the one-parameter (Korobov [4]) form

$$z = (1, a, a^2, \ldots, a^{s-1}) \pmod{r},$$

with $1 \leq a < r$, or, for the special case $n = 2$, $1 \leq a < r/2$. The second set consists of vectors of the form

$$z = (R^{(1)}, \ldots, R^{(s)}),$$

with $R^{(i)}$ a random variable in the range 1 to $r$. Then, for $n > 2$, the total number of $z$ vectors in each of the search sets is $(r - 1)$, and for $n = 2$ it is $(r - 1)/2$. The two sets of $z$ vectors (which we shall call ‘Korobov’ and ‘random’, respectively) were searched separately.

For comparison we also carried out searches of rank-1 rules with the same total number of points (i.e., $N = n^2r$), and with $p$ in (6) given either by

$$p = (1, a, a^2, \ldots, a^{s-1}) \pmod{N},$$

with $1 \leq a \leq N/2$, or by

$$p = (R^{(1)}, \ldots, R^{(s)}),$$

with $R^{(i)}$ a random variable in the range 1 to $N$. The number of $p$ vectors of the first (‘Korobov’) type is $\lceil N/2 \rceil$. The number of $p$ vectors of the second (‘random’) type was taken to be $N(r - 1)$ if $n > 2$, or $N(r - 1)/2$ if $n = 2$, to give the same number of ‘random’ rules searched as in the rank-2 case.

Searches were restricted to 3, 4, 5, 6, 7 and 8 dimensions, with $n = 2$ or $n = 3$, and to selected values of $N < 131070$, with $r$ such that $N = n^2r$ and $(n, r) = 1$. Higher values of $n$ were excluded after preliminary studies showed that the results with $n = 5$ were usually worse, for a given total number of points $N$, than those with $n = 3$ and (especially) $n = 2$. The restriction to $N < 131070$ originates in the architecture of the Cyber 205 computer and the consequent structure of the programs written to do the searching: the programs contain vectors of length equal to the number of abscissas in a rule—or more precisely to half that number, because of symmetry in the functions $\phi_n$ defined in (17). (The maximum vector length in the Cyber 205 is $65535$ ($= 2^{16} - 1$).)

Details of the vectorization of the searches are given in the Appendix.

Rules falling in three ‘windows’ of $N$ values were considered, as follows:

(i) 1000-point window with a full search (described in subsection 3.1);
(ii) 10000-point window with both full and ‘reduced’ searches (described in subsection 3.2);
(iii) 100000-point window with a further reduced search (described in subsection 3.3).

The searches were done in the above order. At each stage the results provided suggestions for reducing the search procedure for the next window, so as to keep the work required within acceptable limits. The ways in which the searches were reduced are stated in subsections 3.1, 3.2 and 3.3.
3.1. **1000-point window.** Rules were searched for a number of \(N\) values in the range 948 to 1052, including all the values of \(N\) that, for a given \(n\), satisfy \((n, r) = 1\) and \(N = n^2 r\) for some \(r\). With \(n = 2\) these are \(N = 948, 956, 964, \ldots, 1052\) (i.e., 14 values); with \(n = 3\) these are 954, 963, 981, 990, 1008, 1017, 1035, and 1044 (i.e., 8 values). At each \(N\), rank-1 and rank-2 rules were searched (in the latter case with \(n = 2\) or \(n = 3\), as appropriate), for both 'Korobov' and 'random' vectors \(p\) and \(z\), and (in the rank-2 case) for all of the \(N_y\) nonequivalent \(y_1, y_2\) pairs. The numbers of rules searched by this 'full search' procedure are as set out in Table 3.

### Table 3

Numbers of rules searched in the ‘full’ search used for the 1000-point window.

<table>
<thead>
<tr>
<th>Type of Rule</th>
<th>Number of Rules Searched</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Rank 2, ‘Korobov’ (z) vectors</td>
<td>(n = 2) (\frac{(r-1)}{2}N_y)</td>
</tr>
<tr>
<td>(2) Rank 2, ‘random’ (z) vectors</td>
<td>(n = 3) (\frac{(r-1)}{2}N_y)</td>
</tr>
<tr>
<td>(3) Rank 1, ‘Korobov’ (p) vectors</td>
<td>(for comparison with the rank-2 searches with (n = 2)) (\frac{N}{2})</td>
</tr>
<tr>
<td>(4) Rank 1, ‘random’ (p) vectors</td>
<td>(for comparison with the rank-2 searches with (n = 3)) (\frac{N}{2})</td>
</tr>
</tbody>
</table>

Note that there is a discrepancy between the numbers of rank-1 and rank-2 rules searched, arising from the different numbers of 'Korobov' vectors \(z\) and \(p\) when the parameter \(a\) is allowed its full range in (23) or (24). Consequently, it is not possible to search exactly the same number of rules.

As we shall see in §4, a striking conclusion from the searches in the 1000-point window was the overwhelming predominance among the 'best' rank-2 rules of rules with a particular \(y_1, y_2\) pair, namely \(y_1 = Y_1, y_2 = Y_2\), where

\[
Y_1 := (1, 0, 0, 0, \ldots, 0), \quad Y_2 := (0, 1, 0, 0, \ldots, 0),
\]

that is, with all components zero except for those fixed by (9). Whatever may be the origin of this phenomenon, it immediately suggests an empirical way of dramatically reducing the time for the rank-2 searches: namely, that instead of considering all of the \(N_y\) nonequivalent \(y_1, y_2\) pairs, one could impose from the start \(y_1 = Y_1\) and \(y_2 = Y_2\). The resulting concept of a 'reduced search' (in which the number of rules searched is exactly as in Table 3, but with \(N_y\) replaced by 1) was central to our consideration of rules in the 10000-point window, to keep computer costs within acceptable limits.
3.2. 10 000-point window. To test the validity of the reduced search strategy, we first considered in detail \( N = 9972 \), carrying out the following searches:

(i) rank-2 \( n = 2 \) full search;
(ii) rank-2 \( n = 2 \) reduced search,

where a full search is as in Table 3, and a reduced search, as above, refers to the inclusion of only the pair \( y_1 = Y_1, y_2 = Y_2 \). Thus, in a reduced search, \( N_y \) in Table 3 is replaced by 1. The results (to be discussed in §4) convinced us to rely solely on the reduced search strategy in the future.

We then investigated 8 values of \( N \) between 9972 and 10 764 chosen such that both \( n = 2 \) and \( n = 3 \) rank-2 rules exist (requiring \( N \) to be a multiple of 36, and \( N/36 \) a multiple of neither 2 nor 3). The \( N \) values satisfying these conditions are 9972, 10 116, 10 188, 10 332, 10 404, 10 548, 10 620, 10 764. For these we carried out:

(i) rank-2 \( n = 2 \) reduced search;
(ii) comparison rank-1 search;
(iii) rank-2 \( n = 3 \) reduced search;
(iv) comparison rank-1 search.

3.3. 100 000-point window. For \( N \approx 100 000 \) the search procedure was very expensive and required the full memory resources of Cyber 205, so we further reduced the searches by choosing even the parameter \( a \) in the ‘Korobov’ \( z \) vector at random (following the example of Haber [2] for the rank-1 case). For \( N = 100 044 \) we considered:

(i) \( s = 3 \) to 5, rank-2 \( n = 2 \) reduced search with ‘random’ and ‘Korobov-random’ vectors \( z \), with the number of rules of each type searched equal to \( r/20 \);
(ii) \( s = 3 \) to 8, rank-2 \( n = 2 \) reduced search with ‘random’ and ‘Korobov-random’ vectors \( z \), with the number of rules of each type searched equal to \( r/200 \).

And for \( N = 131 004 \) we considered:

(iii) \( s = 7 \) and 8, rank-2 \( n = 2 \) reduced search with ‘random’ and ‘Korobov-random’ vectors \( z \), with the number of rules of each type searched equal to \( r/200 \).

4. Results of the computer searches

For each of the searches described in §3, the ‘best’ rules, as judged by \( P_2 \) and \( P_6 \) separately, are recorded in the microfiche supplement at the end of this issue.

Here we attempt to give an overall view of the results so obtained, and at the end select (in Tables 4, 5 and 6) some rules that seem to be particularly promising.

A first useful observation is that a rule that minimizes \( P_2 \) usually gives a reasonably small value of \( P_6 \) and vice versa—a fact that is easily understood from the error expression (20). (For a more complete discussion, see [6].) In the following we therefore concentrate on the results for \( P_2 \).
The full results also record, in each case, separate 'best' results for 'Korobov' and 'random' vectors \( z \) or \( p \). (See subsection 3.1.) In the following, however, we shall generally quote just the minimum of the two values so obtained.

4.1. The 1000-point window. Figures 1 to 3 show the minimum values of \( P_2 \) obtained, for dimensions 3, 5 and 8, for all values of \( N \) in the 1000-point window. The three separate graphs in each case are for the rank-1 case (solid line), the rank-2 \( n = 3 \) case (short dashed line) and the rank-2 \( n = 2 \) case (long dashed line). (The lines are included merely to guide the eye between the plotted points, and have no other significance.)

For the three-dimensional results in Figure 1 one sees the rank-2 \( n = 3 \) rules performing, on the whole, as well as the rank-1 results, and the rank-2 \( n = 2 \) rules performing rather better than either of the others.

Similar conclusions hold in all dimensions; but the apparent superiority of the rank-2 \( n = 2 \) rules becomes much more marked as the dimensionality
Integration error $P_2$ (as defined in (15)) as a function of the number of integrand evaluations ($N$ or $n^2r$) in 5 dimensions, for the 1000-point window. Each point gives the smallest $P_2$ value obtained from searches at fixed $N$.

increases: for, in the 5-dimensional and 8-dimensional results of Figures 2 and 3, the graph of the best rank-2 $n = 2$ rules is very well separated from both of the others. In no case, in any of the figures, have we found a rank-1 rule that competes with the best of the rank-2 $n = 2$ rules.

Of the rules in Figures 1 to 3, some have $z$ or $p$ vectors of ‘Korobov’ type, and some have ‘random’ $z$ or $p$ vectors. The honors are about even, with the proportion of ‘Korobov’ vectors rising slowly (to about 60%) as the dimension $s$ increases.

A striking (and to us totally unexpected) observation from the complete tabulation of the ‘best’ rules in the 1000-point window is that most of them (for example, 65% of them when $s = 3$, and 86% when $s = 6$) have the particular $y_1, y_2$ pair given by (25). The bias is particularly striking for the larger values of $s$, since the total number $N_y$ of possible $y_1, y_2$ pairs rises rapidly with $s$ (see Table 2).
A bias towards the special $y_1, y_2$ pair given by (25) also persists, though in less marked form, in the rank-2 $n = 3$ tabulated results for the 1000-point window. In this case, about one third of the 'best' rules have the special $y_1, y_2$ pair for all dimensions from 3 to 8, compared to a much smaller expected proportion (see Table 2) if the distribution were statistical.

In summary, the salient features of the results for the 1000-point window appear to be that the 'best' of the rank-2 $n = 2$ rules consistently perform better than both the rank-1 and rank-2 $n = 3$ rules; and that these best rules have a very strong bias towards the particular $y_1, y_2$ pair given by (25).

In Table 4 we have selected, rather subjectively, a subset of rules from the complete tabulation of rules for the 1000-point window, that seem to us particularly good as judged by the performance for both $P_2$ and $P_6$. All, of course, are rank-2 $n = 2$ rules, and (with just one exception) all have the special vector pair $y_1, y_2$ given by (25). Many of these rules minimize both $P_2$ and $P_6$. If
Table 4
A selection of rank-2 rules from the computer searches, in the 1000-point window, for 3 to 8 dimensions. All have \( n = 2 \) and, except for the one case shown otherwise, the special vector pairs \( \mathbf{y}_{1} = \mathbf{y}_{1} \), \( \mathbf{y}_{2} = \mathbf{y}_{2} \) given by (25).

<table>
<thead>
<tr>
<th>( s )</th>
<th>( y_{1} )</th>
<th>( y_{2} )</th>
<th>( z )</th>
<th>( r )</th>
<th>( N )</th>
<th>No. of rules searched</th>
<th>( P_{2} )</th>
<th>( P_{6} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>85 173 215</td>
<td></td>
<td></td>
<td>253</td>
<td>1012</td>
<td>504</td>
<td>0.48779E-02</td>
<td>0.35318E-10</td>
</tr>
<tr>
<td>3</td>
<td>24 69 235</td>
<td></td>
<td></td>
<td>257</td>
<td>1028</td>
<td>512</td>
<td>0.49103E-02</td>
<td>0.55095E-10</td>
</tr>
<tr>
<td>3</td>
<td>1 118 197</td>
<td></td>
<td></td>
<td>259</td>
<td>1036</td>
<td>516</td>
<td>0.49369E-02</td>
<td>0.88647E-10</td>
</tr>
<tr>
<td>3</td>
<td>1 125 108</td>
<td></td>
<td></td>
<td>263</td>
<td>1052</td>
<td>524</td>
<td>0.46389E-02</td>
<td>0.62585E-10</td>
</tr>
<tr>
<td>3</td>
<td>1 77 143</td>
<td></td>
<td></td>
<td>263</td>
<td>1052</td>
<td>524</td>
<td>0.47776E-02</td>
<td>0.57511E-10</td>
</tr>
<tr>
<td>4</td>
<td>59 13 151 219</td>
<td></td>
<td></td>
<td>251</td>
<td>1004</td>
<td>750</td>
<td>0.68468E-01</td>
<td>0.56933E-07</td>
</tr>
<tr>
<td>4</td>
<td>134 181 44 175</td>
<td></td>
<td></td>
<td>255</td>
<td>1020</td>
<td>762</td>
<td>0.68871E-01</td>
<td>0.52421E-07</td>
</tr>
<tr>
<td>4</td>
<td>1000 0011</td>
<td>1 50 169 162</td>
<td></td>
<td>259</td>
<td>1036</td>
<td>774</td>
<td>0.68817E-01</td>
<td>0.28850E-07</td>
</tr>
<tr>
<td>4</td>
<td>1 50 133 75</td>
<td></td>
<td></td>
<td>263</td>
<td>1052</td>
<td>786</td>
<td>0.64661E-01</td>
<td>0.50868E-07</td>
</tr>
<tr>
<td>5</td>
<td>165 155 50 116 31</td>
<td></td>
<td></td>
<td>243</td>
<td>972</td>
<td>968</td>
<td>0.57793E+00</td>
<td>0.2054E-04</td>
</tr>
<tr>
<td>5</td>
<td>125 29 142 76 12</td>
<td></td>
<td></td>
<td>243</td>
<td>972</td>
<td>968</td>
<td>0.59426E+00</td>
<td>0.16535E-04</td>
</tr>
<tr>
<td>5</td>
<td>34 210 142 39 155</td>
<td></td>
<td></td>
<td>261</td>
<td>1044</td>
<td>1040</td>
<td>0.55351E+00</td>
<td>0.17458E-04</td>
</tr>
<tr>
<td>5</td>
<td>1 37 84 24 117</td>
<td></td>
<td></td>
<td>257</td>
<td>1028</td>
<td>1024</td>
<td>0.53875E+00</td>
<td>0.12919E-04</td>
</tr>
<tr>
<td>5</td>
<td>1 90 210 227 179</td>
<td></td>
<td></td>
<td>263</td>
<td>1052</td>
<td>1048</td>
<td>0.51525E+00</td>
<td>0.10321E-04</td>
</tr>
<tr>
<td>5</td>
<td>1 103 89 225 31</td>
<td></td>
<td></td>
<td>263</td>
<td>1052</td>
<td>1048</td>
<td>0.53621E+00</td>
<td>0.96425E-05</td>
</tr>
<tr>
<td>6</td>
<td>231 85 107 143 5147</td>
<td></td>
<td></td>
<td>243</td>
<td>972</td>
<td>1210</td>
<td>0.35331E+01</td>
<td>0.12668E-02</td>
</tr>
<tr>
<td>6</td>
<td>61 16 65 76 2326</td>
<td></td>
<td></td>
<td>251</td>
<td>1004</td>
<td>1250</td>
<td>0.34273E+01</td>
<td>0.17762E-02</td>
</tr>
<tr>
<td>6</td>
<td>65 46 116 239 179 57</td>
<td></td>
<td></td>
<td>251</td>
<td>1004</td>
<td>1250</td>
<td>0.34313E+01</td>
<td>0.89440E-03</td>
</tr>
<tr>
<td>6</td>
<td>250 92 116 155 5159</td>
<td></td>
<td></td>
<td>263</td>
<td>1052</td>
<td>1310</td>
<td>0.32123E+01</td>
<td>0.77498E-03</td>
</tr>
<tr>
<td>6</td>
<td>1 98 136 178 86 12</td>
<td></td>
<td></td>
<td>263</td>
<td>1052</td>
<td>1310</td>
<td>0.31434E+01</td>
<td>0.73189E-03</td>
</tr>
<tr>
<td>7</td>
<td>99 38 138 228 106 201 99</td>
<td></td>
<td></td>
<td>243</td>
<td>972</td>
<td>1452</td>
<td>0.16982E+02</td>
<td>0.41247E-01</td>
</tr>
<tr>
<td>7</td>
<td>6 24 173 229 97 206 77</td>
<td></td>
<td></td>
<td>243</td>
<td>972</td>
<td>1452</td>
<td>0.17953E+02</td>
<td>0.15417E-01</td>
</tr>
<tr>
<td>7</td>
<td>120 178 189 203 141 199 25</td>
<td></td>
<td></td>
<td>257</td>
<td>1028</td>
<td>1536</td>
<td>0.16441E+02</td>
<td>0.13001E+00</td>
</tr>
<tr>
<td>7</td>
<td>147 6 120 170 173 35 179</td>
<td></td>
<td></td>
<td>257</td>
<td>1028</td>
<td>1536</td>
<td>0.16750E+02</td>
<td>0.10966E-01</td>
</tr>
<tr>
<td>7</td>
<td>1 32 44 183 221 212 169</td>
<td></td>
<td></td>
<td>245</td>
<td>980</td>
<td>1464</td>
<td>0.17332E+02</td>
<td>0.90423E-02</td>
</tr>
<tr>
<td>7</td>
<td>1 82 42 103 222 214 72</td>
<td></td>
<td></td>
<td>257</td>
<td>1028</td>
<td>1536</td>
<td>0.16245E+02</td>
<td>0.90464E-02</td>
</tr>
<tr>
<td>7</td>
<td>1 77 18 101 67 19 178</td>
<td></td>
<td></td>
<td>257</td>
<td>1028</td>
<td>1536</td>
<td>0.16486E+02</td>
<td>0.80419E-02</td>
</tr>
<tr>
<td>8</td>
<td>198 132 75 93 201 181 159 174</td>
<td></td>
<td></td>
<td>255</td>
<td>1020</td>
<td>1778</td>
<td>0.75847E+02</td>
<td>0.45566E+00</td>
</tr>
<tr>
<td>8</td>
<td>92 3 144 60 185 141 109 37</td>
<td></td>
<td></td>
<td>255</td>
<td>1020</td>
<td>1778</td>
<td>0.76042E+02</td>
<td>0.22930E+00</td>
</tr>
<tr>
<td>8</td>
<td>90 39 112 244 12 78 118 156</td>
<td></td>
<td></td>
<td>261</td>
<td>1044</td>
<td>1820</td>
<td>0.72828E+02</td>
<td>0.25408E+00</td>
</tr>
<tr>
<td>8</td>
<td>1 62 49 2 124 98 4 248</td>
<td></td>
<td></td>
<td>253</td>
<td>1012</td>
<td>1764</td>
<td>0.73147E+02</td>
<td>0.24833E+00</td>
</tr>
<tr>
<td>8</td>
<td>1 60 181 77 149 261 143 164</td>
<td></td>
<td></td>
<td>263</td>
<td>1052</td>
<td>1834</td>
<td>0.74247E+02</td>
<td>0.26772E+00</td>
</tr>
<tr>
<td>8</td>
<td>1 68 153 147 2 136 43 31</td>
<td></td>
<td></td>
<td>263</td>
<td>1052</td>
<td>1834</td>
<td>0.74308E+02</td>
<td>0.23635E+00</td>
</tr>
</tbody>
</table>

this is not the case, then we show two different rules for the same values of \( s \) and \( N \), one minimizing \( P_{2} \), and one minimizing \( P_{6} \). (The quantity not minimized by the particular rule is enclosed in parentheses.)

The rules in Table 4 in which the \( z \) vectors are of the 'Korobov' form are distinguished by the fact that \( z^{(1)} = 1 \); in all other cases, the \( z \)-vectors are 'random'.

Finally, Table 4 allows us to make a useful point about the use of \( P_{2} \) as a search criterion. If we recall that \( P_{2} \) is the error in the integration of a function (namely \( f_{2} \)) whose exact integral is 1, it may seem bizarre in the extreme to persist with \( P_{2} \) when its value is comparable to or greater than 1. Yet, we see in Table 4 that rules that minimize \( P_{2} \) often also minimize \( P_{6} \), which is an integration error of much more sensible size. Thus, the use of \( P_{2} \) as a criterion may still be defensible even when its value is large.
4.2. The 10 000-point window. We recall from subsection 3.2 that for the 10 000-point window the first step was to carry out both a 'full' search and a 'reduced' search (i.e., one restricted to the special pair \(y_1 = Y_1, y_2 = Y_2\)) for the rank-2 \(n = 2\) case, the purpose being to assess the effectiveness of the reduced search strategy.

The full search required in total 2940 cpu seconds on a Cyber 205, compared with 185 cpu seconds for the reduced search. Yet, in 9 cases out of 12 (counting separately the \(P_2\) and \(P_5\) minimizations, and dimensions from 3 to 8), the two searches yielded exactly the same result. In the other three cases, the full search yielded marginally better results, but hardly enough to justify fifteen times the expense. We therefore resolved to concentrate exclusively on reduced searches.

Adopting the reduced search, rules for a range of \(N\)-values between 9972 and 10 764 were searched as described in §3, and the results compared graphically in a manner similar to the 1000-point window. Again, a comparison of rank-1 and rank-2 rules was made, and it was found that the rank-2 \(n = 2\) rules consistently performed better than the others in four to eight dimensions, and the performance of the rank-2 \(n = 3\) and rank-1 rules were about equal. This can be seen for the cases of three, five and eight dimensions in Figures 4, 5 and 6. The case of three dimensions, in Figure 4, has all three types of rules giving roughly equal performance. The superiority of the rank-2 \(n = 2\) rules thus

| Table 5 |
| A selection of rank-2 rules from the computer searches, in the 10 000-point window, for 3 to 8 dimensions. All have \(n = 2\) and the special vector pairs \(y_1 = Y_1, y_2 = Y_2\) given by (25). |

<table>
<thead>
<tr>
<th>(s)</th>
<th>(z)</th>
<th>(r)</th>
<th>(N)</th>
<th>No. of rules searched</th>
<th>(P_2)</th>
<th>(P_5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1 1186 2056</td>
<td>2601</td>
<td>10404</td>
<td>1300</td>
<td>0.92970E-04</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1026 1597 2113</td>
<td>2637</td>
<td>10548</td>
<td>1318</td>
<td>0.96583E-04</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1 425 2398 1448</td>
<td>2583</td>
<td>10332</td>
<td>1291</td>
<td>0.21414E-02</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1 337 178 1972</td>
<td>2637</td>
<td>10548</td>
<td>1318</td>
<td>0.21490E-02</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1 473 376 242</td>
<td>2691</td>
<td>10764</td>
<td>1345</td>
<td>0.21399E-02</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1 638 703 1808</td>
<td>2691</td>
<td>10764</td>
<td>1345</td>
<td>0.21573E-02</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1 721 253 1576 334</td>
<td>2547</td>
<td>10188</td>
<td>1273</td>
<td>0.26107E-01</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1 961 571 235 1690</td>
<td>2637</td>
<td>10548</td>
<td>1318</td>
<td>0.24746E-01</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1 988 1759 1522 1006</td>
<td>2655</td>
<td>10620</td>
<td>1327</td>
<td>0.24233E-01</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1 1159 1012 1288 250 1939</td>
<td>2547</td>
<td>10188</td>
<td>1273</td>
<td>0.20869E+00</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1 554 968 1704 1618 989 1163</td>
<td>2583</td>
<td>10332</td>
<td>1291</td>
<td>0.20161E+00</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1 1747 213 901 552 1690 699</td>
<td>2583</td>
<td>10332</td>
<td>1291</td>
<td>0.22026E+00</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1 422 199 1673 2431 1052</td>
<td>2655</td>
<td>10620</td>
<td>1327</td>
<td>0.19951E+00</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1 253 289 1432 2126 2323</td>
<td>2655</td>
<td>10620</td>
<td>1327</td>
<td>0.20859E+00</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1 758 1138 2465 961 32 1009</td>
<td>2583</td>
<td>10332</td>
<td>1291</td>
<td>0.12749E+01</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1 514 730 685 802 1531 1702</td>
<td>2583</td>
<td>10332</td>
<td>1291</td>
<td>0.12752E+01</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1 1887 1679 713 121 257 252 2415</td>
<td>2601</td>
<td>10404</td>
<td>1300</td>
<td>0.12471E+01</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1 1211 2617 1880 94 812 1117</td>
<td>2691</td>
<td>10764</td>
<td>1345</td>
<td>0.11802E+01</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1 1151 361 350 424 1547 244 674</td>
<td>2547</td>
<td>10188</td>
<td>1273</td>
<td>0.67017E+01</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1 641 814 2186 376 1598 424 1802</td>
<td>2547</td>
<td>10188</td>
<td>1273</td>
<td>0.67083E+01</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1 895 2014 1459 490 808 622 2431</td>
<td>2637</td>
<td>10548</td>
<td>1318</td>
<td>0.62780E+01</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1 620 2035 1214 1135 2258 2350 1376</td>
<td>2637</td>
<td>10548</td>
<td>1318</td>
<td>0.63798E+01</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1 83 1507 1295 2536 590 532 1100</td>
<td>2691</td>
<td>10764</td>
<td>1345</td>
<td>0.63424E+01</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1 1199 607 1223 2473 2336 2224 2486</td>
<td>2691</td>
<td>10764</td>
<td>1345</td>
<td>0.64452E+01</td>
<td></td>
</tr>
</tbody>
</table>
A COMPUTER SEARCH OF RANK-2 LATTICE RULES

becomes much more marked as the dimensionality increases; and only in three
dimensions, at a high level of convergence, have we found a rank-1 rule that
competes with the best rank-2 rules. Moreover, if the best rank-1 rules have \( p \)

vectors of Korobov type, then they have been found by searches of about four
times as many rules as the best rank-2 \( n = 2 \) rules, and about nine times as
many as the best rank-2 \( n = 3 \) rules. Similar results also hold when the criterion
is \( P_6 \), except that the \( P_6 \) errors obtained with the best rank-2 \( n = 3 \) rules are
consistently greater than in the rank-1 case. Table 5 contains a selection of some
of our best rank-2 \( n = 2 \) rules which appear useful for practical applications.

Table 6

A selection of rank-2 rules from the computer searches, in the 100 000-point
window, for 3 to 8 dimensions. All have \( n = 2 \) and the special vector pairs
\( y_1 = Y_1, y_2 = Y_2 \) given by (25).

<table>
<thead>
<tr>
<th>( s )</th>
<th>( r )</th>
<th>( N )</th>
<th>No. of rules searched</th>
<th>( P_2 )</th>
<th>( P_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1534 10173 9373</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.22978E-05</td>
</tr>
<tr>
<td>3</td>
<td>7019 2348 9701</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.21889E-05</td>
</tr>
<tr>
<td>1</td>
<td>2962 19954</td>
<td>2501</td>
<td>100 0044</td>
<td>125 &amp; 1250</td>
<td>0.23571E-05</td>
</tr>
<tr>
<td>4</td>
<td>8619 4353 6230 7317</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.81738E-04</td>
</tr>
<tr>
<td>4</td>
<td>7317 17161 15361 20189</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.70548E-04</td>
</tr>
<tr>
<td>4</td>
<td>5416 20164 10198</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.98855E-04</td>
</tr>
<tr>
<td>4</td>
<td>3112 5277 20917</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.63731E-04</td>
</tr>
<tr>
<td>5</td>
<td>5477 3083 3651 6657 22019</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.11765E-02</td>
</tr>
<tr>
<td>5</td>
<td>21049 2618 14782 19415 18425</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.11338E-02</td>
</tr>
<tr>
<td>5</td>
<td>1 5416 20164 10198</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.98855E-04</td>
</tr>
<tr>
<td>5</td>
<td>1 3112 5277 20917</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.63731E-04</td>
</tr>
<tr>
<td>6</td>
<td>3169 13979 18315 2471 8357 20188</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.11444E-01</td>
</tr>
<tr>
<td>6</td>
<td>1 3476 2263 12734 18925 4370</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.10695E-01</td>
</tr>
<tr>
<td>7</td>
<td>21555 6000 20758 18301 23971 19852 12294</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.92860E-01</td>
</tr>
<tr>
<td>7</td>
<td>9552 24926 22900 19119 21046 1864 23884</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.93730E-01</td>
</tr>
<tr>
<td>7</td>
<td>1 12245 24091 14561 21037 976 4474</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.97294E-01</td>
</tr>
<tr>
<td>7</td>
<td>1 9734 9088 21696 5422 4538 3466</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.99940E-01</td>
</tr>
<tr>
<td>8</td>
<td>14747 13874 19956 21322 20288 5245 19183 10465</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.56734E+00</td>
</tr>
<tr>
<td>8</td>
<td>3212 5555 17120 4251 18381 3147 4888 9668</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.57531E+00</td>
</tr>
<tr>
<td>8</td>
<td>1 1342 1747 2575 4573 9271 11215 18919</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.54047E+00</td>
</tr>
<tr>
<td>8</td>
<td>1 5066 3070 20789 20764 19169 17452 22958</td>
<td>2501</td>
<td>100 0044</td>
<td>125</td>
<td>0.54125E+00</td>
</tr>
</tbody>
</table>

4.3. The 100 000-point window. The full details of these searches are given at
the end of §3, and all the rules found in this search are listed in Table 6. This
table contains two rules in each dimension for \( N = 100 044 \) when the number of
rules searched is 125; and six rules in 3 to 5 dimensions, with the same \( N \),
but with the number of rules searched being 1250. The larger search could not
be continued in 6, 7 and 8 dimensions due to insufficient computer resources.
Table 6 also contains two rules in each of 7 and 8 dimensions for \( N = 131 004 \).
(The \( N = 100 044 \) point rules were considered to give adequate convergence,
in up to 6 dimensions, for most practical applications, so the further effort of
searching \( N = 131 004 \) was restricted to 7 and 8 dimensions.) The computer
Integration error $P_2^2$ (as defined in (15)) as a function of the number of integrand evaluations ($N$ or $n^2r$) in 3 dimensions, for the 10,000-point window. Each point gives the smallest $P_2^2$ value obtained from searches at fixed $N$.

The cpu time taken for all the $N = 100,044$ results with 125 searches in Table 6 was 1031 seconds; the $N = 131,004$ results for $s = 7$ and 8 alone took 979 seconds and required the full memory resources of the Cyber 205 for their attainment.

The quality of the rules in Table 6 may be assessed by comparison with rank-1 rules previously published by Maisonneuve [7] and Haber [2]. The quantity $P_2^2$ in this work is the same as $|R_NP_2^2|$ of Haber (where $s = $ dimension) and $P_{(2)}^2(g)$ of Maisonneuve; this was checked computationally by using some of Haber's and Maisonneuve's rules in the programs used to compute $P_2$. A comparison between the rank-1 $N = 100,063$ rules in Maisonneuve and our rank-2 $n = 2$, $N = 100,044$ results from searching 125 rules shows that the latter gives $P_2^2$ values which are between 47% and 74% of Maisonneuve's $P_{(2)}^2(g)$ values. A comparison between the rank-1 $N = 98,304$ rules in Haber and our rank-2 $n = 2$, $N = 100,044$, with 125 searched, rules shows that the latter give $P_2^2$ values.
5. Discussion

Two empirical conclusions emerge clearly from the computer searches reported in this paper. The first is that, by the standard criterion and for dimensions greater than 3, the 'best' of our rank-2 \( n = 2 \) rules (i.e., lattice rules with invariants \( 2r, 2 \) and \( r \) odd) consistently outperform the best rank-1 rules obtained with a comparable search effort. This is true whether we look at rules with around 1000, or 10000 or 100000 abscissas.

The second conclusion is that for the best of the rank-2 \( n = 2 \) rules there is a marked tendency for the vectors \( y_1, y_2 \) in (8) to have the special values \( y_1 = Y_1, y_2 = Y_2 \) given by (25).
Integration error $P_2$ (as defined in (15)) as a function of the number of integrand evaluations ($N$ or $n^2 r$) in 8 dimensions, for the 10,000-point window. Each point gives the smallest $P_2$ value obtained from searches at fixed $N$.

At the present time, no convincing explanation has emerged for either phenomenon.

APPENDIX. Programming considerations

Here we consider only the case of rank-2 rules. The situation for rank-1 rules is similar, and even easier.

Using (15) and (8), the object is to compute, for $\alpha$ a positive even integer,

$$P_\alpha = \frac{1}{n^2 r} \sum_{j=1}^{r} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \bar{f}_\alpha \left( j \frac{z}{r} + k_1 \frac{y_1}{n} + k_2 \frac{y_2}{n} \right) - 1,$$

with $f_\alpha(x) = \prod \phi_\alpha(x^{(i)})$, and $\phi_\alpha$ given by (17).

The function $\phi_\alpha$ has the symmetry property

$$\phi_\alpha(x) = \phi_\alpha(1-x),$$
leading to
\[ f_n(x) = f_n((1, 1, \ldots, 1) - x). \]

Since the lattice rule (8), like every lattice rule, is symmetric about the center of the cube, this symmetry may be used to reduce the number of terms needed in the sum: it can easily be seen that (A1) can be replaced by
\[ P_n = \frac{2}{n^r} \sum_{j=0}^{\lfloor r/2 \rfloor} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \frac{1}{n^r} \left( j \frac{z}{r} + k_1 \frac{y_1}{n} + k_2 \frac{y_2}{n} \right) - 1, \]
where the asterisk indicates that the term with \( j = 0 \) is to be halved, and if \( r \) is even, also that with \( j = r/2 \).

In our searches we evaluated \( P_n \) for many different rules, all having the same values of \( n \) and \( r \). Now the values of \( f_n \) needed for all rules with given \( n \) and \( r \) are products of a quite small set of \( \phi_n \) values, namely the set
\[ \{ \phi_n(l/nr) : l = 0, 1, \ldots, nr - 1 \}. \]
Thus the first step in a search, following an idea of [8], was to calculate an array of these \( nr \) values of \( \phi_n \). (The number of \( \phi_n \) values can be further reduced by a factor of close to two, by exploiting the symmetry (A2); but this is of no great consequence.)

For a given rule, our object was to construct a vector, of length \( n^2(\lfloor r/2 \rfloor + 1) \), of the values of \( f_n \) occurring in (A3), from which it is a trivial matter to deduce \( P_n \). Since
\[ f_n \left( j \frac{z}{r} + k_1 \frac{y_1}{n} + k_2 \frac{y_2}{n} \right) = \prod_{i=1}^{s} \phi_n \left( \left\{ j \frac{z^{(i)}}{r} + k_1 \frac{y_1^{(i)}}{n} + k_2 \frac{y_2^{(i)}}{n} \right\} \right), \]
\[ = \prod_{i=1}^{s} \phi_n \left( \left\{ nj^{(i)} + rk_1 y_1^{(i)} + rk_2 y_2^{(i)} \right\} \frac{1}{nr} \right), \]
an intermediate step was to create, for each component \( x^{(i)} \), a vector of length \( n^2(\lfloor r/2 \rfloor + 1) \), labelled by \( j, k_1, k_2 \), of the \( \phi_n \) values occurring in (A5), after which it is an efficient operation on the Cyber 205 to form the vector of \( f_n \) values by taking component-by-component products of the \( s \) individual vectors.

A final comment concerns the method of formation of the vectors of \( \phi_n \) values needed in (A5). For the \( i \)th component, the values needed are of the form \( \phi_n(l/nr) \), where
\[ l = nj^{(i)} + rk_1 y_1^{(i)} + rk_2 y_2^{(i)} \pmod{nr}. \]
for \( 0 \leq j \leq \lfloor r/2 \rfloor, 1 \leq k_1 \leq n, 1 \leq k_2 \leq n \). Thus for each component one first forms an integer vector of length \( n^2(\lfloor r/2 \rfloor + 1) \) by means of (A6), and then forms the vector of \( \phi_n \) values from the values already stored (see (A4) above). The process of forming a vector from a vector of indices of a stored array can
be efficiently handled on the Cyber 205 by means of a utility, the ‘vector data motion’ macro Q8VGATHR.

ACKNOWLEDGMENTS

We are grateful to Dr. J. N. Lyness for valuable discussions, to the Australian Research Council for generous financial support and to CSIRO for a merit award for use of the CSIRONET Cyber 205 computer.

BIBLIOGRAPHY

4. —, Number-theoretic methods of approximate analysis, Fizmatgiz, Moscow, 1963. (Russian)

School of Mathematics, University of New South Wales, Kensington, New South Wales 2033, Australia