

ON COMPUTING MAXIMAL LATTICE DIMENSIONS OF THE ICG

POUL E.J. PETERSEN

4 Jun 1998

ABSTRACT. In this paper we present an introduction to Inversive Congruential Generators and the Marsaglia Lattice Test. We then develop efficient algorithms for generating the entire sequence of an Inversive Congruential Generator and computing the Maximal Lattice dimension and apply these techniques to an exhaustive analysis of maximal lattice dimensions of all ICGs for all primes $5 \leq p < 100,000$. We also apply these optimizations to the Compound ICG.

INTRODUCTION

Since the advent of computers, people have wanted a deterministic algorithm for generating random numbers. The applications range from Monte Carlo simulations to cryptography. The interest in *deterministic* algorithms stems from two principal concerns. First, non-deterministic generators such as radio noise or Lava Lite[®] lamps [1] are impractical. Secondly, such chaotic generators do not provide reproducible data for numeric experiments; if a deterministic algorithm is used, one need only provide the initial conditions and the entire experiment is reproducible.

Of course the idea of “deterministic randomness” is a bit of an oxymoron. Instead, algorithms are chosen which provide sequences which behave like random numbers under various statistical tests [12; Chap. 5]; these then are so-called pseudorandom number generators. The classic example is the Linear Congruential Generator, introduced in 1949 by Lehmer [13], wherein a seed value x_0 , two parameters a and b and a prime p are chosen and the sequence is then generated by computing $x_{n+1} = ax_n + b \pmod{p}$. While this generator performs well under several tests [14], Marsaglia, in 1968 [2] introduced the Lattice Test and demonstrated that this generator has a “flaw”. That is, regardless of the parameters chosen, if we consider n -tuples of the sequence $\{x_1, x_2, \dots\}$ generated by an LCG plotted in n -space, then “the points are about as randomly spaced in the n -cube as

The author wishes to extend thanks to Mary Flahive, who provided invaluable suggestions through at least a half-dozen drafts. Also to OSU Mathematics Department at large for putting up with all those background processes. .

the atoms in a perfect crystal at absolute zero” [2; pg. 25]. This structure can have a disastrous effect on the outcome of a simulation, as demonstrated by Eichenauer and Lehn [3; pg. 323]

Note that this does not mean the LCG is not a “good” pseudorandom generator. In fact, every pseudorandom number generator must fail some statistical test simply because it is deterministic. Peter Hellekalek suggests the following interpretation: “Pseudorandom number generators are like antibiotics. ... Any type of generator has some unwanted side-effects” [10; pg. 1] This presents a problem because for example, the result of a Monte Carlo simulation should be dependent on the model not the nuances of the generator. In order to separate the behavior of the model from the generator, the idea is to run the simulation several times using generators with mutually exclusive defects.¹

In order to do this, we need examples of different generators and the properties that they possess. This leads us to the question of which generators avoid the lattice behavior of the LCG. In 1986, Eichenauer and Lehn [3; def. 2] introduced the Inversive Congruential Generator which is similar to the LCG except the recursion uses an inverse: $x_{n+1} = ax_n^{-1} + b$. In 1988, Niederreiter [6; Thm. 1] and Eichenauer et al. [11; thm. 2] provided lower bounds for the maximal lattice dimension of the ICG and thus proved that it has “good” lattice properties. In 1992, Flahive and Niederreiter [4; thm. 3] improved this lower bound again, but the theoretical lower bound is still lower than empirical tests suggest. In order to explore the lattice behavior of the ICG, it may be helpful to construct more empirical evidence. The goal of this paper is to test the maximal lattice dimension for all maximal period ICGs for all primes $5 \leq p < 100000$. In the process of generating these results, we will arrive at some efficient techniques for computing complete ICG sequences for small p , as well as efficient ways of finding the proper coefficients to guarantee full-length periods.

§1 BACKGROUND

1.1 Definition: ICG. *Let p be a prime integer and fix elements a and b in Z_p with $a \neq 0$. Choose a seed value x_0 and generate a sequence $\{x_n\}$ in Z_p by :*

$$x_{n+1} = \begin{cases} ax_n^{-1} + b & \text{if } x_n \neq 0 \\ b & \text{if } x_n = 0 \end{cases}$$

This generator is called an Inversive Congruential Generator, or ICG.

¹Alternatively, one could use a generator whose defects were known not to interfere with the particular simulation.

Note: In order to avoid the arbitrary seed value x_0 , and to ensure compatibility with other work, the initial value will always be $x_0 = b$. This is not a loss of generality since all of the generators considered here will be maximal period, except where noted (maximal period is defined shortly). With this convention the generator is completely determined by the integers a, b and p and we will use the notation $ICG(a, b; p)$ to represent a particular generator.

Notice also that in light of Euler's theorem, $x^{p-1} = 1 \pmod{p}$, we could use the alternative definition $x_{n+1} = ax_n^{p-2} + b$ to avoid the piecewise definition. Also, even when referring to a fragment of a sequence, we will use the notation $\{x_0, x_1, \dots, x_k\}$.

Example. Consider $ICG(2046865, 2342; 7531829)$:

$$x_0 = 2342$$

$$x_1 = 2046865(2342)^{-1} + 2342 = 0$$

$$x_2 = 2342$$

Note that the sequence generated by this ICG is $\{0, 2342\}$. This would not be a good generator for a pseudorandom application since the sequence is only length 2. However, given any ICG the finiteness of modular arithmetic guarantees that the sequence must repeat eventually. In fact, since inverses are bijective, the sequence will be purely periodic. The trick then is to pick coefficients a and b so that $ICG(a, b; p)$ will have a very long period. The best possible conditions would be a full-length period, that is $ICG(a, b; p)$ generates a sequence $\{x_0, x_1, \dots, x_{p-1}\} = Z_p$.

1.2 Definition: Maximal Period Length. *Let $\{x_0, x_1, \dots, x_n\}$ be a sequence of distinct integers in Z_p . We say the sequence has maximal length (or period length p) if and only if $\{x_0, x_1, \dots, x_n\} = Z_p$, that is every integer in Z_p occurs in the sequence.*

Thus, one way to test an ICG for maximal length is to compute the entire sequence and verify that each element of Z_p is in the sequence. This can easily be done by hand for $ICG(2, 2; 31)$ which proves the existence of maximal period ICGs. Of course, maximal period is a property we would like to guarantee before computing the sequence, especially for large p . As it turns out, there is a very nice relationship between maximal period ICGs and polynomials. First, some notation:

1.3 Definition: Multiplicative Order. *Let z be an element of a finite field F . The multiplicative order of z is the least positive integer k such that $z^k = 1$. We will use the notation $|z|$ to represent the multiplicative order of z .*

As shown in [9; sec3.2, thm 8], if we let t be the number of elements in F , then $z^{t-1} = 1 \pmod{p}$ for every $z \in F^*$. This shows that the multiplicative order is well defined. Also, in the same section of [9] we have if $|z| = k$ then k must divide $t - 1$.² Finally, notice that if $F = Z_p$, then this result is Euler's Theorem $x^{p-1} = 1 \pmod{p}$.

Example. Compute the order of 2θ in $F = \{a + b\theta \mid a, b \in Z_3, \theta^2 = 2\}$:

$$(2\theta)^2 = \theta^2 = 2$$

$$(2\theta)^3 = 2(2\theta) = \theta$$

$$(2\theta)^4 = \theta(2\theta) = 1$$

Thus, $|2\theta| = 4$, which as expected divides $3^2 - 1$.

This example also motivates the idea of field extensions. Notice that the polynomial $f(x) = x^2 + 1$ is irreducible over Z_3 since it has no roots, but is reducible over F , since $f(\theta) = 2 + 1 = 0$. This of course is expected since in the definition of F , $\theta^2 = 2$ which is equivalent to $\theta^2 - 2 = 0$. Thus F contains a root, namely θ , of $x^2 - 2$, which is equivalent to $f(x)$ since $-2 = 1$ in Z_3 . Consequently, by adjoining θ to Z_3 in the context of a vector space, we have created a new field F in which $f(x)$ factors.³

We can extend this idea to general p by considering a monic quadratic polynomial $f(x) = x^2 + bx + a \in Z_p[x]$ which is irreducible. We then let θ be a root of $f(x)$ and define $F_p(\theta) = \{a + b\theta \mid a, b \in Z_p, f(\theta) = 0\}$. Then, by definition $f(\theta) = 0$ and by observation, $f(-b - \theta) = 0$. Thus both roots of $f(x)$ are in $F_p(\theta)$, and $f(x)$ factors completely over $F_p(\theta)$.

Because of the dependence of $F_p(\theta)$ on the irreducible polynomial chosen, it would seem that each monic quadratic in $Z_p[x]$ would have a different field $F_p(\theta)$. However, the field $F_p(\theta)$ is itself unique, up to isomorphism. In other words every monic quadratic in $Z_p[x]$ factors in $F_p(\theta)$. For example, the irreducible polynomial $g(x) = x^2 + x + 2 \in Z_3[x]$ factors in F since $g(2\theta + 1) = 0$. This leads us to the definition of the unique field $GF(p^2)$.⁴

²The number of elements in the multiplicative group of F is $t - 1$ since the additive identity $\bar{0} \in F$ does not have a multiplicative inverse.

³We are avoiding the question of whether F is really a Field. This will always be the case provided that $f(x)$ is irreducible over Z_p [9; Ch. 7, Prop. 12, pg. 254].

⁴While we have limited this discussion to quadratics, the result is valid for degree n irreducible polynomials over Z_p , where the extension is a n -dimensional vector space over Z_p , called $GF(p^n)$. In fact, extensions need not be limited to finite fields. For example, by adjoining a root of $x^2 + 1$ to the Real numbers, we get the field of Complex numbers.

1.4 Definition. $GF(p^2)$ is the unique, up to isomorphism, finite field with p^2 elements.

Because of the uniqueness, we can represent $GF(p^2)$ by choosing any irreducible monic quadratic polynomial $f(x)$ over Z_p and if $f(\theta) = 0$ then

$$GF(p^2) \equiv \{a + b\theta \mid a, b \in Z_p, f(\theta) = 0\}$$

Notice that this is only a representation of $GF(p^2)$. The properties of $GF(p^2)$ can be entirely arrived at by starting with the definition of a finite field and the requirement that $GF(p^2)$ has p^2 elements.

Now we return to the task of establishing a link between maximal period ICGs and polynomials.

1.5 Definition: Related Polynomial. Given $ICG(a, b; p)$, we call the polynomial $f(x) = x^2 - bx - a \in Z_p[x]$ the related polynomial.

1.6 Definition: IMP. Let $f(x) = x^2 - bx - a$ in $Z_p[x]$ have roots α and β in $GF(p^2)$. We say that $f(x)$ is an Inversive Maximal Period polynomial, or IMP, if and only if the quotient $\frac{\alpha}{\beta}$ in $GF(p^2)$ has multiplicative order $p + 1$. In other words, $|\frac{\alpha}{\beta}| = p + 1$.

Note that if $f(x) = x^2 - bx - a$ is reducible over $Z_p[x]$ then since $f(x)$ is a quadratic, both roots must be in Z_p . Therefore $f(x) = (x - \alpha)(x - \beta)$ for $\alpha, \beta \in Z_p$. But then the quotient $\frac{\alpha}{\beta}$ is also an element of Z_p and $|\frac{\alpha}{\beta}|$ divides $p - 1$. Consequently, an IMP is necessarily irreducible over $Z_p[x]$. Also, for an IMP $b \neq 0$, otherwise $\frac{\alpha}{\beta} = -1 \in Z_p$.

1.7 Proposition. Let $f(x) = x^2 - bx - a$ in $Z_p[x]$ be an IMP, then $f(x)$ is irreducible and $b \neq 0$.

Finally, we have the following theorem due to Flahive and Niederreiter [4; Theorem 1] which completely describes maximal period ICGs:

1.8 Theorem. $ICG(a, b; p)$ is maximal period if and only if the related polynomial is an IMP.

Note: Because of the bi-conditional, any $ICG(a, b; p)$ of maximal period must have a related polynomial which is an IMP. Thus we will often refer to a maximal period ICG as an IMP. We close this section with a proof that $ICG(2, 2; 31)$ is maximal period.

Example. Consider $ICG(2, 2; 31)$ with related polynomial $f(x) = x^2 - 2x - 2$. The discriminant is 12 which is a quadratic non-residue by Euler's Criterion: $12^{15} = -1 \pmod{31}$. Thus, $f(x)$ is irreducible. Let $\beta \in GF(31^2)$ be a root of $f(x)$, so that $f(\beta) = \beta^2 - 2\beta - 2 = 0$.

We get the following two relations:

$$\beta^2 = 2\beta + 2, \quad \text{and} \quad \frac{1}{\beta} = 2^{-1}(\beta - 2)$$

Now, since $f(x) = (x - \alpha)(x - \beta) = x^2 - (\alpha + \beta)x + \alpha\beta$ we have $\alpha\beta = -2$ and so $\alpha = \frac{-2}{\beta}$. We now look at the quotient of the two roots and apply the above relations:

$$\frac{\alpha}{\beta} = \frac{-2}{\beta^2} = -2^{-1}(\beta - 2)^2 = -2^{-1}(\beta^2 - 4\beta + 4) = -2^{-1}(-2\beta + 6) = \beta - 3$$

By repeated reduction, we compute the following powers of $\frac{\alpha}{\beta}$:

$$\begin{aligned} \left(\frac{\alpha}{\beta}\right)^2 &= (\beta - 3)^2 = 11 - 4\beta \\ \left(\frac{\alpha}{\beta}\right)^4 &= (11 - 4\beta)^2 = 6\beta - 2 \\ \left(\frac{\alpha}{\beta}\right)^8 &= (6\beta - 2)^2 = 17\beta + 14 \\ \left(\frac{\alpha}{\beta}\right)^{16} &= (17\beta + 14)^2 = -1 \end{aligned}$$

The last line shows that $(\frac{\alpha}{\beta})^{32} = 1$ so that the order of $|\frac{\alpha}{\beta}|$ must divide 32. However, the proper divisors of 32 are 2, 4, 8, 16 and we've seen that none of these exponents of $\frac{\alpha}{\beta}$ yield 1. Thus, $|\frac{\alpha}{\beta}| = 32$ and $f(x)$ is an IMP.

§2 MARSAGLIA LATTICE TEST

Now that we have some background on maximal period ICGs, we shall introduce the Marsaglia Lattice Test. Consider a sequence $\{x_0, x_1, \dots, x_{N-1}\} \subseteq Z_p$ with period length $N \geq 2$. For a fixed dimension d , we define a set of vectors \mathbf{v}_i^d for $0 \leq i \leq N - 1$ in the vector space Z_p^d as follows:

$$\mathbf{v}_i^d = (x_i - x_0, x_{i+1} - x_1, \dots, x_{i+d-1} - x_{d-1}) \in Z_p^d$$

and let V^d be the subspace of Z_p^d spanned by the vectors $\{\mathbf{v}_0^d, \mathbf{v}_1^d, \dots, \mathbf{v}_{N-1}^d\}$. We now get the following definition of the MLT [15]:

2.1 Definition: Marsaglia Lattice Test or MLT. A sequence $\{x_n\}$ passes the lattice test for dimension d if $V^d = Z_p^d$; that is, d equals the dimension of V^d as a vector space over Z_p .

Notice that if for some D the set $\{\mathbf{v}_i^D\}$ spans Z_p^D then for non-negative $d < D$ the projections, \mathbf{v}_i^d , into Z_p^d will span Z_p^d . By the contrapositive, if $\{x_n\}$ fails the lattice test for a dimension D , then it also fails the lattice test for all $d > D$. This proves the existence of a maximal dimension:

2.2 Definition: Maximal Lattice Dimension. Given a sequence $\{x_n\}$, we define the maximal lattice dimension to be the unique positive integer D such that the sequence passes the MLT for all $d \leq D$ and fails the MLT for all $d > D$.

Notice that Marsaglia's Lattice Test can be applied to any modulus. Since we are interested in maximal period ICGs, we will consider only the case $N = p$. With this restriction, the assumption that $x_0 = b$ does not change the maximal lattice dimension, as proved in [4; Lemma 2]. We now obtain lower and upper bounds for the maximal lattice dimension D .

2.3 Theorem. Let $\{x_n\}$ be any sequence in Z_p of period $p \neq 2$. Then the maximal lattice dimension D satisfies $D \leq p - 2$.

Proof. First we observe that since $\mathbf{v}_0^p = 0$ the span of the p -vectors $\mathbf{v}_0^p, \dots, \mathbf{v}_{p-1}^p$ can not be Z_p^p . Thus the sequence fails the lattice test for $d = p$ implying $D < p$. For the dimension $p - 1$ lattice test we have the following:

$$\begin{aligned}
\sum_{i=0}^{p-1} \mathbf{v}_i^{p-1} &= \sum_{i=0}^{p-1} (x_i - x_0, x_{i+1} - x_1, \dots, x_{i+p-2} - x_{p-2}) \\
&= \left(\sum_{i=0}^{p-1} (x_i - x_0), \sum_{i=0}^{p-1} (x_{i+1} - x_1), \dots, \sum_{i=0}^{p-1} (x_{i+p-2} - x_{p-2}) \right) \\
&= \left(\sum_{i=0}^{p-1} (x_i) - px_0, \sum_{i=0}^{p-1} (x_{i+1}) - px_1, \dots, \sum_{i=0}^{p-1} (x_{i+p-2}) - px_{p-2} \right) \\
&= \left(\sum_{i=0}^{p-1} x_i, \sum_{i=0}^{p-1} x_{i+1}, \dots, \sum_{i=0}^{p-1} x_{i+p-2} \right)
\end{aligned}$$

Now, since the sequence x_n has period length p , we have that $\{x_0, x_1, \dots, x_{p-1}\} = Z_p$ so

that that for any integer $k \in Z_p$,

$$\sum_{i=0}^{p-1} x_{i+k} = \sum_{i=0}^{p-1} i = \frac{(p)(p+1)}{2} = 0$$

Thus, $\sum_{i=0}^{p-1} \mathbf{v}_i^{p-1} = (0, 0, \dots, 0)$. Consequently, the $\{\mathbf{v}_i^{p-1}\}$ vectors are linearly dependent and $V^{p-1} \neq Z_p^{p-1}$.

Notice that the critical step in the proof of (2.3) was the fact that the sum included every element of Z_p , so that we could apply the Gaussian identity and directly prove that the $\{\mathbf{v}_i^{p-1}\}$ vectors are linearly dependent. Fortunately, when computing the maximal lattice dimension $D \leq p - 2$ of IMP sequences, we can apply a result by Eichenauer and Niederreiter [5; pg. 246] to avoid working directly with the linear dependence of the vectors \mathbf{v}_i^d :

2.4 Theorem⁵. *An IMP sequence passes the MLT for all dimensions d such that:*

$$d \leq \max\{k \leq p - 2 \mid \sum_{n \in Z_p} n^{p-1-k} x_n \neq 0\}$$

Corollary. *An IMP has maximal lattice dimension $D = p - 2$ if and only if*

$$\sum_{n \in Z_p} n x_n \neq 0$$

As an example of the application of this result, we prove that $ICG(2, 2; 31)$ has a maximal lattice dimension of $p - 2$:

Example. Compute the maximal lattice dimension of $ICG(2, 2; 31)$.

From the last section we have seen that $ICG(2, 2; 31)$ is an IMP. A short computation reveals that if $\{x_n\}$ is the sequence generated by $ICG(2, 2; 31)$, then $\sum_{n=0}^{p-1} n x_n = 8 \neq 0$. Thus $ICG(2, 2; 31)$ has maximal lattice dimension $D = p - 2$.

This result combined with (2.3) proves that the best possible upper bound for the maximal lattice dimension of IMPs is $D \leq p - 2$. This however is not the lower bound for IMP maximal lattice dimensions; for example $ICG(28, 14; 31)$ has a maximal lattice dimension $D = p - 4$. What then can we say about a lower bound for the maximal lattice dimension of IMPs? Niederreiter [6: Theorem 1] has the following result:

⁵Eichenauer and Niederreiter prove the theorem for nonlinear maximal period generators, of which IMPs are a subclass

2.5 Theorem. *Let $\{x_n\}$ be an IMP sequence in Z_p . Then the maximal lattice dimension D is at least $(p + 1) / 2$.*

Recall from the Introduction that part of the reason that ICGs were introduced in [3; def. 2] was in response to the MLT, which showed a “defect” in the LCG. For completeness, we now prove this weakness of the LCG:

2.6 Theorem. *The Linear Congruential Generator $x_{n+1} = ax_n + b$ fails the lattice test for $d \geq 2$.*

Proof. We show that the LCG fails the lattice test for $d=2$. Consider the vectors $\mathbf{v}_i^2 = (x_i - x_0, x_{i+1} - x_1)$. Applying the relation $x_{n+1} = ax_n + b$ we have $x_{i+1} - x_1 = ax_i + b - ax_0 - b = a(x_i - x_0)$. Thus for any i , $\mathbf{v}_i^2 = (x_i - x_0)(1, a)$. Consequently, $V^2 = \text{span}\{(1, a)\} \neq Z_p^2$.

While the MLT identifies the weakness of the LCG, it is not, as with any other test, a panacea. With regards to testing pseudorandom number generators, the MLT is conclusive only when the lattice dimension is low; in other words, good pseudorandom numbers should have a high lattice dimension, but a high lattice dimension does not guarantee that a sequence is a good pseudorandom sequence. To prove this, all we need is an example of sequence with a high lattice dimension which would not perform well under other random tests. The following example found in [5; pg. 247] does precisely this:

Example. Consider the sequence $\{x_0, x_1, \dots, x_{p-1}\} = \{0, 1, \dots, p - 3, p - 1, p - 2\}$. Then, applying the previous lemma:

$$\sum_{n \in Z_p} nx_n = \sum_{n=1}^{p-3} n^2 + 2(p-1)(p-2) = p^3 - \frac{p^2}{2} + \frac{p}{6} - 1 = -1 \neq 0$$

Where we’ve applied the identity:

$$\sum_{n=1}^k n^2 = \frac{k(2k+1)(k+1)}{6} = \frac{k^3}{3} + \frac{k^2}{2} + \frac{k}{6}$$

Thus, the maximal lattice dimension of the sequence is $p - 2$.

Again, this example shows the limited applicability of the MLT since despite the good lattice structure, this sequence would certainly not be desirable for most pseudorandom applications. As Eichenauer and Niederreiter [5; pg 247] note: “This result shows the weakness of Marsaglia’s lattice test and indicates that this test should only be applied in addition to other criteria for selecting good pseudorandom number generators.”

Still, IMP sequences do perform well under standard pseudorandom tests and the lower bound for the maximal lattice dimension $D \geq (p + 1)/2$ is good. Interestingly, empirical data shows that the best possible lower bound might be much higher than $(p + 1)/2$. The question which arises then is “what is the lowest maximal dimension of an IMP?”.

§3 COMPUTING WITH MLT - OVERVIEW

Considering the theoretical lower-bound of $(p + 1)/2$ for ICG MLT dimension, two possibilities arise. First, this is the best possible theoretical lower bound for a general ICG; that is, at least one ICG exists with exactly this maximal dimension. Or the second possibility is that no such ICG exists and the theoretical lower bound might be improved. If there is a maximal dimension $(p + 1)/2$ ICG, we might find it with an exhaustive search. If there is no such ICG, then an exhaustive search may provide the empirical data for conjecturing a new lower bound. Thus we search for an example of a low maximal dimension ICG by proceeding through the primes, finding all ICGs and computing the maximal dimension for each ICG. This process raises three important questions:

- Q1. For a given field, how do we find all IMP?
- Q2. Since inverses are computationally demanding, how do we compute the ICG sequence efficiently?
- Q3. Once we have generated the sequence, how do we compute the maximal lattice dimension efficiently?

§4 QUESTION 1

Based on our exposition, one way to approach Q1 is to directly compute the order of the quotient α/β of the roots of $f(x) = x^2 - bx - a$ for each coefficient pair a and b . This is an extremely daunting task not only because there are $(p - 1)^2$ possibilities, but because $\alpha, \beta \in GF(p^2) \setminus Z_p$ whenever $f(x)$ is irreducible.

Fortunately, there exists groups of IMP polynomials which have the same maximal lattice dimension. This was first proven in [4; Theorem 4]:

4.1 Theorem. *Let $f(x) = x^2 - bx - a$ be an IMP polynomial. Let $f_1(x) = x^2 - b_1x - a_1$ with $\frac{b^2}{a} = \frac{b_1^2}{a_1}$ and $b^2 + 4a$ is a quadratic non-residue mod p . Then the sequences generated by f and f_1 have the same maximal lattice dimension.*

Now if we can identify the IMP families, then we need only compute the maximal lattice dimension of a single representative for each family. In order to achieve this, and to analyze the savings, we will find an equivalence between IMP polynomials and a class of order $p + 1$ polynomials [7]. However, we need to introduce some background material first:

4.2 Definition: Order of a Polynomial. Let $f(x)$ be a non-zero polynomial over Z_p , with $f(0) \neq 0$. Then the order of $f(x)$ is the smallest positive integer e such that $f(x)$ divides $x^e - 1$.

Note: If $f(x) \in Z_p[x]$ is degree n , then the splitting field of $f(x)$ is a subfield $GF(p^{n!})$. Thus, all of the roots of $f(x)$ are in $GF(p^{n!})$ which is the splitting field of $x^{p^{n!}} - x$. Since $f(0) \neq 0$, all of the roots of $f(x)$ are also roots of $x^{p^{n!-1}} - 1$. Now, choose k such that $p^k > n$, and consider $(x^{p^{n!-1}} - 1)^{p^k} = (x^{p^{n!}} - 1)^{p^{k-1}} = \dots = x^{p^{n!-1+k}} - 1$ where we have used the identity $(a + b)^p = a^p + b^p \pmod{p}$ [9; Prop. 35, pg. 460]. Since every element of $GF(p^{n!})$ is a root of $x^{p^{n!-1+k}} - 1$ with multiplicity $p^k > n$, then $f(x)$ must divide $x^{p^{n!-1+k}} - 1$. Consequently, the order of $f(x)$ is well defined.

4.3 Proposition. Let $f(x) \in Z_p[x]$ be a quadratic polynomial with $f(0) \neq 0$. If the order of $f(x)$ is $p + 1$, then $f(x)$ is irreducible.

Proof. Assume that $f(x)$ is reducible, then both roots of $f(x)$ are in Z_p since $f(x)$ is a quadratic. If the roots of $f(x)$ are distinct, then since every element of Z_p is a root of $x^p - x$ and $f(0) \neq 0$, $f(x)$ must divide $x^{p-1} - 1$. But then the order of $f(x)$ is less than p .

If $f(x)$ has a double root, say $a \in Z_p$, then we proceed by contradiction: Assume that $f(x)$ has order $p + 1$, then $f(x)$ divides $x^{p+1} - 1$. Thus, $a^{p+1} - 1 = a^2 - 1 = 0$ and $a = \pm 1$. Now, if $a = 1$ then $f(x) = (x - 1)^2$ must divide $x^{p+1} - 1$, but this is impossible since

$$\left. \frac{x^{p+1} - 1}{x - 1} \right|_{x=1} = \sum_{j=0}^p x^j \Big|_{x=1} = p + 1 = 1 \neq 0$$

Similarly, if $a = -1$, we have:

$$\left. \frac{x^{p+1} - 1}{x + 1} \right|_{x=-1} = - \sum_{j=0}^p (-x)^j \Big|_{x=-1} = -(p + 1) = -1 \neq 0$$

If a polynomial is irreducible, then there is a nice equivalence between the order of the polynomial and the order of the roots [8; Theorem 3.3]:

4.4 Theorem: Order of an Irreducible Polynomial. Let $f(x)$ be an irreducible polynomial over Z_p and let α be any root of $f(x)$. Then the order of $f(x)$ is equal to the multiplicative order of α in the splitting field of $f(x)$.

4.5 Definition: $m_f(x)$. Given $f(x) = x^2 - bx - a$ with $a \neq 0$, define $m_f(x) = x^2 - (\frac{-b^2}{a} - 2)x + 1$

We are now ready to redefine IMPs in terms of the order of $m_f(x)$, as found in [7; Theorem 2]:

4.6 Theorem. Let $f(x) \in Z_p[x]$ be a monic quadratic polynomial of the form $f(x) = x^2 - bx - a$. Then $f(x)$ is an IMP polynomial if and only if $m_f(x)$ has order $p + 1$.

Proof. First, assume that $f(x)$ is an IMP. Then if α and β are the roots of $f(x)$ we have

$$f(x) = x^2 - bx - a = (x - \alpha)(x - \beta) = x^2 - (\alpha + \beta)x + \alpha\beta$$

from which we see that $b = \alpha + \beta$ and $a = -\alpha\beta$. Now, since $f(x)$ is an IMP, then $|\alpha/\beta| = p + 1 > p - 1$ which implies that the element α/β is not in Z_p . At the same time, α/β must be in $GF(p^2)$ since both α and β are the roots of a quadratic in $Z_p[x]$. Applying Galois theory, the minimal polynomial of α/β must be:

$$m_{\alpha/\beta}(x) = (x - \alpha/\beta)(x - (\alpha/\beta)^p)$$

Now, since $|\alpha/\beta| = p + 1$ we have that

$$(\alpha/\beta)^p = (\alpha/\beta)^{-1} = \beta/\alpha$$

Combining these results yields:

$$m_{\alpha/\beta}(x) = (x - \alpha/\beta)(x - \beta/\alpha) = x^2 - (\alpha/\beta + \beta/\alpha)x + 1$$

Focusing on the middle coefficient, $\alpha/\beta + \beta/\alpha$ we see that:

$$\alpha/\beta + \beta/\alpha = \frac{\alpha^2 + \beta^2}{\alpha\beta} = \frac{(\alpha + \beta)^2 - 2\alpha\beta}{\alpha\beta} = \frac{-b^2}{a} - 2.$$

Notice that $a \neq 0$ (1.1) so this equation is well defined. Now $m_{\alpha/\beta}(x) = m_f(x)$, thus the two polynomials are the same order. Since $m_{\alpha/\beta}(x)$ is the minimal polynomial of α/β it is irreducible and thus has order $p + 1$ (4.4). Consequently $m_f(x)$ must have order $p + 1$ and the first direction is proved.

Now let $m_f(x) = x^2 - (\frac{-b^2}{a} - 2)x + 1$ have order $p + 1$. We wish to show that the polynomial $f(x) = x^2 - bx - a$ is an IMP. First, notice that $m_f(x)$ is irreducible (4.3). Thus the roots of $m_f(x)$ must also be order $p + 1$ (4.4). Now, let α and β be the roots of $f(x)$, then as before we get the equalities $b = \alpha + \beta$ and $a = -\alpha\beta$ as well as:

$$m_f(x) = x^2 - \left(\frac{\alpha}{\beta} + \frac{\beta}{\alpha}\right)x + 1$$

It is now trivial to show directly that $m_f\left(\frac{\alpha}{\beta}\right) = 0$ so that $\frac{\alpha}{\beta}$ is a root of $m_f(x)$ and has order $p + 1$. Thus, $f(x)$ is an IMP (1.6).

With this new definition of IMP polynomials, we can now define IMP families.

4.7 Definition: IMP families. *If $f(x)$ and $g(x)$ are both IMPs and if $m_f(x) = m_g(x)$ then we say that $f(x)$ and $g(x)$ belong to the same IMP family.*

Now we cast (4.1) in terms of $m_f(x)$ [7; Theorem 2]

4.8 Theorem. *Let $f_1(x) = x^2 - b_1x - a_1$ and $f_2(x) = x^2 - b_2x - a_2$ in $Z_p[x]$ such that $m_{f_1}(x) = m_{f_2}(x)$. $f_1(x)$ is an IMP if and only if $f_2(x)$ is an IMP. Furthermore, the ICGs sequences associated with $f_1(x)$ and $f_2(x)$ have the same maximal lattice dimension.*

Proof. The IMP implication is a trivial consequence of (4.6). Now, if $f_1(x)$ and $f_2(x)$ are both IMP polynomials then they are both irreducible (1.7) and $b_1^2 + 4a_1$ and $b_2^2 + 4a_2$ are quadratic non-residues. Finally, since $m_{f_1}(x) = m_{f_2}(x)$ we have $\frac{b_1^2}{a_1} = \frac{b_2^2}{a_2}$. Thus, the hypotheses of (4.1) are satisfied and the conclusion follows similarly.

As we discussed before, each member of an IMP family will have the same maximal lattice dimension, thus we need only test the maximal lattice dimension of a single representative of each family. To analyze how much work this saves in computation we consider the following found in Chou [7; Theorem 3]:

4.9 Theorem. *In the Field Z_p : 1) There are exactly $\frac{\phi(p+1)}{2}$ IMP families, where ϕ is the Euler totient function. 2) Each IMP family has exactly $p - 1$ IMP members.*

Proof. 1) Each family is governed by the polynomial $m_f(x)$, which has order $p + 1$. If $\tau \in GF(p^2)$ is a root of $m_f(x)$ then by Galois Theory the other root of $m_f(x)$ in $GF(p^2)$ must be τ^p , which is τ^{-1} since $|\tau| = p + 1$. Thus, $m_f(x) = (x - \tau)(x - \tau^{-1})$ and so for each pair τ and τ^{-1} there is a different $m_f(x)$. Consequently, there are exactly half as many $m_f(x)$ as there are τ of order $p + 1$ in $GF(p^2)$. Since there are exactly $\phi(p + 1)$ elements of order $p + 1$ in $GF(p^2)$, the result follows.

2) Let $m(x) = x^2 - cx + 1 \in Z_p[x]$ be a polynomial of order $p + 1$ and let $f(x) = x^2 - bx - a$ such that $m_f(x) = m(x)$. Thus, $\frac{-b^2}{a} - 2 = c$ (4.5) and since $m(x)$ has order $p + 1$ it is irreducible (4.3). Thus $c^2 - 4$ is a quadratic non-residue; in particular $c \neq -2$. Thus $\frac{-b^2}{a} - 2 = c$ is equivalent to $a = \frac{-b^2}{c+2}$. Finally, $f(x)$ is an IMP (4.6), thus $b \neq 0$

(1.7). Consequently for every $b \in Z_p^*$ the equation $a = \frac{-b^2}{c+2}$ has a non-zero solution for a . Therefore, the IMP family of $m(x)$ has exactly $p - 1$ members.

Corollary. *In Z_p there are exactly $\frac{\phi(p+1)(p-1)}{2}$ IMP polynomials.*

Consequently, by using IMP families we reduce the number of IMP dimension tests by a factor of $p - 1$. While this does reduce the number of overall computations, we still need a way to find all IMP families. Because of (4.6), this task is equivalent to finding all order $p + 1$ polynomials, which in turn is equivalent to finding all $\alpha_k \in GF(p^2)$ of order $p + 1$ (4.4). Consider $m(x) = x^2 - cx + 1 \in Z_p[x]$ of order $p + 1$. If α is a root of $m(x)$, then we have already seen that $m(x) = (x - \alpha)(x - \alpha^p)$ thus $c = \alpha + \alpha^p$. Since $m(x)$ is irreducible (4.3), we know that α has order $p + 1$ (4.4). Now, since $GF(p^2)$ is cyclic, all of the order $p + 1$ elements of $GF(p^2)$ are equal to α^k for some k with $\gcd(k, p + 1) = 1$. Thus, the order $p + 1$ polynomials can be generated by considering $m(x) = x^2 - c_k x + 1$ where $c_k = \alpha^k + \alpha^{pk}$ and $\gcd(k, p + 1) = 1$. This last expression for c_k is similar to the solution to the recursion $u_{n+2} = cu_{n+1} - u_n$ since the characteristic polynomial is $x^2 - cx + 1$. This is the essence of Chou's Algorithm [7; Theorem 7]:

4.10 Theorem: Chou's Algorithm.

Let $m(x) = x^2 - cx + 1 \in Z_p[x]$ have order $p + 1$. Let u_i be the sequence in Z_p generated by the recursion:

$$u_0 = 2, \quad u_1 = c, \quad u_{n+2} = cu_{n+1} - u_n, \quad n \geq 0$$

Then for $1 \leq k \leq (p - 1)/2$ the polynomials $g_k(x)$ given by $g_k(x) = x^2 - u_k x + 1$ have order $p + 1$ whenever $\gcd(k, p + 1) = 1$. Furthermore, this recursion will generate all monic quadratic polynomials of order $p + 1$.

Recall that Q1 was "How do we find the coefficients of all IMP?" We are now in a position to completely answer this question. First, we discovered that IMPs in the same family have the same maximal lattice dimension, thus we only need to find the coefficients of one IMP per family. So the question becomes, "how do we find all IMP families?" First we choose any IMP; we could use the GF package in Maple for example to find a polynomial $f(x) = x^2 - bx - a$ with roots α and β such that $|\alpha/\beta| = p + 1$. Then the polynomial $m_f(x) = x^2 - (\frac{-b^2}{a} - 2)x + 1$ has order $p + 1$. Thus, we can use the value $c = \frac{-b^2}{a} - 2$ as a seed for Chou's Algorithm. We then compute the sequence u_k and each time $\gcd(k, p + 1) = 1$ we know that the polynomial $g_k = x^2 - u_k x + 1$ has order $p + 1$. For each such $g_k(x)$ of order $p + 1$ choose $b_k \neq 0$ at random and solve $a_k = \frac{-b_k^2}{u_k + 2}$; then the related polynomial $f_k = x^2 - b_k x + a_k$ is an IMP. Since Chou's Algorithm will generate all

monic quadratic polynomials of order $p + 1$ and each IMP family corresponds to a unique monic quadratic polynomial of order $p + 1$, this will generate all IMP families. This process is summarized in the following example:

Example. Given the IMP $f(x) = x^2 - 2x - 5 \in Z_{17}[x]$ find a representative for each IMP family in Z_{17} . First we compute c :

$$c = \frac{-b^2}{a} - 2 = \frac{-2^2}{5} - 2 = -4(7) - 2 = 4$$

Now we use c to seed Chou's Algorithm, $u_0 = 2$ and $u_1 = c = 4$, then we use $u_{k+2} = u_1 u_{k+1} - u_k$ for $1 \leq k \leq (p - 1)/2$:

$$\begin{aligned} u_1 &= 4 & u_2 &= 4(4) - 2 = 14 & u_3 &= 4(14) - 4 = 1 \\ u_4 &= 4(1) - 14 = 7 & u_5 &= 4(7) - 1 = 10 & u_6 &= 4(10) - 7 = 16 \\ u_7 &= 4(16) - 10 = 3 & u_8 &= 4(3) - 16 = 13 \end{aligned}$$

Now $\gcd(k, 18) = 1$ only when $k \in \{1, 5, 7\}$. Thus u_1, u_5, u_7 correspond to order $p + 1$ polynomials. Notice that this gives 3 IMP families which agrees with $\phi(17 + 1)/2 = 3$ (4.9). Now we choose $b = 1$ for each u_k and solve for a_k :

$$\begin{aligned} a_1 &= \frac{-1}{u_1 + 2} = 14 \\ a_5 &= \frac{-1}{u_5 + 2} = 7 \\ a_7 &= \frac{-1}{u_7 + 2} = 10 \end{aligned}$$

Thus we have the three representatives:

$$ICG(14, 1; 17), \quad ICG(7, 1; 17), \quad ICG(10, 1; 17)$$

§5 QUESTION 2

Now that we can find all IMP families for a given p , we want to find an efficient way for computing the IMP sequences; the problem is that inverses are time consuming to compute. While good algorithms for computing inverses already exist, the interest in computing the full sequence of all IMP for a fixed prime introduces special optimization opportunities.

Consider computing the full sequence for all IMP for a fixed p . We would need no less than $p - 1$ inverses for each of the $\phi(p + 1)/2$ IMP families thus $(p - 1)\phi(p + 1)/2$ inverses. However, we can simplify this by noticing that for a fixed prime each of the IMP sequences needs the inverses of the same elements, namely Z_p , so in fact we need compute no more than $(p - 1)$ inverses. The drawback to this idea is storage; in order to implement this, we need to remember the inverses of all of Z_p . While this approach would not work for large primes, it is feasible for primes less than a few million.⁶

With the mind set of computing inverses and storing them, we consider the following approach: We employ an efficient inversion algorithm to compute the inverse of each element in Z_p and store the results in a $1 \times p$ matrix. This technique can be further optimized, which the following example will demonstrate:

Example. Compute the inverse of $231 \in Z_{4001}$.

The principle technique for computing inverses is to apply the Euclidean Algorithm to a Linear Diophantine equation as follows: solve the equation $a231 + b4001 = 1$ for integers $a, b \in Z_p$, then $231^{-1} = a$. First, we apply the Euclidean Algorithm to 4001 and 231:

$$4001 = 17(231) + 74$$

$$231 = 3(74) + 9$$

$$74 = 8(9) + 2$$

$$9 = 4(2) + 1$$

Now we begin with the last equation and substitute the previous equations for the intermediate remainders:⁷

$$\begin{aligned} 1 &= 9 - 4(2) = 9 - 4(74 - 8(9)) = 33(9) - 4(74) \\ &= 33(231 - 3(74)) - 4(74) = 33(231) - 103(74) \\ &= 33(231) - 103(4001 - 17(231)) = 1784(231) - 103(4001) \end{aligned}$$

Now that we have the solution $1784(231) - 103(4001) = 1$, we see that $231^{-1} = 1784$.

Notice that $1784^{-1} = 231$ and since $(-1)^{-1} = -1$ we also get two more inversions, namely $(-1784)^{-1} = -231$ and $(-231)^{-1} = -1784$. Thus the single inversion of 231 has provided the inverse of four elements in Z_p .

⁶It is not unusual for a modern computer to have more than 100 megabytes of RAM, which can store more than 25 million 32-bit integers, equivalent to testing $p < 25 \times 10^6$.

⁷This can be done more efficiently, but for clarity we present the general idea

Moreover, since we need the inverse of all the elements in Z_p , we can actually eliminate the Euclidean Algorithm altogether. Consider the first equation in the previous example $4001 = 17(231) + 74$ which can be reduced to $231^{-1} = (-17)74^{-1}$. That is, if 74^{-1} is known, then we can compute the inverse of 231 with only one division and one multiplication. This idea is summarized in the following:

5.1 Proposition. *Given $x \in Z_p \setminus \{0, 1\}$, $p > 2$, there exist $r, d \in Z_p$, $0 < r < x$ such that $x^{-1} = -dr^{-1}$.*

Proof. Since x can not divide p , this is a single application of the Quotient-Remainder Theorem to p and x : $p = dx + r$ for $0 < r < x$.

5.2 Definition: I_p -function. *Let $I_p : Z_p \mapsto Z_p$ be the function $I_p(x) = x^{-1}$ for every $x \in Z_p^*$ and $I_p(0) = 0$.*

We now apply the previous inversion ideas to computing I_p in as few steps as possible. We consider multiplications and divisions to be one operation, or op, each.

5.3 Proposition. *The I_p -function can be completely determined in $2[(p - j)/4]$ ops, where j is the number of initial conditions.*

Proof. We begin by considering the I_p -function as a $1 \times p$ matrix. Then we set the initial values $I_p(0) = 0$, $I_p(1) = 1$, and $I_p(p-1) = p-1$. Next we compute $I_p(2)$ by applying (5.1) to the equation $p = d(2) + r$. Since the only possible value for r is 1 and since $I_p(1)$ is known, the value of $I_p(2)$ can be computed in one division and one multiplication, or two ops. We then set the symmetric values $I_p(-2) = I_p(2)$, $I_p(I_p(2)) = 2$, and $I_p(-I_p(2)) = -2$. We continue in this fashion for each $x > 2$, skipping the computation if $I_p(x)$ is already known; otherwise we compute $I_p(x) = -d_x I_p(r_x)$ where $I_p(r_x)$ is already known since $r_x < x$. Since inverses are 1-1 each step produces 4 values which are unique from previous steps. This combined with j initial values implies that the algorithm will terminate in $2[(p - j)/4]$ ops.

Once the I_p function is initialized, any IMP sequence in Z_p can be computed using $x_{n+1} = aI_p(x_n) + b$. While this is better than re-computing the inverses for each of the $\phi(p+1)/2$ IMP families for fixed p , we are still computing a multiplication at each step of the IMP sequence, namely the $aI_p(x_n)$ term. Since we always multiply by a , this can actually be included in the inverse function by changing the initial conditions. This gives the $I_{a,p}$ -function:

5.4 Definition: $I_{a,p}$ -function. *Given $a \in Z_p^*$ we define the function $I_{a,p} : Z_p \mapsto Z_p$ by $I_{a,p}(x) = ax^{-1}$ for every $x \in Z_p^*$ and $I_{a,p}(0) = 0$. When the values a and p are clear we will refer to this function as the I -function or $I(x)$.*

As with the I_p -function, the $I_{a,p}$ -function satisfies the following identities:

$$I(-x) = -I(x), \quad I(I(x)) = x, \quad I(-I(x)) = -x, \quad I(xy) = I(x)y^{-1} = I(y)x^{-1}$$

From the last identity, we see that the $I_{a,p}$ -function satisfies Proposition 5.1. Also, notice that if $I_{a,p}(x) = I_{a,p}(y)$ then $ax^{-1} = ay^{-1}$ which is equivalent to $x = y$ since $a \neq 0$. Consequently, the I -function is 1-1 and Proposition 5.3 holds, with the initial values:

$$I(1) = a, \quad I(a) = 1, \quad I(-1) = -a, \quad I(-a) = -1, \quad I(0) = 0$$

As defined, the terms of an ICG can be computed with $I_{a,p}(x)$ by $x_{n+1} = I_{a,p}(x_n) + b$. While we have certainly eliminated the extra multiplications of using the I_p function, the savings from utilizing $I_p(x)$ was based on the principle that all of the IMP sequences for a fixed prime would use the same matrix. By letting the matrix $I_{a,p}$ depend on a we appear to have created an obstacle to this objective. However, recall that Chou's Algorithm generates a u_k term for each IMP family which is converted to an IMP family representative by choosing a_k and b_k so that $u_k = \frac{-b_k^2}{a_k} - 2$. If we can fix $a = a_k$ and solve $b_k^2 = -a(u_k + 2)$ then we could use the same $I_{a,p}$ to compute the sequence of each IMP. But, can we fix a so that the equation $b_k^2 = -a(u_k + 2)$ always has a solution, and how can we find the solution?

5.5 Lemma. *If $f(x) = x^2 - bx - a$ is an IMP, then $-a$ is a quadratic non-residue.*

Proof. As before, let α and β be the roots of $f(x)$ so that $-a = \alpha\beta$. Since $f(x)$ is irreducible over Z_p , the zeros are invariant under any Z_p -automorphism of $GF(p^2)$; $\beta = \alpha^p$. Now choose a primitive element $\sigma \in GF(p^2)$ so that $\alpha = \sigma^t$ for some $1 \leq t \leq p^2 - 1$. Then $\alpha/\beta = \alpha^{p-1} = \sigma^{t(p-1)}$. Since $f(x)$ is an IMP, $|\alpha/\beta| = p + 1$ and $\gcd(t, p + 1) = 1$. In particular, since $p + 1$ is even t must be odd.

Now assume that $-a$ is a quadratic residue, then by Euler's Theorem, $(-a)^{\frac{p-1}{2}} = 1$. This gives us the equality $\sigma^{\frac{t}{2}(p+1)(p-1)} = 1$. But since σ is a primitive element, this would imply that t is even. Therefore, $-a$ must be a quadratic non-residue.

5.6 Definition: r-parameter. *Given $f(x) = x^2 - bx - a$ an IMP, the r -parameter is $r = \frac{b^2}{a}$.*

Note: Consider two IMP polynomials $f_1(x) = x^2 - b_1x - a_1$ and $f_2(x) = x^2 - b_2x - a_2$ which belong to the same IMP family, thus $m_{f_1}(x) = m_{f_2}(x)$ (4.7). This is equivalent to the condition that $\frac{b_1^2}{a_1} = \frac{b_2^2}{a_2}$ (4.5). Thus, the r -parameter is invariant for an IMP family.

Also note that given the r -parameter of an IMP family, one can find all $p - 1$ members by solving $a = \frac{b^2}{r}$ for every $b \in Z_p^*$ (Recall, $b \neq 0$ (1.7)). Moreover, in the context of Chou's Algorithm, the r -parameters are generated by $r_k = -u_k - 2$.

5.7 Lemma. *Let r be the r -parameter for an IMP family. Then $-(r + 4)$ is a quadratic residue, and $-r$ is a quadratic non-residue.*

Proof. Let $f(x) = x^2 - bx - a$ be a member of the IMP family corresponding to r . Then $f(x)$ is irreducible (1.7) and $b^2 + 4a$ must be a quadratic non-residue. However, $b^2 + 4a = ra + 4a = a(r + 4) = (-a)(-(r + 4))$. Now $-a$ is a quadratic non-residue (5.5) and thus $-(r + 4)$ must be a quadratic residue. Now consider $m_f(x) = x^2 - (\frac{-b^2}{a} - 2)x + 1 = x^2 - (-r - 2)x + 1$ which is also irreducible (4.3). Thus, $(-r - 2)^2 - 4$ must be a quadratic non-residue, but this term factors: $(-r - 2)^2 - 4 = (r + 2 - 2)(r + 2 + 2) = r(r + 4) = -(-r)(r + 4)$. From the previous result, $-(r + 4)$ is a quadratic residue, and so $-r$ is a quadratic non-residue.

Now, for a fixed prime, all r -parameters satisfy the property that $-r_k$ is a non-quadratic residue. Thus, for a fixed element $-a$ which is a non-quadratic residue, the equation $b_k^2 = ar_k$ will always have a solution for b_k . That is, for fixed $a \neq 0$ there exists an IMP representative for each r -parameter, namely $ICG(a, t; p)$ where $t^2 = ar_k$. Consequently, we can use the same $I_{a,p}$ matrix for computing the sequence of each IMP.

However, while the computation of $I_{a,p}$ has been optimized, we now face a new problem which is how to solve $b_k^2 = ar_k$ for each of the r -parameters. Notice that for fixed p , we will need to compute a square root for each IMP family, thus we need $\phi(p + 1)/2$ or on average approximately $3p/16$ square roots.⁸ The need for many square roots provides an optimization, provided that p is relatively small. As with $I_{a,p}$, we solve square roots by storing the squares of Z_p in a $1 \times p$ matrix. It is important to note that this is only efficient in the context of this paper; that is when several square roots are required.

5.8 Definition: $B_{a,p}$ -function. *Given $a \in Z_p^*$ let $\Phi = \{0, 1, \dots, (p - 1)/2\}$ and let $\Psi = \{x^2 a^{-1} \mid \forall x \in \Phi\}$. Now let $\Omega : \Phi \mapsto \Psi$ be $\Omega(x) = x^2 a^{-1}$. Finally, define $B_{a,p} : \Psi \mapsto \Phi$ by $B_{a,p}(x) = \Omega^{-1}(x)$ When the values a and p are clear we will refer to this function as the B -function or $B(x)$.*

For this definition to make sense, we need to show that $\Omega(x)$ is 1-1. Let $x_1, x_2 \in \Phi$ such that $\Omega(x_1) = \Omega(x_2)$, then $x_1^2 a^{-1} = x_2^2 a^{-1}$. This is equivalent to $x_1^2 - x_2^2 = (x_1 + x_2)(x_1 - x_2) = 0$. Thus, $x_1 = x_2$ or $x_1 + x_2 = 0$. If $x_1 \neq x_2$ and $x_1 + x_2 = 0$ then $x_1 + x_2 = kp$ for some $k > 0$. But this is impossible since $x_1, x_2 \in \Phi$ implies that $x_1 + x_2 \leq p - 1$. Thus, $x_1 = x_2$, and $\Omega(x)$ is 1-1.

⁸This was computed numerically for $2 < p < 100000$.

This function can (like $I(x)$) be implemented as a $1 \times p$ matrix by setting $B_{a,p}(j^2 a^{-1}) = j$. for $1 \leq j \leq (p-1)/2$. In order to use this array to solve $b_k^2 = ar_k$ we set $b_k = B(r_k)$ since if $r_k = j^2 a^{-1}$ then $b_k^2 = (B(r_k))^2 = j^2 = ar_k$.

Example. Given the IMP $f(x) = x^2 - 2x - 5 \in Z_{17}[x]$ find a representative for each IMP family in Z_{17} with fixed $a = 5$.

First we set up the B -function which has the non-zero entries:

$$\begin{aligned} B[3] &= 7 & B[5] &= 5 & B[6] &= 8 & B[7] &= 1 \\ B[10] &= 4 & B[11] &= 2 & B[12] &= 3 & B[14] &= 6 \end{aligned}$$

Recall from the last section where we applied Chou's Algorithm to compute u_k , which we now present as r-parameters $r_k = -u_k - 2$:

$$r_1 = 11, \quad r_5 = 5, \quad r_7 = 12$$

In order to find a representative for the r_5 and the r_7 IMP families, we fix $a = 5$ and then solve $b_k = B[r_k]$: $b_5 = B[5] = 5$ and $b_7 = B[12] = 3$. Thus we have the three representatives

$$ICG(5, 2; 17), \quad ICG(5, 5; 17), \quad ICG(5, 3; 17)$$

Recall that Q2 is "Since inverses are computationally demanding, how do we compute the ICG sequence efficiently?" In this section we argued that since we need the entire IMP sequence, an efficient way to compute the sequence would be to efficiently store the inverses. We then showed that for a fixed prime, all of the IMP families could be represented by members with the same a parameter so that we could define an Inverse function which contains all of the necessary multiplications. This reduced the problem of computing the ICG sequences to simple addition, once the I -function is initialized with $2\lceil(p-3)/4\rceil$ operations. We then resolved the issue of computing square-roots with a similar technique. Now we need to consider efficient ways to compute the maximal lattice dimensions.

§6 QUESTION 3

Now that we have addressed both Q1 and Q2, it is time to consider Q3, which is given an ICG "How do we efficiently compute its maximal lattice dimension?" To do this, we will apply a few symmetries and properties of the ICG sequences. We begin with the a property of the maximal lattice dimension of ICGs due to [4; Theorem 3]:

6.1 Theorem. *The maximal lattice dimension of an IMP sequence, $\{x_n\}$, is an odd integer.*

Proof. We claim that for all x_n the following holds:

$$x_n + x_{p-1-n} = b \quad \text{for} \quad 0 \leq n < p \quad (*)$$

First notice that $x_0 = b$ and $x_{p-1} = 0$, thus the $n = 0$ case is true. Now suppose $x_k + x_{p-1-k} = b$ for some $0 \leq k \leq p-2$. Then since $x_k \neq 0$ and $x_{p-1-k} \neq 0$, $x_{p-1-k} = ax_{p-2-k}^{-1} + b$ from which $ax_{p-2-k}^{-1} = x_{p-1-k} - b = -x_k$ by the Inductive Hypothesis. Then $x_{p-2-k} = -ax_k^{-1} = b - x_{k+1}$ and consequently $x_{k+1} + x_{p-1-(k+1)} = b$ and (*) follows.

Now let $G(x) : Z_p \mapsto Z_p$ be the unique function such that $G(n) = x_n$. Notice that this function, as well as any other function on Z_p , can be written as a polynomial.⁹ Then the result of [6; Lemma3] shows that the maximal lattice dimension of the sequence $\{x_n\}$ is equal to the degree of $G(x)$. From (*) we get the polynomial identity $G(n) + G(-1-n) = b$. If we let $s = \deg(G)$, then by comparing the coefficients of the n^s term:

$$G(n) = a_s n^s + a_{s-1} n^{s-1} + \dots + a_0$$

$$G(-1-n) = a_s (-1-n)^s + a_{s-1} (-1-n)^{s-1} + \dots + a_0 = (-1)^s a_s n^s + \dots$$

Therefore $1 + (-1)^s = 0$ and s must be odd. Thus the maximal lattice dimension of $\{x_n\}$ is odd.

This result provides two simplifications to the computing of maximal lattice dimensions. First, we can skip all even dimension tests. Second, we can combine this result with the symmetry $x_n + x_{p-1-n} = b$ to shorten the dimension computation as follows:

6.2 Theorem. *The maximal lattice dimension of the IMP sequence $\{x_n\}$ is the largest odd integer $m \leq p-2$ such that:*

$$\sum_{n=0}^{(p-3)/2} (n+1)^{p-1-m} (x_n + I(x_n)) \neq 0$$

Proof. First recall (2.4) which states that ICGs pass the Marsaglia Lattice test for all d such that:

$$d \leq \max\{k \leq p-2 \mid \sum_{n \in Z_p} n^{p-1-k} x_n \neq 0\}$$

⁹Since the domain Z_p is finite, we could represent any function with a finite Legendre Polynomial. That is, if $f(x_k) = y_k$, then $f(x) = \sum_{k \in Z_p} \frac{y_k \prod_{j \neq k} x - x_j}{\prod_{j \neq k} x_k - x_j}$.

If m is the maximal lattice dimension, then the above sum will be 0 for all $p - 2 \geq k > m$. Since $x_{p-1} = 0$ and $nx_n = 0$ for $n = 0$, we make the following simplifications in the sum:

$$\sum_{n=0}^{p-1} n^{p-1-m} x_n = \sum_{n=1}^{p-2} n^{p-1-m} x_n = \sum_{n=1}^{(p-1)/2} n^{p-1-m} x_n + \sum_{(p+1)/2}^{p-2} n^{p-1-m} x_n$$

Now we re-index the second sum and apply (*) written as: $x_{n+(p-1)/2} = b - x_{(p-1)/2-n}$:

$$\begin{aligned} &= \sum_{n=1}^{(p-1)/2} n^{p-1-m} x_n + \sum_{n=1}^{(p-3)/2} (n + (p-1)/2)^{p-1-m} x_{n+(p-1)/2} \\ &= \sum_{n=1}^{(p-1)/2} n^{p-1-m} x_n + \sum_{n=1}^{(p-3)/2} (n + (p-1)/2)^{p-1-m} (b - x_{(p-1)/2-n}) \end{aligned}$$

Now consider the change of variables $n \mapsto (p-1)/2 - n$:

$$= \sum_{n=1}^{(p-1)/2} n^{p-1-m} x_n + \sum_{n=1}^{(p-3)/2} (p-1-n)^{p-1-m} (b - x_n)$$

Now pull out the $n = (p-1)/2$ term and combine the sums:

$$= \frac{(p-1)^{p-1-m}}{2} x_{(p-1)/2} + \sum_{n=1}^{(p-3)/2} (x_n(n^{p-1-m} - (p-1-n)^{p-1-m}) + (p-1-n)^{p-1-m} b)$$

Since the maximal dimension must be odd, m and consequently $p-1-m$ must also be odd,

$$= \frac{(p-1)^{p-1-m}}{2} x_{(p-1)/2} + \sum_{n=1}^{(p-3)/2} (x_n(n^{p-1-m} + (n+1)^{p-1-m}) - (n+1)^{p-1-m} b)$$

We now re-order the sum to obtain:

$$= x_1 + \sum_{n=1}^{(p-3)/2} (n+1)^{p-1-m} (x_n + x_{n+1}) - \sum_{n=1}^{(p-3)/2} (n+1)^{p-1-m} b$$

By applying the identity $x_{n+1} = ax_n^{-1} + b$:

$$= \sum_{n=0}^{(p-3)/2} (n+1)^{p-1-m} (x_n + ax_n^{-1})$$

And the desired result follows.

Corollary:. *The IMP sequence $\{x_n\}$ passes the Lattice Test for dimension $d = p - 2$ if and only if:*

$$\sum_{n=0}^{(p-3)/2} (n+1)(x_n + I(x_n)) \neq 0$$

Now, since m is the largest odd integer satisfying (6.2), we can compute m for a given IMP by checking first the dimension $p - 2$, then $p - 4$, etc. until the sum in (6.2) is not zero.

Notice that the sum now ranges over $0 \leq n \leq (p - 3)/2$ and so the number of exponentiations needed in $(n + 1)^{p-1-m}$ has been reduced by half. Ironically, this is not a great savings since the vast majority of sequences tested obtain the maximum lattice dimension, $p - 2$ as we will see in the next section.

§7 RESULTS

We now present a complete analysis of the algorithm for computing the maximal lattice dimension of all IMPs for a fixed prime. First, we fix p and choose an IMP at random using Maple, for example, to find coefficients $a, b \in Z_p^*$ so that $ICG(a, b; p)$ is maximal period. We then use a to initialize the $I_{a,p}$ matrix and the $B_{a,p}$ matrix. Next, we compute $r = b^2/a$ and $c = -r - 2$ which we use to seed Chou's Algorithm. We then proceed through the r_k -parameters generated by Chou's Algorithm, compute $b_k = B(r_k)$, and use (6.2) to compute the maximal lattice dimension of the IMP family characterized by r_k . Once Chou's Algorithm terminates, the maximal lattice dimension of all IMPs in Z_p have been computed, grouped by families. Notice that once $I_{a,p}$ and $B_{a,p}$ are initialized, the same matrices are used throughout the entire computation for fixed p .

For the current work, this algorithm was implemented entirely in the C programming language using integer arithmetic.¹⁰ Since the algorithm checks the dimension of all IMP families for a fixed prime, the task of computing for all p in $5 \leq p < 100000$ was easily distributed among the workstations of the OSU Mathematics Department by prime. A single machine held the task of monitoring jobs and submitting new ones, using Maple's GF package to generate, at random, the initial a, b parameters of a single IMP for the next prime. In this fashion, the project was loosely parallelized.

Of the 9,590 primes $5 \leq p < 100000$, there are 84,982,572 IMP families for a total of 5,579,945,320,208 IMPs. It turns out that the vast majority of the IMP families have a maximal lattice dimension of $p - 2$. In fact, of the IMP families only 1,829 families have a maximal lattice dimension of $p - 4$ and only one family, $p = 691, r = 103$, has a

¹⁰Multiplications were cast into floating point, with appropriate sanity checks, to handle overflow.

maximal lattice dimension of $p - 6$. There were no IMP families with a maximal lattice dimension lower than $p - 6$. Moreover, the distribution of these low dimension families among the primes was relatively sparse. Only 145 primes had two families with maximal lattice dimension of $p - 4$, and only 7 primes had three families with maximal lattice dimension of $p - 4$.

While this is certainly not conclusive evidence, the optimistic viewpoint would suggest that the maximal lattice dimension lower bound of $(p + 1)/2$ is improvable. The pessimistic viewpoint would suggest that given the relative scarcity of low dimension IMPs a counterexample will be difficult if not impossible to find.

§8 FURTHER APPLICATIONS

Although for the project at hand it was easiest to initialize the I matrix before computing, this is not necessary. In fact, the I matrix can be recursively defined so that the matrix can be loaded while terms of the IMP sequence are being computed. For example, when we compute $x_1 = I(x_0) + b$, we first divide x_0 into p , as in (5.1) to get $I(x_0) = I(-d_x^{-1}r_x) = -d_x I(r_x)$. Since the remainder r_x will decrease, we can implement the I matrix as a recursive program. We consider the $1 \times p$ matrix I pre-loaded with zeros and the initial values, and the recursive function $I_{a,p}$:

8.1 Definition¹¹: Recursive $I_{a,p}(x)$.

```

I_{a,p}(x): y=I[x]
    if y=0 and x != 0
        then
            solve p=dx+r
            y=-d*I_p(r) mod(p)
            I[x]=y
            I[y]=x
            I[p-x]=p-y
            I[p-y]=p-r
        endif
    return y

```

Now, when $I_{a,p}(x)$ is called, the matrix I will get recursively loaded with all of the intermediate computations. Ironically, the ICG “mixes” well enough that these interme-

¹¹See the Appendix for a C implementation

diate computations rarely overlap. That is, a significant percentage of the complete ICG sequence is often computed before a call to $I_{a,p}(x)$ returns without recursing.

However, if a large percentage of the complete IMP sequence will be needed, this recursive algorithm will provide an improvement over using direct computation of inverses since each recursive call to $I_{a,p}(x)$ is no worse than a typical inversion algorithm. In fact, the time to compute n terms of the IMP sequence with the recursive $I_{a,p}(x)$ is proportional to the logarithm of n , rather than linear. Of course the cost is memory, which forces the use of relatively small primes. However, we can partially circumvent this restriction as well. If we use distinct primes p^j and an IMP from each, we can compute the Compound ICG, or CICG found in Eichenauer-Herrmann [16]:

8.2 Definition: Compound Inversive Congruential Generator. *Let p^j be a set of distinct primes and y_n^j be a collection of IMP sequences¹² with each y_n^j in Z_{p^j} . Then the Compound ICG is given by:*

$$x_n = \sum_j \frac{y_n^j}{p^j} \pmod{1}$$

Notice that the CICG generates pseudorandom numbers in the interval $[0, 1)$. Also, since the p^j are distinct and each sequence y_n^j is an IMP, the CICG x_n will have period $\prod_j p^j$.

This generator preserves many of the good features of the ICG (see [16], [10]) and can achieve relatively long periods with small primes. Thus, we could implement the I matrix as a $j \times k$ matrix, where j is the number of primes and $k = \max_j \{p^j\}$ and write a similar recursive initialization as (8.1). In this manner, we can achieve much longer period lengths than the ICG in the same amount of memory. For example, if we wanted a generator with a period length of 10^{24} , we could use 4 primes on the order of 10^6 for a 4×10^6 matrix; contrast this to using a single ICG as in (8.1) which would require initializing a 1×10^{24} matrix. Moreover, if we wanted 10^6 terms of this generator, we could choose 6 primes on the order of 10^4 and use a 6×10^4 matrix. Then, the 6×10^4 matrix would be fully initialized after 10^4 terms of the sequence have been generated. This will significantly improve the performance of the CICG.

§9 APPENDIX - CODE AND ALGORITHMS

This program is an implementation of (8.1). It takes as input the parameters a, p, x and returns the value of $I_{a,p}(x)$ computed recursively.

¹²[16] states this definition for non-linear sequences

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int *I;

int I_p (int x, int p);

int I_p (int x, int p)
{
    int d,r,y;
    if (I[x] == 0 && x != 0)
    {
        d=p/x; r=p-d*x;
        y=((p-d)*I_p(r,p))%p;
        I[x]=y;
        I[y]=x;
        I[p-x]=p-y;
        I[p-y]=p-x;
    }
    return I[x];
}

void main (argc,argv)
int argc;
char **argv;

{
    int x, a, p;
    sscanf(argv[1],"%d",&a);
    sscanf(argv[2],"%d",&p);
    sscanf(argv[3],"%d",&x);
    I = (int *) calloc(p,sizeof(int));
    I[0]=0;I[1]=a;I[a]=1;
    I[p-a]=p-1;I[p-1]=p-a;
    printf("%d",I_p(x,p));
}
```

This program, `checkdimall`, takes as input the parameters a, b, p and returns a printout of all r -parameters of non $p - 2$ maximal lattice dimension families.

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int mult (int x, int y, int p);

int powermod (int x, int y, int p);

int gcd (int x, int y);

int mult (int x, int y, int p)
{ return((int)fmod((double)x*(double)y,(double)p));}

int gcd (int x, int y)
{
    int r;
    while (y != 0)
        { r=x%y; x=y;y=r; }
    return x;
}

int powermod (int x, int y, int p)
{
    int pow, prod;
    prod=1;
    pow=x;
    while (y != 0)
        { if (y%2 == 1) { prod=mult(prod,pow,p); }
          pow=mult(pow,pow,p);
          y=(y/2);
        }
    return prod;
}
```

```

}

void main (argc,argv)
int argc;
char **argv;

{ int r, x, mlt, dim, a, b, p, t, t0, t1, t2;
  int half_p, full, i_j, temp, a_inv;
  int i,j,k,count,nopr;
  int *B, *I;
  if (argc != 4)
  { printf("\n Usage: dim a b p \n"); exit(0);}
  sscanf(argv[1],"%d",&a);
  sscanf(argv[2],"%d",&b);
  sscanf(argv[3],"%d",&p);
  if (p%2 != 1)
  { printf("\n p must be an odd prime\n"); exit(0);}
  if ((double)p > 32*1024*1024)
  { printf("\n Hmmm... this prime may be too large!\n"); exit(0);}
  half_p=(p+1)/2;
  a_inv=powermod(a,p-2,p);
  B = (int *) calloc(p,sizeof(int));
  for (i=1;i<half_p;i++) {B[mult(mult(i,i,p),a_inv,p)]=i;}
  I = (int *) calloc(p,sizeof(int));
  count=1;j=2;I[0]=0;I[1]=a;I[a]=1;
  I[p-a]=p-1;I[p-1]=p-a;
  full=((p-1)/4)+((p-1)%4)/2+1;
  while (count < full && j < p-1)
  { if (I[j] == 0)
    { temp=p/j;
      i_j=p-mult(temp,I[p-(temp*j)],p);
      I[j]=i_j;
      I[i_j]=j;
      I[p-j]=p-i_j;
      I[p-i_j]=p-j;
      count++;
    }
  }
}

```

```

    }
    j++;
}
r=mult(mult(b,b,p),a_inv,p);
nopr=1;
t=p-(r+2)%p;
k=1;t2=t;t1=2;t0=0;
while (r != 0)
{
    dim=1;
    b=B[r];mlt=0;x=b;
    for (j=0;j<=half_p-2;j++)
    { mlt=(mlt+mult(j+1,x+I[x],p))%p;
      x=(I[x]+b)%p;
    }
    while (mlt == 0)
    { x=b;dim=dim+2;
      for (j=0;j<=half_p-2;j++)
      { mlt=(mlt+mult(powermod(j+1,dim,p),x+I[x],p))%p;
        x=(I[x]+b)%p;
      }
    }
}
if (dim != 1)
{ printf("r=%d, (a=%d, b=%d), p=%d is dim p-%d\n",r,a,b,p,dim+1); }
r=0;
while (r==0 && k<half_p-2)
{ t0=t1;t1=t2;k++;
  t2=(mult(t,t1,p)+(p-t0))%p;
  t0=t1;t1=t2;k++;
  t2=(mult(t,t1,p)+(p-t0))%p;
  temp=p-(t2+2)%p;
  if (gcd(k,p+1) == 1) { r=temp; nopr++; }
}
}
printf("total r-values: %i \n",nopr);
}

```

The following two listings are “getab.sh” and “getab.txt”. The former is a Unix shell script which takes as input a prime p and returns the coefficients a, b of an IMP over Z_p .

```
#!/bin/sh
command='readlib(GF):read('/amaterasu/sd2e/labm/petersp/ICG/code/
get_ab.txt'):get_ab('$1');quit;'
echo $command | maple -q | tail -4
```

And, the Maple segment:

```
get_ab:=proc(p) local r,a,b,alpha,beta,tau,G;
r:=rand(1..p-1):a:=r():b:=r():
while not Irreduc(x^2-b*x-a) mod(p) do a:=r(): b:=r(): od:
G:=GF(p,2,x^2-b*x-a):
beta:=G[ConvertIn](x):
alpha:=G['-'](G[ConvertIn](b),beta):
tau:=G['/'](beta,alpha):
while G[order](tau)<>p+1 do
  while not Irreduc(x^2-b*x-a) mod(p) do a:=r(): b:=r(): od:
  G:=GF(p,2,x^2-b*x-a):
  beta:=G[ConvertIn](x):
  alpha:=G['-'](G[ConvertIn](b),beta):
  tau:=G['/'](beta,alpha):
od:
print(a);
print(b);
end;
```

REFERENCES

- [1] Silicon Graphics, Inc., [Online Document] Available <http://lavarand.sgi.com> (cited 1998 Jun 1).
- [2] G. Marsaglia, *Random numbers fall mainly in the planes*, Proc. Nat. Acad. Sci. **61** (1968), 25-28.
- [3] J. Eichenauer and J. Lehn, *A non-linear congruential pseudo random number generator*, Statistische Hefte **27** (1986), 315-326.
- [4] M. Flahive and H. Niederreiter, *On inversive congruential generators for pseudorandom numbers*, Finite Fields, coding theory, and advances in communications and computing **1** (1993), 75-80.
- [5] J. Eichenauer, H. Niederreiter, *On Marsaglia's Lattice Test for Pseudorandom Numbers*, Manuscripta Mathematica **62** (1988), 245-248.
- [6] H. Niederreiter, *Remarks on nonlinear congruential pseudorandom numbers*, Metrika **35** (1988), 321-328.
- [7] Wun-Seng Chou, *On Inversive Maximal Period Polynomials over Finite Fields*, Applicable Algebra in Engineering, Communication and Computing **6** (1994), 245-250.
- [8] R. Lidl, H. Niederreiter, *Finite Fields*, Addison-Wesley, Reading, MA., 1983.
- [9] D. Dummit, R. Foote, *Abstract Algebra*, Prentice-Hall, Englewood Cliffs, NJ., 1991.
- [10] P. Hellekalek, *Inversive Pseudorandom Number Generators: Concepts, Results and Links*, Proc. 1995 Winter Simulation Conference [Online Document] see <ftp://random.mat.sbg.ac.at/pub/data/wsc95.ps> (1995), 255-262.
- [11] J. Eichenauer, H. Grothe, J. Lehn, *Marsaglia's Lattice Test and Non-Linear Congruential Pseudo Random Number Generators*, Metrika **35** (1988), 241-250.
- [12] S. Tezuka, *Uniform Random Numbers: Theory And Practice*, Kluwer Academic Publishers, 1995.
- [13] D. Lehmer, *Mathematical Methods in Large Scale Computing Units*, Proc. 2nd Symposium on Large Scale Digital Calculating Machinery (1951), 141-146, Harvard University Press, Cambridge, MA..
- [14] D. Knuth, *The Art of Computer Programming*, 2nd ed., vol. 2: Seminumerical Analysis, Addison-Wesley, Reading MA., 1981.
- [15] G. Marsaglia, *The Structure of Linear Congruential Sequences*, Applications of Number Theory to Numerical Analysis **61** (1972), 249-285, Academic Press, New York.
- [16] J. Eichenauer-Herrmann, *Compound Nonlinear Congruential Pseudorandom Numbers*, Monatshefte für Mathematik **117** (1994), 213-222, Springer-Verlag, New York.

DEPARTMENT OF MATHEMATICS, OREGON STATE UNIVERSITY, CORVALLIS OR 97331

E-mail: petersp@math.orst.edu