New perspectives on $(0, s)$-sequences

Christiane Lemieux

joint work with Henri Faure

MCQMC'08, Montreal, July 7–11

Outline

1. Definitions
   - Digital sequences
   - Original construction for $(0, s)$-sequences
   - Improvements

Outline

Outline

1. Definitions
   - Digital sequences
   - Original construction for $(0, s)$-sequences
   - Improvements
2. Framework of Generalized Niederreiter Sequences
   - Formal definition
   - Translation using generating matrices
   - Examples
3. Ideas for new constructions
   - Insight from recent work on Halton
   - 3 ideas
   - Numerical Results

Outline

1. Definitions
   - Digital sequences
   - Original construction for $(0, s)$-sequences
   - Improvements
2. Framework of Generalized Niederreiter Sequences
   - Formal definition
   - Translation using generating matrices
   - Examples
3. Ideas for new constructions
   - Insight from recent work on Halton
   - 3 ideas
   - Numerical Results
4. Conclusion

Digital sequences

Digital sequences:

Digital sequences

Digital sequences: (sorry for the repetition)...

Digital sequences

Digital sequences: (sorry for the repetition)...

Input

A (prime) base b, a dimension s, and s generating matrices $C_1, \ldots, C_s$, with elements in $\mathbb{Z}_b$.

Digital sequences

Digital sequences: (sorry for the repetition)...

Input

A (prime) base b, a dimension s, and s generating matrices $C_1, \ldots, C_s$, with elements in $\mathbb{Z}_b$.

Output (how to get the ith point)

Write $i = a_0 + a_1 \times b + a_2 \times b^2 + \ldots$ in base b;

Digital sequences

Digital sequences: (sorry for the repetition)...

Input

A (prime) base b, a dimension s, and s generating matrices $C_1, \ldots, C_s$, with elements in $\mathbb{Z}_b$.

Output (how to get the ith point)

Write $i = a_0 + a_1 \times b + a_2 \times b^2 + \ldots$ in base b; compute

$$\mathbf{y}_1 = C_1\mathbf{a},\ \mathbf{y}_2 = C_2\mathbf{a}, \ldots, \mathbf{y}_s = C_s\mathbf{a},$$

where $\mathbf{a} = [a_0, a_1, a_2, \ldots]^T$;

Digital sequences

Digital sequences: (sorry for the repetition)...

Input

A (prime) base b, a dimension s, and s generating matrices $C_1, \ldots, C_s$, with elements in $\mathbb{Z}_b$.

Output (how to get the ith point)

Write $i = a_0 + a_1 \times b + a_2 \times b^2 + \ldots$ in base b; compute

$$\mathbf{y}_1 = C_1\mathbf{a},\ \mathbf{y}_2 = C_2\mathbf{a}, \ldots, \mathbf{y}_s = C_s\mathbf{a},$$

where $\mathbf{a} = [a_0, a_1, a_2, \ldots]^T$; form

$$\begin{aligned}
u_{i1} &= 0.y_{11}y_{12}y_{13}\ldots && \text{in base } b\\
u_{i2} &= 0.y_{21}y_{22}y_{23}\ldots && \text{in base } b\\
\ldots &= \ldots\\
u_{is} &= 0.y_{s1}y_{s2}y_{s3}\ldots && \text{in base } b
\end{aligned}$$

$(0, s)$-sequences

- $(0, s)$-sequences are digital sequences for which the *t-parameter* is 0

$(0, s)$-sequences

- $(0, s)$-sequences are digital sequences for which the t-parameter is 0 $\longrightarrow$ introduced by Faure in 1982

$(0, s)$-sequences

- $(0, s)$-sequences are digital sequences for which the *t*-parameter is 0 $\longrightarrow$ introduced by Faure in 1982
- optimality w.r.t. to t, but also $\lim_{s \to \infty} c_s = 0$ in

$$D_n^* \leq c_s (\log n)^s + O((\log n)^{s-1})$$

$(0, s)$-sequences

- $(0, s)$-sequences are digital sequences for which the *t*-parameter is 0 $\rightarrow$ introduced by Faure in 1982
- optimality w.r.t. to t, but also $\lim_{s\to\infty} c_s = 0$ in

$$D_n^* \leq c_s(\log n)^s + O((\log n)^{s-1})$$

Original $(0, s)$-sequences—Faure (1982)

Take $b \geq s$ and $C_j = P^{j-1}$, where P is the (NUT) Pascal matrix in base b.

$(0, s)$-sequences

- $(0, s)$-sequences are digital sequences for which the *t*-parameter is 0 $\rightarrow$ introduced by Faure in 1982
- optimality w.r.t. to t, but also $\lim_{s\to\infty} c_s = 0$ in

$$D_n^* \leq c_s(\log n)^s + O((\log n)^{s-1})$$

Original $(0, s)$-sequences—Faure (1982)

Take $b \geq s$ and $C_j = P^{j-1}$, where P is the (NUT) Pascal matrix in base b.

Example: $b = 3$

$(0, s)$-sequences

- $(0, s)$-sequences are digital sequences for which the t-parameter is 0 $\longrightarrow$ introduced by Faure in 1982
- optimality w.r.t. to t, but also $\lim_{s\to\infty} c_s = 0$ in

$$D_n^* \leq c_s(\log n)^s + O((\log n)^{s-1})$$

Original $(0, s)$-sequences—Faure (1982)

Take $b \geq s$ and $C_j = P^{j-1}$, where P is the (NUT) Pascal matrix in base b.

Example: $b = 3$

$$C_1 = I,$$

$(0, s)$-sequences

- $(0, s)$-sequences are digital sequences for which the *t*-parameter is 0 $\longrightarrow$ introduced by Faure in 1982
- optimality w.r.t. to t, but also $\lim_{s\to\infty} c_s = 0$ in

$$D_n^* \leq c_s(\log n)^s + O((\log n)^{s-1})$$

Original $(0, s)$-sequences—Faure (1982)

Take $b \geq s$ and $C_j = P^{j-1}$, where P is the (NUT) Pascal matrix in base b.

Example: $b = 3$

$$C_1 = I, C_2 = \begin{bmatrix} 1 & 1 & 1 & \dots \\ 0 & 1 & 2 & \dots \\ 0 & 0 & 1 & \dots \end{bmatrix},$$

$(0, s)$-sequences

- $(0, s)$-sequences are digital sequences for which the t-parameter is 0 $\longrightarrow$ introduced by Faure in 1982
- optimality w.r.t. to t, but also $\lim_{s\to\infty} c_s = 0$ in

$$D_n^* \leq c_s(\log n)^s + O((\log n)^{s-1})$$

Original $(0, s)$-sequences—Faure (1982)

Take $b \geq s$ and $C_j = P^{j-1}$, where P is the (NUT) Pascal matrix in base b.

Example: $b = 3$

$$C_1 = I, C_2 = \begin{bmatrix} 1 & 1 & 1 & \dots \\ 0 & 1 & 2 & \dots \\ 0 & 0 & 1 & \dots \end{bmatrix}, C_3 = \begin{bmatrix} 1 & 2 & 1 & \dots \\ 0 & 1 & 1 & \dots \\ 0 & 0 & 1 & \dots \end{bmatrix},$$

$(0, s)$-sequences

- $(0, s)$-sequences are digital sequences for which the *t*-parameter is 0 $\rightarrow$ introduced by Faure in 1982
- optimality w.r.t. to t, but also $\lim_{s\to\infty} c_s = 0$ in

$$D_n^* \leq c_s(\log n)^s + O((\log n)^{s-1})$$

Original $(0, s)$-sequences—Faure (1982)

Take $b \geq s$ and $C_j = P^{j-1}$, where P is the (NUT) Pascal matrix in base b.

Example: $b = 3$

$$C_1 = I, C_2 = \begin{bmatrix} 1 & 1 & 1 & \dots \\ 0 & 1 & 2 & \dots \\ 0 & 0 & 1 & \dots \end{bmatrix}, C_3 = \begin{bmatrix} 1 & 2 & 1 & \dots \\ 0 & 1 & 1 & \dots \\ 0 & 0 & 1 & \dots \end{bmatrix},$$

Problem with large bases

Problem with large bases

If s is large, and therefore $b \geq s$ is large, first portion of the sequence clumps onto main diagonal because all C_j's have ones on the diagonal.

Problem with large bases

If s is large, and therefore $b \geq s$ is large, first portion of the sequence clumps onto main diagonal because all C_j's have ones on the diagonal.

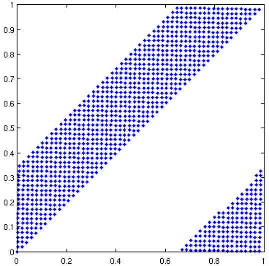


Figure: 49th and 50th coordinates of first 1000 points of original $(0, s)$-sequence in base $b = 53$

Generalizations

Generalization proposed by S. Tezuka in 1993: take

$$C_j = A_j P^{j-1},$$

where $A_1, \ldots, A_s$ are NLT matrices.

Generalizations

Generalization proposed by S. Tezuka in 1993: take

$$C_j = A_j P^{j-1},$$

where $A_1, \ldots, A_s$ are NLT matrices.
$\rightarrow$ called "Generalized Faure sequences",

Generalizations

Generalization proposed by S. Tezuka in 1993: take

$$C_j = A_j P^{j-1},$$

where $A_1, \ldots, A_s$ are NLT matrices.

$\rightarrow$ called "Generalized Faure sequences", also referred to as "deterministic linear scrambling"

Generalizations

Generalization proposed by S. Tezuka in 1993: take

$$C_j = A_j P^{j-1},$$

where $A_1, \ldots, A_s$ are NLT matrices.
$\rightarrow$ called "Generalized Faure sequences", also referred to as "deterministic linear scrambling"

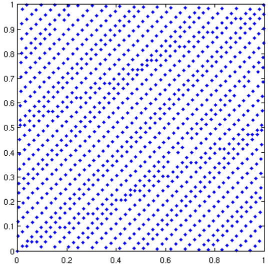


Figure: 49th and 50th coordinates of first 1000 points of generalized Faure sequence in base $b = 53$, with randomly chosen NLT A_j's

Our goal

Our goal

- How to choose those NLT matrices A_j...?

Our goal

- How to choose those NLT matrices A_j...?
- But we'll also think of the base b as something else that we can choose...

Our goal

- How to choose those NLT matrices A_j...?
- But we'll also think of the base b as something else that we can choose...
- Sobol' sequences with good direction numbers works very well: how can we do something equivalent here?

Our goal

- How to choose those NLT matrices A_j...?
- But we'll also think of the base b as something else that we can choose...
- Sobol' sequences with good direction numbers works very well: how can we do something equivalent here?
- We'll use the framework of *Generalized Niederreiter Sequences*, proposed by Tezuka

Our goal

- How to choose those NLT matrices A_j...?
- But we'll also think of the base b as something else that we can choose...
- Sobol' sequences with good direction numbers works very well: how can we do something equivalent here?
- We'll use the framework of *Generalized Niederreiter Sequences*, proposed by Tezuka
- We'll translate this framework using the language of generating matrices.

Our goal

- How to choose those NLT matrices A_j...?
- But we'll also think of the base b as something else that we can choose...
- Sobol' sequences with good direction numbers works very well: how can we do something equivalent here?
- We'll use the framework of *Generalized Niederreiter Sequences*, proposed by Tezuka
- We'll translate this framework using the language of generating matrices.
- We'll borrow ideas used for other families of constructions (Halton, Sobol', Korobov).

Framework of Generalized Niederreiter Sequences

Framework of Generalized Niederreiter Sequences

Need to work with:

- Polynomials over $\mathbb{F}_b[z]$: e.g., $4z^2 + z + 3 \in \mathbb{F}_5[z]$
- Formal Laurent series over $\mathbb{F}_b[z]$: denote this set by $\mathbb{L}_b(z)$

Framework of Generalized Niederreiter Sequences

Need to work with:

- Polynomials over $\mathbb{F}_b[z]$: e.g., $4z^2 + z + 3 \in \mathbb{F}_5[z]$
- Formal Laurent series over $\mathbb{F}_b[z]$: denote this set by $\mathbb{L}_b(z)$

example: $3z + 1 + 2z^{-1} + 4z^{-2} + z^{-4} + \ldots \in \mathbb{L}_5(z)$

Framework of Generalized Niederreiter Sequences

Need to work with:

- Polynomials over $\mathbb{F}_b[z]$: e.g., $4z^2 + z + 3 \in \mathbb{F}_5[z]$
- Formal Laurent series over $\mathbb{F}_b[z]$: denote this set by $\mathbb{L}_b(z)$

  example: $3z + 1 + 2z^{-1} + 4z^{-2} + z^{-4} + \ldots \in \mathbb{L}_5(z)$

Construction

Framework of Generalized Niederreiter Sequences

Need to work with:

- Polynomials over $\mathbb{F}_b[z]$: e.g., $4z^2 + z + 3 \in \mathbb{F}_5[z]$
- Formal Laurent series over $\mathbb{F}_b[z]$: denote this set by $\mathbb{L}_b(z)$

  example: $3z + 1 + 2z^{-1} + 4z^{-2} + z^{-4} + \ldots \in \mathbb{L}_5(z)$

Construction

- Need s polynomials $p_1(z), \ldots, p_s(z)$ in $\mathbb{F}_b[z]$

Framework of Generalized Niederreiter Sequences

Need to work with:

- Polynomials over $\mathbb{F}_b[z]$: e.g., $4z^2 + z + 3 \in \mathbb{F}_5[z]$
- Formal Laurent series over $\mathbb{F}_b[z]$: denote this set by $\mathbb{L}_b(z)$

  example: $3z + 1 + 2z^{-1} + 4z^{-2} + z^{-4} + \ldots \in \mathbb{L}_5(z)$

Construction

- Need s polynomials $p_1(z), \ldots, p_s(z)$ in $\mathbb{F}_b[z]$
- Let $e_i = \deg(p_i(z))$

Framework of Generalized Niederreiter Sequences

Need to work with:

- Polynomials over $\mathbb{F}_b[z]$: e.g., $4z^2 + z + 3 \in \mathbb{F}_5[z]$
- Formal Laurent series over $\mathbb{F}_b[z]$: denote this set by $\mathbb{L}_b(z)$

  example: $3z + 1 + 2z^{-1} + 4z^{-2} + z^{-4} + \ldots \in \mathbb{L}_5(z)$

Construction

- Need s polynomials $p_1(z), \ldots, p_s(z)$ in $\mathbb{F}_b[z]$
- Let $e_i = \deg(p_i(z))$
- For each $j = 1, \ldots, s$, need a sequence $y_{j,1}(z), y_{j,2}(z), \ldots$ of polynomials, independent within each group of size e_i.

Framework of Generalized Niederreiter Sequences

Need to work with:

- Polynomials over $\mathbb{F}_b[z]$: e.g., $4z^2 + z + 3 \in \mathbb{F}_5[z]$
- Formal Laurent series over $\mathbb{F}_b[z]$: denote this set by $\mathbb{L}_b(z)$

$$\text{example: } 3z + 1 + 2z^{-1} + 4z^{-2} + z^{-4} + \ldots \in \mathbb{L}_5(z)$$

Construction

- Need s polynomials $p_1(z), \ldots, p_s(z)$ in $\mathbb{F}_b[z]$
- Let $e_i = \deg(p_i(z))$
- For each $j = 1, \ldots, s$, need a sequence $y_{j,1}(z), y_{j,2}(z), \ldots$ of polynomials, independent within each group of size e_i.
  ex: if $e_i = 3$,

Framework of Generalized Niederreiter Sequences

Need to work with:

- Polynomials over $\mathbb{F}_b[z]$: e.g., $4z^2 + z + 3 \in \mathbb{F}_5[z]$
- Formal Laurent series over $\mathbb{F}_b[z]$: denote this set by $\mathbb{L}_b(z)$

$$\text{example: } 3z + 1 + 2z^{-1} + 4z^{-2} + z^{-4} + \ldots \in \mathbb{L}_5(z)$$

Construction

- Need s polynomials $p_1(z), \ldots, p_s(z)$ in $\mathbb{F}_b[z]$
- Let $e_i = \deg(p_i(z))$
- For each $j = 1, \ldots, s$, need a sequence $y_{j,1}(z), y_{j,2}(z), \ldots$ of polynomials, independent within each group of size e_i.
  ex: if $e_i = 3$, then want

$$y_{j,1}(z) \bmod p_j(z), y_{j,2}(z) \bmod p_j(z), , y_{j,3}(z) \bmod p_j(z)$$

to be independent.

Generalized Niederreiter Sequences (continued)

- The coefficients $a^{(j)}(k,l,r)$ in the development of
$$\frac{y_{j,l}(z)}{p_j(z)^k} = \sum_{r=w}^{\infty} a^{(j)}(k,l,r) z^{-r} \in \mathbb{L}_b(z)$$
are used to construct the generating matrices.

Generalized Niederreiter Sequences (continued)

- The coefficients $a^{(j)}(k,l,r)$ in the development of

$$\frac{y_{j,l}(z)}{p_j(z)^k} = \sum_{r=w}^{\infty} a^{(j)}(k,l,r) z^{-r} \in \mathbb{L}_b(z)$$

  are used to construct the generating matrices.
- More precisely, for $l \leq e_j$, the lth row of C_j is given by

$$a^{(j)}(1,l,1), a^{(j)}(1,l,2), \ldots$$

Generalized Niederreiter Sequences (continued)

- The coefficients $a^{(j)}(k,l,r)$ in the development of

$$\frac{y_{j,l}(z)}{p_j(z)^k} = \sum_{r=w}^{\infty} a^{(j)}(k,l,r) z^{-r} \in \mathbb{L}_b(z)$$

  are used to construct the generating matrices.

- More precisely, for $l \leq e_j$, the lth row of C_j is given by

$$a^{(j)}(1,l,1), a^{(j)}(1,l,2), \ldots$$

  i.e., coefficients from development of

$$\frac{y_{j,l}(z)}{p_j(z)}.$$

Generalized Niederreiter Sequences (continued)

- The coefficients $a^{(j)}(k,l,r)$ in the development of

$$\frac{y_{j,l}(z)}{p_j(z)^k} = \sum_{r=w}^{\infty} a^{(j)}(k,l,r) z^{-r} \in \mathbb{L}_b(z)$$

  are used to construct the generating matrices.

- More precisely, for $l \leq e_j$, the lth row of C_j is given by

$$a^{(j)}(1,l,1), a^{(j)}(1,l,2), \ldots$$

  i.e., coefficients from development of

$$\frac{y_{j,l}(z)}{p_j(z)}.$$

  and next block of e_j rows is based on the coefficients in the development of

$$\frac{y_{j,l}(z)}{(p_j(z))^2}.$$

Generalized Niederreiter Sequences (continued)

- The coefficients $a^{(j)}(k,l,r)$ in the development of

$$\frac{y_{j,l}(z)}{p_j(z)^k} = \sum_{r=w}^{\infty} a^{(j)}(k,l,r) z^{-r} \in \mathbb{L}_b(z)$$

  are used to construct the generating matrices.
- More precisely, for $l \le e_j$, the lth row of C_j is given by

$$a^{(j)}(1,l,1), a^{(j)}(1,l,2), \ldots$$

  i.e., coefficients from development of

$$\frac{y_{j,l}(z)}{p_j(z)}.$$

  and next block of e_j rows is based on the coefficients in the development of

$$\frac{y_{j,l}(z)}{(p_j(z))^2}.$$

  and so on.

Translation using generating matrices

Translation using generating matrices

So we build C_j e_j rows at a time:

Translation using generating matrices

So we build C_j e_j rows at a time:

$$\left[\begin{array}{c} \text{coefficients of } \frac{y_{j,l}(z)}{p_j(z)} \\ \hline \text{coefficients of } \frac{y_{j,l}(z)}{(p_j(z))^2} \\ \hline \vdots \end{array}\right]$$

Translation using generating matrices

So we build C_j e_j rows at a time:

$$\left[\begin{array}{c} y_{j,1}(z) \\ \vdots \qquad\qquad \text{coefficients of } \frac{y_{j,l}(z)}{p_j(z)} \\ y_{j,e_j}(z) \\ \hline \\ \qquad\qquad \text{coefficients of } \frac{y_{j,l}(z)}{(p_j(z))^2} \\ \\ \hline \vdots \end{array}\right]$$

Translation using generating matrices

So we build C_j e_j rows at a time:

$$\left[\begin{array}{ll} y_{j,1}(z) & \\ \vdots & \text{coefficients of } \frac{y_{j,l}(z)}{p_j(z)} \\ y_{j,e_j}(z) & \\ \hline y_{j,e_j+1}(z) & \\ \vdots & \text{coefficients of } \frac{y_{j,l}(z)}{(p_j(z))^2} \\ y_{j,2e_j}(z) & \\ \hline \vdots & \vdots \end{array}\right]$$

Example

Suppose $b = 2$, $p_j(z) = z^2 + z + 1$:

Example

Suppose $b = 2$, $p_j(z) = z^2 + z + 1$: Can show that

$$\frac{1}{z^2 + z + 1} = z^{-2} + z^{-3} + z^{-5} + \dots .$$

Example

Suppose $b = 2$, $p_j(z) = z^2 + z + 1$: Can show that

$$\frac{1}{z^2 + z + 1} = z^{-2} + z^{-3} + z^{-5} + \dots .$$

(use recurrence $a_r = a_{r-1} + a_{r-2}$)

Suppose $y_{i,1}(z) = 1, y_{i,2}(z) = z$,

Example

Suppose $b = 2$, $p_j(z) = z^2 + z + 1$: Can show that

$$\frac{1}{z^2 + z + 1} = z^{-2} + z^{-3} + z^{-5} + \ldots.$$

(use recurrence $a_r = a_{r-1} + a_{r-2}$)

Suppose $y_{i,1}(z) = 1, y_{i,2}(z) = z$, then

$$\left[\begin{array}{cc|ccccc} 0 & 1 & 1 & 0 & 1 & & \ldots \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{array} \right.$$

Example

Suppose $b = 2$, $p_j(z) = z^2 + z + 1$: Can show that

$$\frac{1}{z^2 + z + 1} = z^{-2} + z^{-3} + z^{-5} + \dots.$$

(use recurrence $a_r = a_{r-1} + a_{r-2}$)
Suppose $y_{i,1}(z) = 1, y_{i,2}(z) = z$, then

$$\left[\begin{array}{cc|cccc} 0 & 1 & 1 & 0 & 1 & \dots \\ 1 & 1 & 0 & 1 & 1 & \dots \\ \hline & & & & & \\ \hline & & & \dots & & \end{array}\right]$$

Example

Suppose $b = 2$, $p_j(z) = z^2 + z + 1$: Can show that

$$\frac{1}{z^2 + z + 1} = z^{-2} + z^{-3} + z^{-5} + \dots.$$

(use recurrence $a_r = a_{r-1} + a_{r-2}$)
Suppose $y_{i,1}(z) = 1, y_{i,2}(z) = z$, then

$$\left[\begin{array}{cc|cccc} 0 & 1 & 1 & 0 & 1 & \dots \\ 1 & 1 & 0 & 1 & 1 & \dots \\ \hline & & & & & \\ & & & & & \\ \hline & & & \dots & & \end{array}\right]$$

To get 3rd and 4th row, write

$$\frac{1}{(z^2 + z + 1)^2} = \frac{1}{z^4 + z^2 + 1} = z^{-4} + z^{-6} + \dots$$

Example

Suppose $b = 2$, $p_j(z) = z^2 + z + 1$: Can show that

$$\frac{1}{z^2 + z + 1} = z^{-2} + z^{-3} + z^{-5} + \dots.$$

(use recurrence $a_r = a_{r-1} + a_{r-2}$)
Suppose $y_{i,1}(z) = 1, y_{i,2}(z) = z$, then

$$\left[\begin{array}{cc|cccc} 0 & 1 & 1 & 0 & 1 & \dots \\ 1 & 1 & 0 & 1 & 1 & \dots \\ \hline & & & & & \\ & & & & & \\ \hline & & & \dots & & \end{array}\right]$$

To get 3rd and 4th row, write

$$\frac{1}{(z^2 + z + 1)^2} = \frac{1}{z^4 + z^2 + 1} = z^{-4} + z^{-6} + \dots$$

then, if $y_{j,3}(z) = 1$, $y_{j,4}(z) = z^3$, get two next rows.

Example

Suppose $b = 2$, $p_j(z) = z^2 + z + 1$: Can show that

$$\frac{1}{z^2 + z + 1} = z^{-2} + z^{-3} + z^{-5} + \dots.$$

(use recurrence $a_r = a_{r-1} + a_{r-2}$)
Suppose $y_{i,1}(z) = 1, y_{i,2}(z) = z$, then

$$\left[\begin{array}{cc|cc|ccc} 0 & 1 & 1 & 0 & 1 & & \dots \\ 1 & 1 & 0 & 1 & 1 & & \dots \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & \dots \\ 1 & 0 & 1 & 0 & 0 & 0 & \dots \\ \hline & & & \dots & & & \end{array}\right]$$

To get 3rd and 4th row, write

$$\frac{1}{(z^2 + z + 1)^2} = \frac{1}{z^4 + z^2 + 1} = z^{-4} + z^{-6} + \dots$$

then, if $y_{j,3}(z) = 1$, $y_{j,4}(z) = z^3$, get two next rows.

Example

Suppose $b = 2$, $p_j(z) = z^2 + z + 1$: Can show that

$$\frac{1}{z^2+z+1} = z^{-2} + z^{-3} + z^{-5} + \dots.$$

(use recurrence $a_r = a_{r-1} + a_{r-2}$)
Suppose $y_{i,1}(z) = 1, y_{i,2}(z) = z$, then

$$\left[\begin{array}{cc|cc|ccc} 0 & 1 & 1 & 0 & 1 & & \dots \\ 1 & 1 & 0 & 1 & 1 & & \dots \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & \dots \\ 1 & 0 & 1 & 0 & 0 & 0 & \dots \\ \hline & & & \dots & & & \end{array}\right]$$

To get 3rd and 4th row, write

$$\frac{1}{(z^2+z+1)^2} = \frac{1}{z^4+z^2+1} = z^{-4} + z^{-6} + \dots$$

then, if $y_{j,3}(z) = 1$, $y_{j,4}(z) = z^3$, get two next rows. And so on.

Example 1: Faure's 1982 construction

Example 1: Faure's 1982 construction

Here $p_j(z) = z - j + 1$ of degree 1, and $y_{j,l}(z) = 1$.

Example 1: Faure's 1982 construction

Here $p_j(z) = z - j + 1$ of degree 1, and $y_{j,l}(z) = 1$. The development of

$$\frac{1}{(z-j+1)^l}$$

gives us the coefficients for the lth row.

Example 1: Faure's 1982 construction

Here $p_j(z) = z - j + 1$ of degree 1, and $y_{j,l}(z) = 1$. The development of

$$\frac{1}{(z - j + 1)^l}$$

gives us the coefficients for the lth row. Example $b = 3$, $j = 2$:

Example 1: Faure's 1982 construction

Here $p_j(z) = z - j + 1$ of degree 1, and $y_{j,l}(z) = 1$. The development of

$$\frac{1}{(z - j + 1)^l}$$

gives us the coefficients for the lth row. Example $b = 3$, $j = 2$:

- $\frac{1}{z-1} = z^{-1} + z^{-2} + \dots$ 1 1 1 1 1 1

Example 1: Faure's 1982 construction

Here $p_j(z) = z - j + 1$ of degree 1, and $y_{j,l}(z) = 1$. The development of

$$\frac{1}{(z-j+1)^l}$$

gives us the coefficients for the lth row. Example $b = 3$, $j = 2$:

- $\frac{1}{z-1} = z^{-1} + z^{-2} + \dots$ 1 1 1 1 1 1
- $\frac{1}{(z-1)^2} = \frac{1}{z^2-2z+1} = z^{-2} + 2z^{-3} + z^{-5} + 2z^{-6} + \dots$

Example 1: Faure's 1982 construction

Here $p_j(z) = z - j + 1$ of degree 1, and $y_{j,l}(z) = 1$. The development of

$$\frac{1}{(z-j+1)^l}$$

gives us the coefficients for the lth row. Example $b = 3$, $j = 2$:

- $\frac{1}{z-1} = z^{-1} + z^{-2} + \dots$
- $\frac{1}{(z-1)^2} = \frac{1}{z^2-2z+1} = z^{-2} + 2z^{-3} + z^{-5} + 2z^{-6} + \dots$

1	1	1	1	1	1
0	1	2	0	1	2

Example 1: Faure's 1982 construction

Here $p_j(z) = z - j + 1$ of degree 1, and $y_{j,l}(z) = 1$. The development of

$$\frac{1}{(z-j+1)^l}$$

gives us the coefficients for the lth row. Example $b = 3$, $j = 2$:

- $\frac{1}{z-1} = z^{-1} + z^{-2} + \dots$
- $\frac{1}{(z-1)^2} = \frac{1}{z^2-2z+1} = z^{-2} + 2z^{-3} + z^{-5} + 2z^{-6} + \dots$
- $\frac{1}{(z-1)^3} = \frac{1}{z^3-1} = z^{-3} + z^{-6} + z^{-9} + \dots$

1	1	1	1	1	1
0	1	2	0	1	2

Example 1: Faure's 1982 construction

Here $p_j(z) = z - j + 1$ of degree 1, and $y_{j,l}(z) = 1$. The development of

$$\frac{1}{(z-j+1)^l}$$

gives us the coefficients for the lth row. Example $b = 3$, $j = 2$:

- $\frac{1}{z-1} = z^{-1} + z^{-2} + \dots$
- $\frac{1}{(z-1)^2} = \frac{1}{z^2-2z+1} = z^{-2} + 2z^{-3} + z^{-5} + 2z^{-6} + \dots$
- $\frac{1}{(z-1)^3} = \frac{1}{z^3-1} = z^{-3} + z^{-6} + z^{-9} + \dots$

1	1	1	1	1	1
0	1	2	0	1	2
0	0	1	0	0	1

Example 1: Faure's 1982 construction

Here $p_j(z) = z - j + 1$ of degree 1, and $y_{j,l}(z) = 1$. The development of

$$\frac{1}{(z-j+1)^l}$$

gives us the coefficients for the lth row. Example $b = 3$, $j = 2$:

- $\frac{1}{z-1} = z^{-1} + z^{-2} + \dots$
- $\frac{1}{(z-1)^2} = \frac{1}{z^2-2z+1} = z^{-2} + 2z^{-3} + z^{-5} + 2z^{-6} + \dots$
- $\frac{1}{(z-1)^3} = \frac{1}{z^3-1} = z^{-3} + z^{-6} + z^{-9} + \dots$
- etc

1	1	1	1	1	1
0	1	2	0	1	2
0	0	1	0	0	1

$\dots$

Example 2: Sobol'

- Take $p_j(z)$ to be the jth primitive polynomial in base 2 (sorted in increasing degree).

Example 2: Sobol'

- Take $p_j(z)$ to be the jth primitive polynomial in base 2 (sorted in increasing degree).
- Need $y_{j,1}(z), \ldots, y_{j,e_j}(z)$ to initialize each block of e_j polynomials.

Example 2: Sobol'

- Take $p_j(z)$ to be the jth primitive polynomial in base 2 (sorted in increasing degree).
- Need $y_{j,1}(z), \ldots, y_{j,e_j}(z)$ to initialize each block of e_j polynomials.
- Requirement that the "direction numbers" be odd means
  - $y_{j,1}(z)$ must be of degree $e_j - 1$
  - $y_{j,2}(z)$ must be of degree $e_j - 2$
  - ...
  - $y_{j,e_j}(z) = 1$ of degree 0

Example 2: Sobol' (continued)

Effect is that C_j is NUT:

Example 2: Sobol' (continued)

Effect is that C_j is NUT: example with $p_j(z) = z^2 + z + 1$,
$v_1 = 1/2$ and $v_2 = 3/4$

Example 2: Sobol' (continued)

Effect is that C_j is NUT: example with $p_j(z) = z^2 + z + 1$, $v_1 = 1/2$ and $v_2 = 3/4$

$$\begin{pmatrix} 1 & 1 & & \\ 0 & 1 & & \\ \hline & & & \\ \hline & & & \end{pmatrix}$$

Example 2: Sobol' (continued)

Effect is that C_j is NUT: example with $p_j(z) = z^2 + z + 1$, $v_1 = 1/2$ and $v_2 = 3/4$

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & \dots \\ 0 & 1 & 1 & 0 & 1 & \dots \\ \hline \\ \hline \\ \end{pmatrix}$$

Example 2: Sobol' (continued)

Effect is that C_j is NUT: example with $p_j(z) = z^2 + z + 1$,
$v_1 = 1/2$ and $v_2 = 3/4 \Leftrightarrow p_2(z) = 1$

$$\left(\begin{array}{cccccc} 1 & 1 & 0 & 1 & 1 & \dots \\ 0 & 1 & 1 & 0 & 1 & \dots \\ \hline \\ \\ \hline \\ \end{array}\right)$$

Example 2: Sobol' (continued)

Effect is that C_j is NUT: example with $p_j(z) = z^2 + z + 1$,
$v_1 = 1/2$ and $v_2 = 3/4 \Leftrightarrow p_2(z) = 1$ and $p_1(z) = z$

$$\left(\begin{array}{cccccc} 1 & 1 & 0 & 1 & 1 & \ldots \\ 0 & 1 & 1 & 0 & 1 & \ldots \\ \hline \\ \\ \hline \\ \end{array}\right)$$

Example 2: Sobol' (continued)

Effect is that C_j is NUT: example with $p_j(z) = z^2 + z + 1$,
$v_1 = 1/2$ and $v_2 = 3/4 \Leftrightarrow p_2(z) = 1$ and $p_1(z) = z$

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 1 & \dots \\ 0 & 1 & 1 & 0 & 1 & \dots \\ \hline \\ 0 & 0 & 0 & 1 & 0 & 1\dots \\ \hline \\ \end{pmatrix}$$

Example 2: Sobol' (continued)

Effect is that C_j is NUT: example with $p_j(z) = z^2 + z + 1$,
$v_1 = 1/2$ and $v_2 = 3/4 \Leftrightarrow p_2(z) = 1$ and $p_1(z) = z$

$$\left(\begin{array}{cccccc} 1 & 1 & 0 & 1 & 1 & \dots \\ 0 & 1 & 1 & 0 & 1 & \dots \\ \hline 0 & 0 & 1 & 0 & 1 & 0 \dots \\ 0 & 0 & 0 & 1 & 0 & 1 \dots \\ \hline & & & \vdots & & \end{array} \right)$$

Example 2: Sobol' (continued)

- Direction numbers $\Leftrightarrow$ way of initializing NUT $e_j \times e_j$ sub-matrices along diagonal (only need to choose first one, others obtained by construction); remaining entries in the rows determined by recurrence

Example 2: Sobol' (continued)

- Direction numbers $\Leftrightarrow$ way of initializing NUT $e_j \times e_j$ sub-matrices along diagonal (only need to choose first one, others obtained by construction); remaining entries in the rows determined by recurrence
- Choosing these submatrices as I_{e_j} is not good

Example 2: Sobol' (continued)

- Direction numbers $\Leftrightarrow$ way of initializing NUT $e_j \times e_j$ sub-matrices along diagonal (only need to choose first one, others obtained by construction); remaining entries in the rows determined by recurrence
- Choosing these submatrices as I_{e_j} is not good $\approx$ having ones on the diagonals of Pascal's matrices for $(0, s)$-sequences

Example 2: Sobol' (continued)

- Direction numbers $\Leftrightarrow$ way of initializing NUT $e_j \times e_j$ sub-matrices along diagonal (only need to choose first one, others obtained by construction); remaining entries in the rows determined by recurrence
- Choosing these submatrices as I_{e_j} is not good $\approx$ having ones on the diagonals of Pascal's matrices for $(0, s)$-sequences
- Various proposals for finding good direction numbers (Sobol', Cheng and Druzdzel, Kuo and Joe, L. et al. in `RandQMC`, Jaffari and Anis)

Example 2: Sobol' (continued)

- Direction numbers $\Leftrightarrow$ way of initializing NUT $e_j \times e_j$ sub-matrices along diagonal (only need to choose first one, others obtained by construction); remaining entries in the rows determined by recurrence
- Choosing these submatrices as I_{e_j} is not good $\approx$ having ones on the diagonals of Pascal's matrices for $(0, s)$-sequences
- Various proposals for finding good direction numbers (Sobol', Cheng and Druzdzel, Kuo and Joe, L. et al. in `RandQMC`, Jaffari and Anis)
- Direction numbers $\leftrightarrow$ deterministic scrambling with a block-diagonal matrix, each block being NUT.

Insight from recent work on Halton

- Generating matrices for Halton sequence are the identity, but different base in each dimension

Insight from recent work on Halton

- Generating matrices for Halton sequence are the identity, but different base in each dimension
- Generalized Halton sequences based on simple linear permutations of the digits $\Rightarrow$ use a diagonal matrix instead

Insight from recent work on Halton

- Generating matrices for Halton sequence are the identity, but different base in each dimension
- Generalized Halton sequences based on simple linear permutations of the digits $\Rightarrow$ use a diagonal matrix instead
- Even taking $C_j = f_j \times I$ gives reasonable results, if the factors f_j are well-chosen

Insight from recent work on Halton

- Generating matrices for Halton sequence are the identity, but different base in each dimension
- Generalized Halton sequences based on simple linear permutations of the digits $\Rightarrow$ use a diagonal matrix instead
- Even taking $C_j = f_j \times I$ gives reasonable results, if the factors f_j are well-chosen
- Our recent work: take f_j so that

Insight from recent work on Halton

- Generating matrices for Halton sequence are the identity, but different base in each dimension
- Generalized Halton sequences based on simple linear permutations of the digits $\Rightarrow$ use a diagonal matrix instead
- Even taking $C_j = f_j \times I$ gives reasonable results, if the factors f_j are well-chosen
- Our recent work: take f_j so that
  1. corresponding one-dimensional generalized van der Corput sequence has good properties w.r.t. L_2-discrepancy (upper bounds, results of Faure 2006)

Insight from recent work on Halton

- Generating matrices for Halton sequence are the identity, but different base in each dimension
- Generalized Halton sequences based on simple linear permutations of the digits $\Rightarrow$ use a diagonal matrix instead
- Even taking $C_j = f_j \times I$ gives reasonable results, if the factors f_j are well-chosen
- Our recent work: take f_j so that
  1. corresponding one-dimensional generalized van der Corput sequence has good properties w.r.t. L_2-discrepancy (upper bounds, results of Faure 2006)
  2. computer search to make sure two-dimensional projections are good

Recent work on Halton (continued)

- More precisely: Faure (2006) shows bound of the form

$$L_2(N, S_b^{\Sigma}) \leq \frac{\theta_b^f}{\log b} \log N + \Theta_b^f + \frac{1}{12}$$

Recent work on Halton (continued)

- More precisely: Faure (2006) shows bound of the form

$$L_2(N, S_b^{\Sigma}) \leq \frac{\theta_b^f}{\log b} \log N + \Theta_b^f + \frac{1}{12}$$

where S_b^{Σ} is the van der Corput sequence in base p with linear permutations $\Sigma = (\pi_r)_{r \geq 0}$ based on factors g for which $\theta_b^g \leq \theta_b^f$.

Recent work on Halton (continued)

- More precisely: Faure (2006) shows bound of the form

$$L_2(N, S_b^{\Sigma}) \leq \frac{\theta_b^f}{\log b} \log N + \Theta_b^f + \frac{1}{12}$$

  where S_b^{Σ} is the van der Corput sequence in base p with linear permutations $\Sigma = (\pi_r)_{r \geq 0}$ based on factors g for which $\theta_b^g \leq \theta_b^f$.
- $\Rightarrow$ can sort factors $f \in \{1, \ldots, b-1\}$ according to θ_b^f

Recent work on Halton (continued)

- More precisely: Faure (2006) shows bound of the form

$$L_2(N, S_b^{\Sigma}) \leq \frac{\theta_b^f}{\log b} \log N + \Theta_b^f + \frac{1}{12}$$

where S_b^{Σ} is the van der Corput sequence in base p with linear permutations $\Sigma = (\pi_r)_{r \geq 0}$ based on factors g for which $\theta_b^g \leq \theta_b^f$.

- $\Rightarrow$ can sort factors $f \in \{1, \ldots, b-1\}$ according to θ_b^f $\Rightarrow$ minimize bound on L_2-discrepancy of one-dimensional van der Corput sequence

Recent work on Halton (continued)

Search algorithm

for $j = 1$ to s

Recent work on Halton (continued)

Search algorithm

for $j = 1$ to s

- build list L of best M factors according to $\theta^{f}_{p_j}$, where p_j is the jth prime number

Recent work on Halton (continued)

Search algorithm

for $j = 1$ to s

- build list L of best M factors according to $\theta^f_{p_j}$, where p_j is the jth prime number
- select $f_j = f \in L$ which minimizes

$$\tau_j(N, W) = \max_{1 \le l \le W} L_2(N, (S^{f_{j-l}}_{p_{j-l}}, S^f_{p_j}))$$

Recent work on Halton (continued)

Search algorithm

for $j = 1$ to s

- build list L of best M factors according to $\theta^f_{p_j}$, where p_j is the jth prime number
- select $f_j = f \in L$ which minimizes

$$\tau_j(N, W) = \max_{1 \le l \le W} L_2(N, (S^{f_{j-l}}_{p_{j-l}}, S^f_{p_j}))$$

- makes sure two-dimensional projections of nearby indices are good

Recent work on Halton (continued)

Search algorithm

for $j = 1$ to s

- build list L of best M factors according to $\theta^f_{p_j}$, where p_j is the jth prime number
- select $f_j = f \in L$ which minimizes

$$\tau_j(N, W) = \max_{1 \le l \le W} L_2(N, (S^{f_{j-l}}_{p_{j-l}}, S^f_{p_j}))$$

- makes sure two-dimensional projections of nearby indices are good
- criterion similar to the one used to find our good direction numbers for Sobol' sequence, replacing L_2-discrepancy by *resolution* (t-parameter restricted to cubic boxes)

Idea # 1

Take $A_j = f_j \times I$ where the f_j's are chosen similarly as in approach for Halton. More precisely, build list L of m best ones ($m \approx b/2$) according to θ_b^f, and then proceed as follows:

Idea # 1

Take $A_j = f_j \times I$ where the f_j's are chosen similarly as in approach for Halton. More precisely, build list L of m best ones ($m \approx b/2$) according to θ_b^f, and then proceed as follows:

Search algorithm

for $j = 1$ to s

Idea # 1

Take $A_j = f_j \times I$ where the f_j's are chosen similarly as in approach for Halton. More precisely, build list L of m best ones ($m \approx b/2$) according to θ_b^f, and then proceed as follows:

Search algorithm

for $j = 1$ to s

- select $f_j = f \in L$ that minimizes

$$\tau_j(N, W) = \max_{1 \leq l \leq W} L_2(N, (S_b^{f_{j-l}}, S_b^f))$$

Idea # 1

Take $A_j = f_j \times I$ where the f_j's are chosen similarly as in approach for Halton. More precisely, build list L of m best ones ($m \approx b/2$) according to θ_b^f, and then proceed as follows:

Search algorithm

for $j = 1$ to s

- select $f_j = f \in L$ that minimizes

$$\tau_j(N, W) = \max_{1 \leq l \leq W} L_2(N, (S_b^{f_{j-l}}, S_b^f))$$

Examples below done with $N = 2500$ and $W = 7$.

Idea # 2

- Choose b around $2s$ and then take f_j as the jth best factor, for $j = 1, \ldots, s$

Idea # 2

- Choose b around $2s$ and then take f_j as the jth best factor, for $j = 1, \ldots, s$
- Means we only use the best upper half of the factors

Idea # 2

- Choose b around $2s$ and then take f_j as the jth best factor, for $j = 1, \ldots, s$
- Means we only use the best upper half of the factors
- No need to search, by contrast with Idea # 1

Idea # 2

- Choose b around $2s$ and then take f_j as the jth best factor, for $j = 1, \ldots, s$
- Means we only use the best upper half of the factors
- No need to search, by contrast with Idea # 1
- Goes against the habit of taking $b \geq s$ as small as possible...

Idea # 3

Idea # 3

...or how to use a $(0, s)$-sequence construction for problems of unbounded dimension

Idea # 3

...or how to use a $(0, s)$-sequence construction for problems of unbounded dimension

Idea: can fix b and then use any s, even $s > b$

Idea # 3

...or how to use a $(0, s)$-sequence construction for problems of unbounded dimension

Idea: can fix b and then use any s, even $s > b$

- Generating matrices have a periodic behavior, i.e., $C_j = C_{j+b} = C_{j+2b} = \ldots$

Idea # 3

...or how to use a $(0, s)$-sequence construction for problems of unbounded dimension

Idea: can fix b and then use any s, even $s > b$

- Generating matrices have a periodic behavior, i.e., $C_j = C_{j+b} = C_{j+2b} = \dots$
- Similar to using a Korobov lattice with $n < s$

Idea # 3

...or how to use a $(0, s)$-sequence construction for problems of unbounded dimension

Idea: can fix b and then use any s, even $s > b$

- Generating matrices have a periodic behavior, i.e., $C_j = C_{j+b} = C_{j+2b} = \dots$
- Similar to using a Korobov lattice with $n < s$
- Can be broken by "scrambling" with $A_j = f_{(j \bmod m)} \times I$ where $m \approx b/2$, and $f_{(1)}, f_{(2)}, \dots$ are sorted according to θ_b^f,

Idea # 3

...or how to use a $(0, s)$-sequence construction for problems of unbounded dimension

Idea: can fix b and then use any s, even $s > b$

- Generating matrices have a periodic behavior, i.e., $C_j = C_{j+b} = C_{j+2b} = \dots$
- Similar to using a Korobov lattice with $n < s$
- Can be broken by "scrambling" with $A_j = f_{(j \bmod m)} \times I$ where $m \approx b/2$, and $f_{(1)}, f_{(2)}, \dots$ are sorted according to θ_b^f, i.e., cycle over list of best half portion of factors

Idea # 3

...or how to use a $(0, s)$-sequence construction for problems of unbounded dimension

Idea: can fix b and then use any s, even $s > b$

- Generating matrices have a periodic behavior, i.e., $C_j = C_{j+b} = C_{j+2b} = \dots$
- Similar to using a Korobov lattice with $n < s$
- Can be broken by "scrambling" with $A_j = f_{(j \bmod m)} \times I$ where $m \approx b/2$, and $f_{(1)}, f_{(2)}, \dots$ are sorted according to θ_b^f, i.e., cycle over list of best half portion of factors
- Not a $(0, s)$-sequence anymore, but any projection of the form $P_n(\{i_1, \dots, i_k\})$ with $i_k - i_1 < b$ and $n = b^m$ is a $(0, m, k)$-net

Idea # 3

...or how to use a $(0, s)$-sequence construction for problems of unbounded dimension

Idea: can fix b and then use any s, even $s > b$

- Generating matrices have a periodic behavior, i.e., $C_j = C_{j+b} = C_{j+2b} = \ldots$
- Similar to using a Korobov lattice with $n < s$
- Can be broken by "scrambling" with $A_j = f_{(j \bmod m)} \times I$ where $m \approx b/2$, and $f_{(1)}, f_{(2)}, \ldots$ are sorted according to θ_b^f, i.e., cycle over list of best half portion of factors
- Not a $(0, s)$-sequence anymore, but any projection of the form $P_n(\{i_1, \ldots, i_k\})$ with $i_k - i_1 < b$ and $n = b^m$ is a $(0, m, k)$-net
- Can be used on problems where s is huge, with a reasonably large base b.

Results

- Use simple digital shifts for randomization

Results

- Use simple digital shifts for randomization
- Compare Sobol', Sobol with unit DN (Sob-1), Faure, Gen. Faure Idea 1 (GF-1), Gen. Faure Idea 2 (GF-2), Halton (H), Gen. Halton (GH), Monte Carlo

Results

- Use simple digital shifts for randomization
- Compare Sobol', Sobol with unit DN (Sob-1), Faure, Gen. Faure Idea 1 (GF-1), Gen. Faure Idea 2 (GF-2), Halton (H), Gen. Halton (GH), Monte Carlo
- Mortgage-backed securities (two sets of parameters, $s = 360$), Digital option $s = 75$,

Results

- Use simple digital shifts for randomization
- Compare Sobol', Sobol with unit DN (Sob-1), Faure, Gen. Faure Idea 1 (GF-1), Gen. Faure Idea 2 (GF-2), Halton (H), Gen. Halton (GH), Monte Carlo
- Mortgage-backed securities (two sets of parameters, $s = 360$), Digital option $s = 75$, test-functions g_1, g_2 and g_3, where

Results

- Use simple digital shifts for randomization
- Compare Sobol', Sobol with unit DN (Sob-1), Faure, Gen. Faure Idea 1 (GF-1), Gen. Faure Idea 2 (GF-2), Halton (H), Gen. Halton (GH), Monte Carlo
- Mortgage-backed securities (two sets of parameters, $s = 360$), Digital option $s = 75$, test-functions g_1, g_2 and g_3, where

$$g_1(\mathbf{u}) = \prod_{j=1}^{s} c(u_j - 0.5) \qquad c = 0.25, s = 96$$

Results

- Use simple digital shifts for randomization
- Compare Sobol', Sobol with unit DN (Sob-1), Faure, Gen. Faure Idea 1 (GF-1), Gen. Faure Idea 2 (GF-2), Halton (H), Gen. Halton (GH), Monte Carlo
- Mortgage-backed securities (two sets of parameters, $s = 360$), Digital option $s = 75$, test-functions g_1, g_2 and g_3, where

$$g_1(\mathbf{u}) = \prod_{j=1}^{s} c(u_j - 0.5) \qquad c = 0.25, s = 96$$

$$g_2(\mathbf{u}) = \prod_{j=1}^{s} \frac{|4u_j - 2| + (s-j)^2}{1 + (s-j)^2} \qquad s = 75$$

Results

- Use simple digital shifts for randomization
- Compare Sobol', Sobol with unit DN (Sob-1), Faure, Gen. Faure Idea 1 (GF-1), Gen. Faure Idea 2 (GF-2), Halton (H), Gen. Halton (GH), Monte Carlo
- Mortgage-backed securities (two sets of parameters, $s = 360$), Digital option $s = 75$, test-functions g_1, g_2 and g_3, where

$$g_1(\mathbf{u}) = \prod_{j=1}^{s} c(u_j - 0.5) \qquad c = 0.25, s = 96$$

$$g_2(\mathbf{u}) = \prod_{j=1}^{s} \frac{|4u_j - 2| + (s-j)^2}{1 + (s-j)^2} \qquad s = 75$$

$$g_3(\mathbf{u}) = \alpha_s \pi^{-s/2} \cos\left(\sqrt{\frac{1}{2}\sum_{j=1}^{s}[\Phi^{-1}(u_j)]^2}\right), s = 120$$

- Idea # 3 will be used on a separate problem

MBS problem

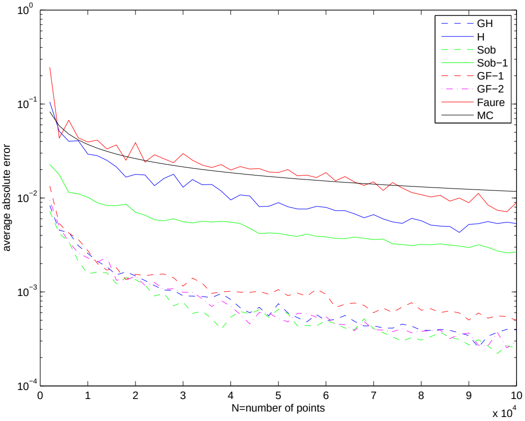


Figure: “Almost linear” case: easy problem

MBS problem

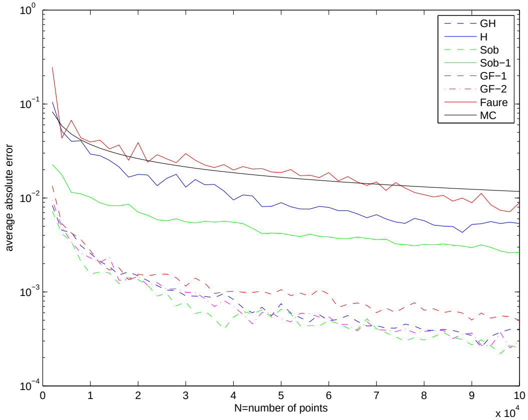


Figure: “Almost linear” case: easy problem

GF-1 and espcially GF-2 are competitive with Sobol’ and Gen. Halton

MBS problem (continued)

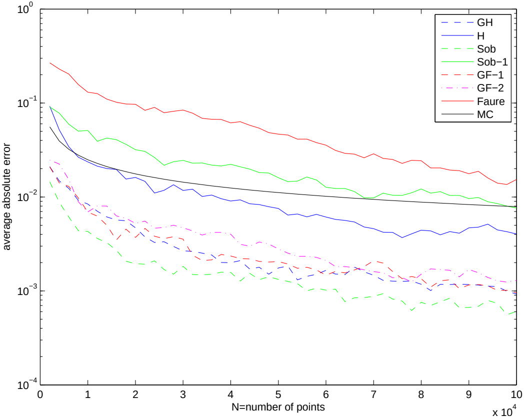


Figure: "non-linear" case: more difficult

MBS problem (continued)

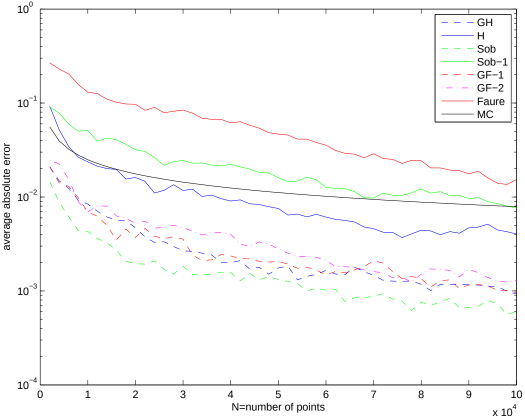
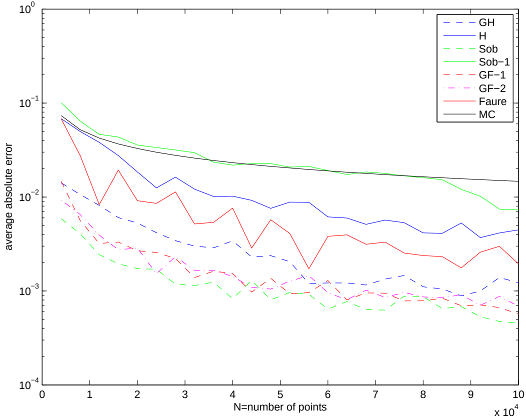


Figure: "non-linear" case: more difficult

Improved sequences much better; naive versions can be worse than MC

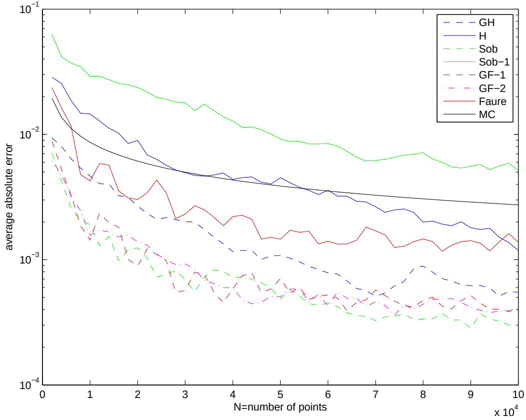
Digital option

- Used by Papageorgiou as a counter-example for advantage of Brownian Bridge techniques
- Payoff of the form $\frac{1}{s}\sum_{j=1}^{s} S(t_j)\mathbf{1}(S(t_j) > S(t_{j-1}))$.



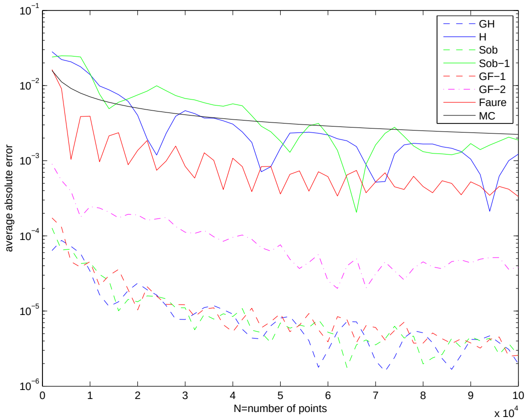
Test function g_1

- effective dimension in superposition sense is about 6
- but all variables are equally important



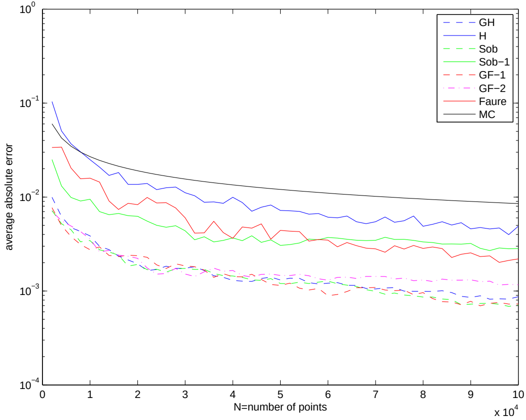
Test function g_2

- most important variables all the last ones



Test function g_3

- isotropic function
- not a product



Queueing problem

- estimate number of clients among first 1000 who waited more than 5 minutes in a single-server system

Queueing problem

- estimate number of clients among first 1000 who waited more than 5 minutes in a single-server system $s = 2000$
- two server speeds; Poisson arrival 1/minute
- $n = 8191$ and choose good Korobov generator for this n
- GF-3 (Idea #3) with $b = 727$
- report average and half-width of 95% CI

	45 seconds	55 seconds
MC	138.29	540.61
	0.25	0.61
Kor.	138.35	540.69
	0.15	0.25
GF-3	138.25	540.94
	0.20	0.28

Conclusion

- Have used the framework of generalized Niederreiter sequences to establish useful analogies between different families of constructions
- Have proposed three ways of constructing "generalized Faure sequences"
- Numerical tests suggest these proposals are almost as good as Sobol' sequences
- Our Idea #3 (use generalized Faure sequences with $b < s$) widens the range of applications for these constructions