HP-SEE

Randomized QMC Algorithms Approximating Eigenvalues

www.hp-see.eu

Aneta Karaivanova and Alexander Marazov
IICT-BAS
Sofia, Bulgaria
anet at parallel dot bas dot bg

HP-SEE

High-Performance Computing Infrastructure
for South East Europe's Research Communities

Overview

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Introduction: MCMs, QMCMs and randomized QMCMs
- Purpose of scrambling
- Scrambling techniques
- Scrambling algorithms for Faure sequence
- QMCMs for extreme eigenvalues
- Numerical tests

Introduction

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Stochastic numerical methods (*Monte Carlo methods*) are based on simulation of random variables/processes and estimation of their statistical properties. They have some advantages for high dimensional problems, problems in complicated domains or when we are interested in part of the solution.
- *Quasi-Monte Carlo methods* are deterministic methods which use low discrepancy sequences. For some problems they offer higher precision and faster convergence.
- *Randomized quasi-Monte Carlo methods* use randomized (scrambled) quasirandom sequences. They combine the advantages of Monte Carlo and quasi-Monte Carlo.

Markov chain based problems

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Consider the following problem :

  $u = Ku + f$

- The formal solution is the truncated Neumann series (for $||K||<1$):

  $u_{k+1} = f + Kf + \ldots + K^{k-1}f + K^k u_0$

  with truncation error $u_k - u = K^k (u_0 - u)$.

- Two main types of the operator K:
  - (i) K is an n x n matrix, u and f are vectors;
  - (ii) K is an integral transformation

- Compute the scalar product $J(u) = (h,u)$, h – given vector
- Define r.v. θ such that $E[\theta] = J(u)$:

  $\theta[h] = h(\xi_0)/\pi(\xi_0) \Sigma_{j=0}^{\infty} Q_j f(\xi_j)$, j=1,2,…

  Here $\xi_0, \xi_1, \ldots$ is a Markov chain (random walk) in G with initial density π(x) and transition density p(x,y), which is equal to the normalized kernel of the integral operator.

- We have to estimate the mathematical expectation

MCM accuracy

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- The MCM convergence rate is $N^{-1/2}$ with sample size N ($\varepsilon \approx \sigma(\theta)N^{-1/2}$);
  - Probabilistic result – there is no absolute upper bound.
  - The statistical distribution of the error is a normal random variable.
- The MCM error and the sample size are connected by:

  $\varepsilon = O(\sigma N^{-1/2})$, $N = O(\sigma/\varepsilon)^2$
- The computing time is proportional to N, i.e., it increases very fast if a better accuracy is needed.
- How to increase the convergence:
  - Variance reduction
  - Change of the underlying sequence
- In this talk we consider improvement through sequence optimization

Quasirandom sequences

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- The quasirandom sequences are deterministic sequences constructed to be as uniform as mathematically possible (and, as a consequence, to ensure better convergence for the integration)
- The uniformity is measured in terms of discrepancy which is defined in the following way: For a sequence with N points in $[0,1]^s$ define

$$R_N(J) = 1/N\#\{x_n \text{ in } J\}\text{-vol}(J) \text{ for every } J \subset [0,1]^s$$

$$D_N^* = \sup_{E^*} |R_N(J)|,$$

E* - the set of all rectangles with a vertex in zero.

- A sequence is called quasirandom if

$$D_N^* \leq c(\log N)^s N^{-1}$$

- Koksma-Hlawka inequality (for integration):

$$\varepsilon[f] \leq V[f] D_N^*$$

(where V[f] is the variation in the sense of Hardy-Kraus)

- The order of the error is $O((\log N)^s N^{-1})$

Quasirandom walk

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Quasirandom walk error:

  $\delta(\zeta) = \lim_{N\to\infty}(\zeta(\omega_i) - \int_\Omega \zeta(\omega)d\mu(\omega))$

  where $\zeta(\omega_i)$ – the estimated variable is the analog of r.v.; ω_i – an element of the quasirandom walks space
- *Chelson's theorem for quasirandom walks* :

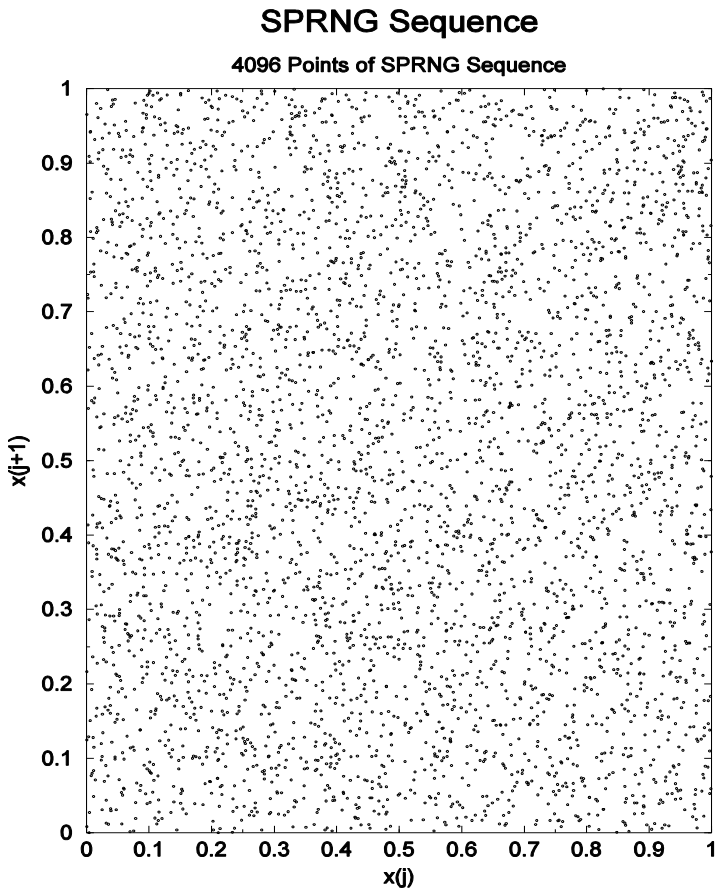
  $\delta_N(\zeta(Q')) \le V(\zeta \circ \Gamma^{-1}).(D^*_N(Q))$

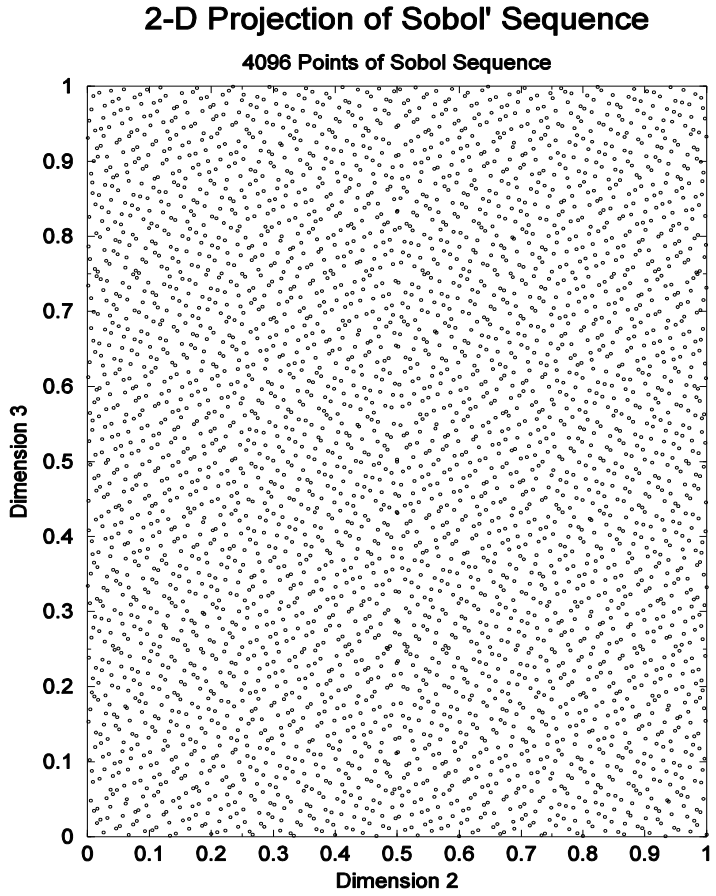
  where $Q = \{\gamma_i\}$ is a sequence of vectors in $[0,1)^{dT}$, $Q' = \{\omega_i\}$ is a sequence of quasirandom walks generated from Q by the mapping Γ;
  - There is a convergence
  - Impractical error as:

    $D^*_N = O((\log N)^{dT}/N)$, where d is the dimension of the original problem and T is the length of the chain

PRNs and QRNs

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities





Quasirandom Sequences and their scrambling

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Star discrepancy:
  - Quasirandom sequences: $D_N^* < c\,(\log N)^s\, N^{-1}$
  - Random numbers: $D_N^* = O\left((\log\log N)^{-1/2}\, N^{-1/2}\right)$
- A few quasirandom sequences are currently widely used: Halton, Faure, Niederreiter and Sobol'
- Unfortunately, the coordinates of the points in high dimensions show correlations. A possible solution to this problem is the so-called scrambling.
- The purpose of scrambling:
  - To improve 2-D projections and the quality of quasirandom sequences in general
  - To provide practical method to obtain error estimates for QMC
  - To provide simple and unified way to generate quasirandom numbers for parallel, distributed and grid-based computing environments
  - To provide more choices of QRN sequences with better (often optimal) quality to be used in QMC applications

Scrambling techniques

HP-SEE
High-Performance Computing Infrastructure for South East European Research Communities

- Scrambling was first proposed by Cranley and Patterson (1979) who took lattice points and randomized them by adding random shifts to the sequences. Later, Owen (1998, 2002, 2003) and Tezuka (2002) independently developed two powerful scrambling methods through permutations
- Although many other methods have been proposed, most of them are modified or simplified Owen or Tezuka schemes (Braaten and Weller, Atanassov, Matousek, Chi and Mascagni, Warnock, etc.)
- There are two basic scrambling methods:
  - Randomized shifting
  - Digital permutations
- The problem with Owen scrambling is its computational complexity

Scrambling

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Digital permutations: Let $(x^{(1)}_n, x^{(2)}_n, \ldots, x^{(s)}_n)$ be any quasirandom number in $[0, 1)^s$, and $(z^{(1)}_n, z^{(2)}_n, \ldots, z^{(s)}_n)$ is its scrambled version. Suppose each $x^{(j)}_n$ has a b-ary representation $x^{(j)}_n, =0. x^{(j)}_{n1}\ x^{(j)}_{n2} \ldots x^{(j)}_{nK}, \ldots$ with K defining the number of digits to be scrambled. Then

  $z^{(j)}_n = \sigma(x^{(j)}_n)$, where $\sigma=\{\Phi_1, \ldots, \Phi_K\}$ и Φ_i, is a uniformly chosen permutation of the digits {0,1,…,b−1}.

- Randomized shifting has the form

  $$z_n = x_n + r \pmod 1,$$

  where x_n is any quasirandom number in $[0, 1)^s$ and r is a single s-dimensional pseudorandom number.

The Halton Sequence

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Let n be an integer presented in base p. The p-ary radical inverse function is defined as

$$\phi_p(n) \equiv \frac{b_0}{p} + \frac{b_1}{p^2} + \ldots + \frac{b_m}{p^{m+1}}$$

where p is prime and b_i comes from

$$n = b_0 + b_1 p + \ldots + b_m p^m, \qquad \text{with } 0 \le b_i < p$$

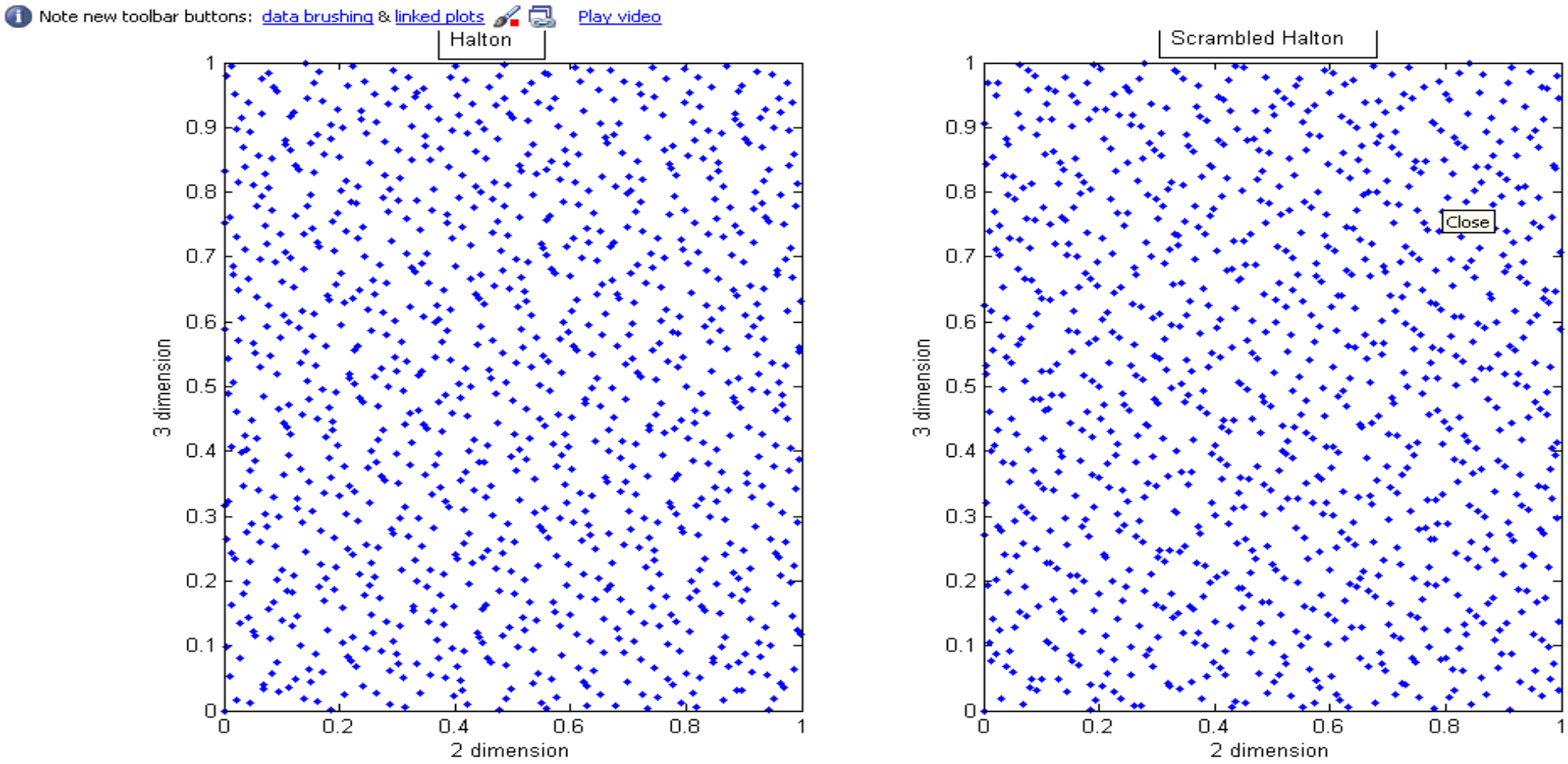
- An s-dimensional Halton sequence is defined as:

$$(\phi_{p_1}(n), \phi_{p_2}(n), \ldots, \phi_{p_s}(n))$$

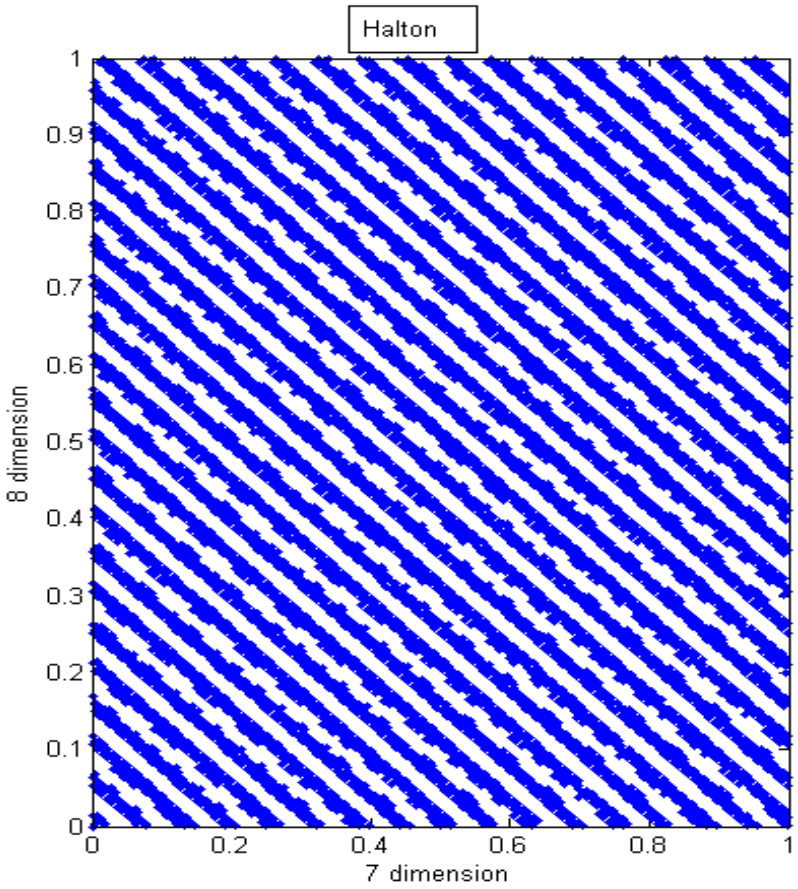
with $p_1\, p_2\, \ldots, p_s$ being relatively prime, and usually the first s primes

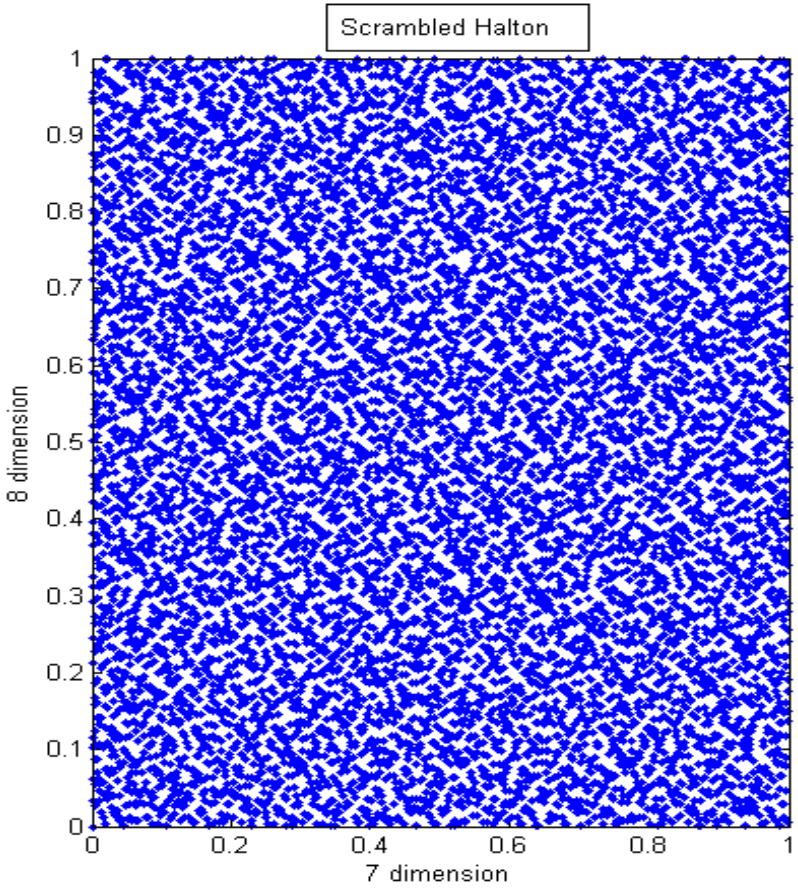
Two-dimensional projection of Halton sequence and scrambled Halton sequence (dimension 3)

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

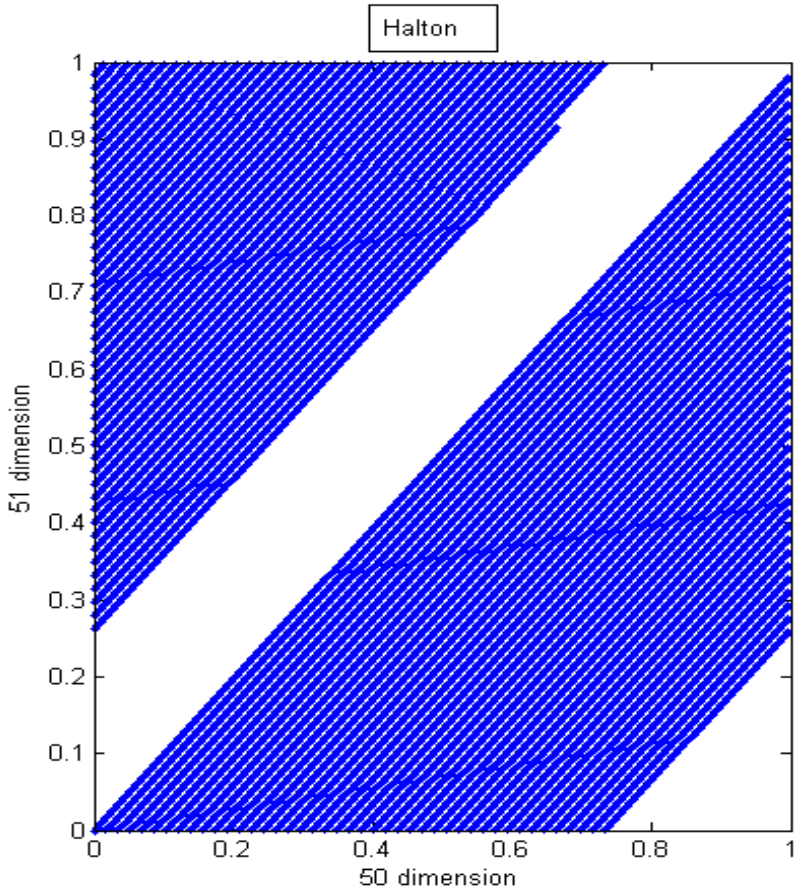


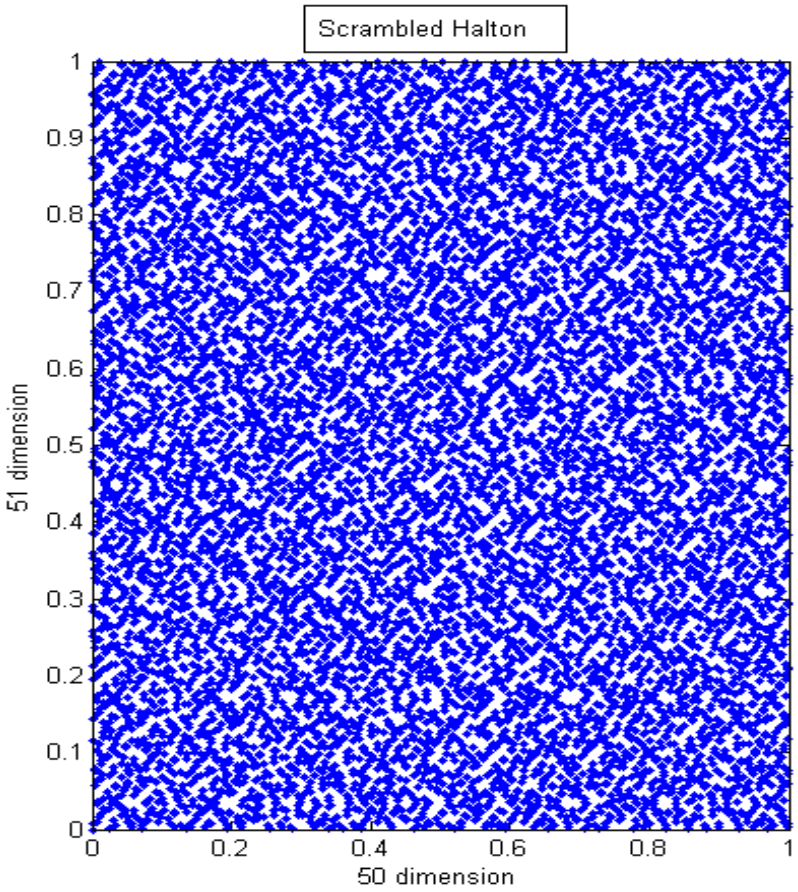
Two-dimensional projection of Halton sequence and scrambled Halton sequence (dimension 8)



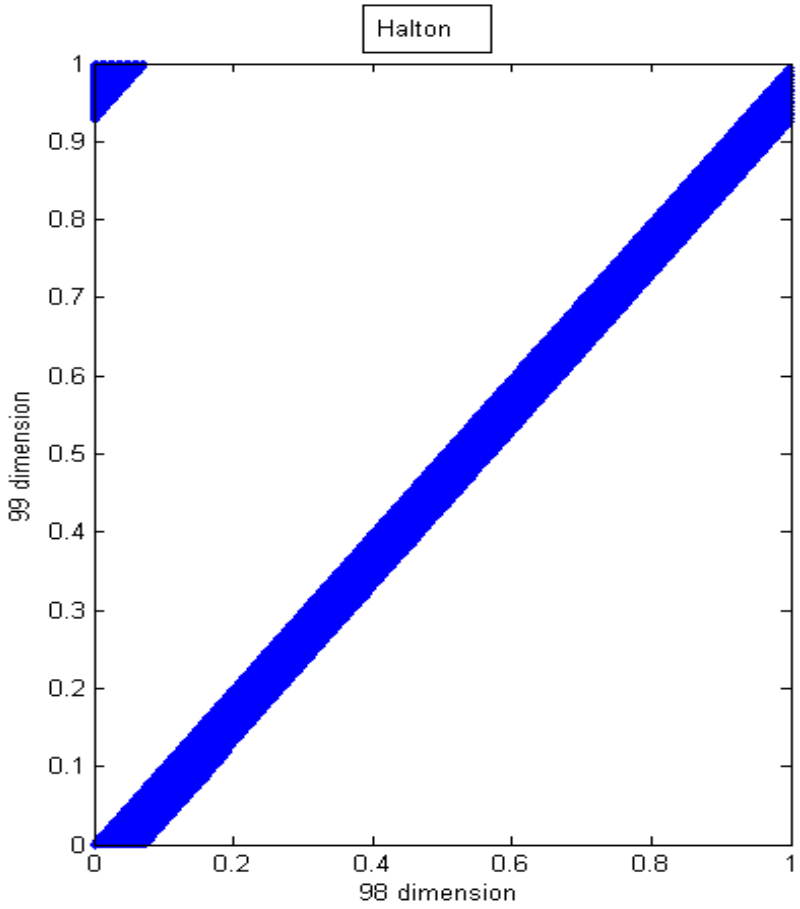


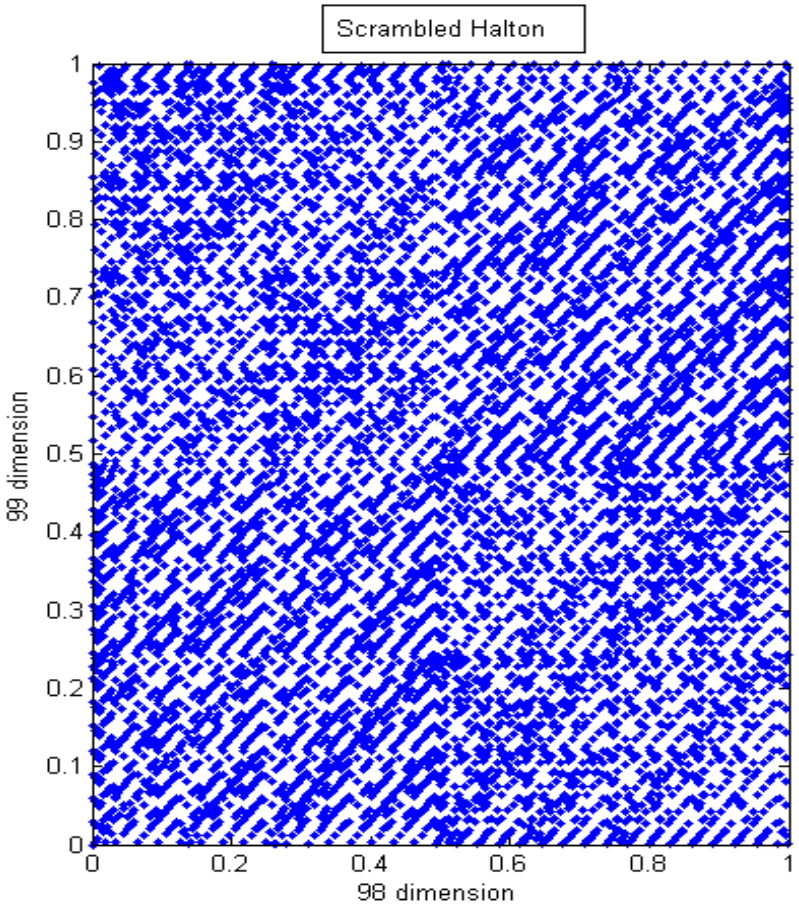
Two-dimensional projection of Halton sequence and scrambled Halton sequence (dimension 50)





Two-dimensional projection of Halton sequence and scrambled Halton sequence (dimension 99)





The Faure sequence

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Faure sequence (x_1, x_2,…, x_3,…), 1982

$x_n=(\Phi_b(P^0\mathbf{n}), \Phi_b(P^1\mathbf{n}), \ldots, \Phi_b(P^{s-1}\mathbf{n}))$

where the generator matrix P is the Pascal matrix modulo b, i.e. $P(i,j) = C_{i-1}^{j-1} \bmod b$

- Generalized Faure sequence, Gfaure (Tezuka, 1994-95)

$x_n=(\Phi_b(A^{(1)}P^0\mathbf{n}), \Phi_b(A^{(2)}P^1\mathbf{n}), \ldots, \Phi_b(A^{(s)}P^{s-1}\mathbf{n}))$,

where $A^{(j)}$ is arbitrary non-singular lower triangular matrices

  - A special case (Faure, 2000) for A: all entries are equal to 1 for all j's

- Scrambled Faure sequence

Scrambled Faure Sequence

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Owen scrambling (Owen 1995, 2000, 2003)
- Random linear scrambling (Matousek, 1997)

$x_n=(\Phi_b(A^{(1)}P^0\mathbf{n}+\mathbf{g}_1), \Phi_b(A^{(2)}P^1\mathbf{n}+\mathbf{g}_2), \ldots, \Phi_b(A^{(s)}Ps^{-1}\mathbf{n}+\mathbf{g}_s),$

where A's are different random scrambling matrices and g's are different random shifts vectors

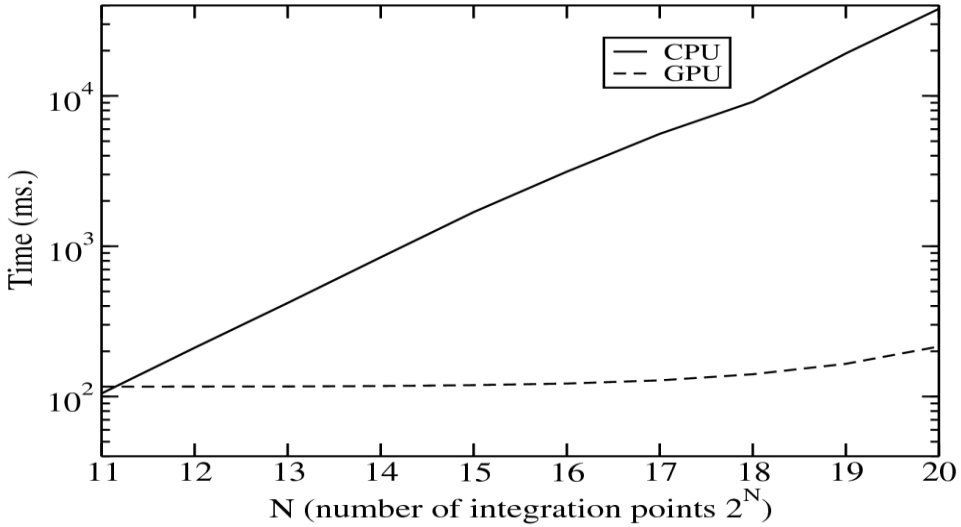
- Faure sequence with I-binomial scrambling
  - I-binomial refers to a property of the generator matrices
- Nonlinear scrambling Vanderwoerstyne&Chi, 2010)

$x_n^{(j)}=(\Phi_b(A^{(j)}\Psi(P^0\mathbf{n})+\mathbf{g}_j),$

where Ψ is a bijection that maps the non-zero digits of a digit vector to their multiplicative inverse modulo b and leaves the zero-digits unchanged.

Full scrambling

- Owen type of scrambling preserves the star discrepancy but is very time consuming
- We have developed GPU-based algorithms for Owen type of scrambling for Sobol (2010, Atanassov, Karaivanova, Ivanovska) and for Halton (2013, Atanassov and Durchova)
- With these algorithms we achieved a reasonable time to produce the scrambled sequences



QMCM application in LA

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

Given a matrix A (n x n) and n-dimensional vector f, consider the problems:

- ✓ Find the scalar product (g,x) of the unknown solution of a system of linear algebraic equations with a given vector, where the system is:

$$x = Ax + f$$

- ✓ Find an element of the inverse matrix $C = A^{-1}$
- ✓ Estimate the eigenpairs (λ,x), $x \in R^n$, such that

$$Ax=\lambda x$$

Neumann series for SLAE

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Consider $x=Ax+f$, all eigenvalues of A lie within unit square
- The truncated Neumann series gives the approximate solution of the system (given $x^{(0)}$):

  $x^{(k)} = f + Af + A^2f + \dots A^kx^{(0)}$, $k>0$
- The problem: estimate (h,x), where h – given vector:

  $(h,x) \approx h^T f + h^T Af + h^T A^2f + \dots h^T A^kx^{(0)}$
- We can apply MCM and QMC for matrix-vector products estimation

Power method for largest eigenvalue

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Consider the eigenvalue problem:

$$Au = \lambda u \ , \ A \in R^{n \times n}, \ u \in R^n$$

$$|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_{n-1}| > |\lambda_n|.$$

- The power method:

$$x^m = Ax^{m-1} / \ ||Ax^{m-1}||$$

$$\lambda^{(m)} = (h, A^m f) / \ (h, A^{m-1} f) \rightarrow \lambda_{max}$$

- We can apply MCM and QMCM to estimate matrix-vector products

Power method for resolvent matrix

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Resolvent matrix associated with A: $R_q=[I-qA]^{-1} \in R^{nxn}$

  $[I-qA]^{-m} = \Sigma_1^{\infty} q^i C_{m+i-1}^{\ i} A^i$, $|q\lambda| < 1$
- The eigenvalues of R and A are connected:

  $$\mu = 1/(1-q\lambda)$$

  $q > 0$, μ_{max} corresponds to λ_{max}

  $q < 0$, μ_{max} corresponds to λ_{min}
- Power method for resolvent matrix gives:

$$\mu^{(m)} = (h, [I-qA]^{-m}f)/ (h, [I-qA]^{-(m-1)}f) \rightarrow \mu = 1/(1-q\lambda)$$

$$\lambda = \{\Sigma_1 q^i C_{m+i-2}^{\ i-1} (h, A^i f)\}/\{\Sigma_0 q^i C_{m+i-1}^{\ i} (h, A^i f)\}$$

MCM for matrix-vector product

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- In order to compute the scalar product $(h, A^m f) = h^T A^m f$, we have to construct:

Markov chain: $k_0 \rightarrow k_1 \rightarrow \ldots \rightarrow k_m \quad (1 \leq k_i \leq n)$

with initial and transition density:

$p_\alpha = 1/n, \quad p_{\alpha\beta} = 1/n$ (crude MCM)

$p_\alpha = |h_\alpha| / \Sigma_1^n |h_\alpha|; \; p_{\alpha\beta} = |a_{\alpha\beta}| / \Sigma_1^n |a_{\alpha\beta}|;$

Random variable $\theta = (h_{k0}/p_{k0}) W_m f_{k0}$

where $W_0 = 1$, $W_j = W_{j-1}(a_{k_{j-1}k_j}/p_{k_{j-1}k_j})$, $j = 1, \ldots, m$

$E\theta = h^T A^m f \approx 1/N \; \Sigma_1^N (\theta)_s$

QMCM for matrix-vector product

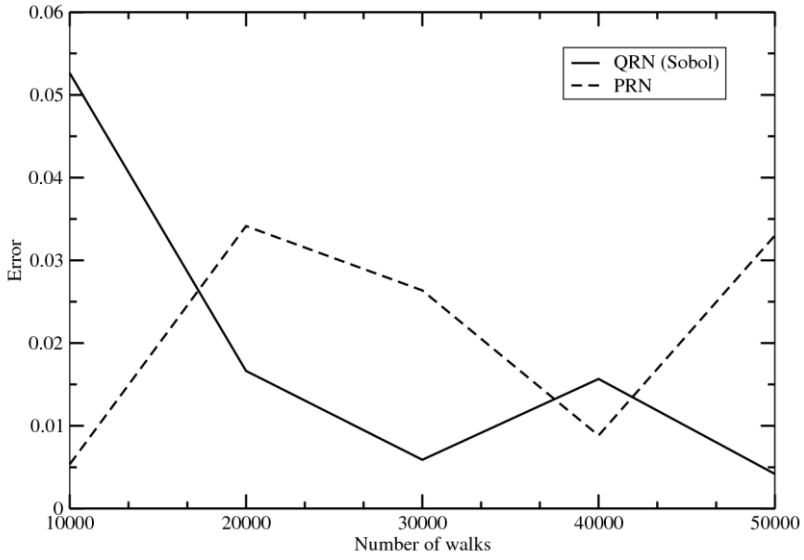
HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

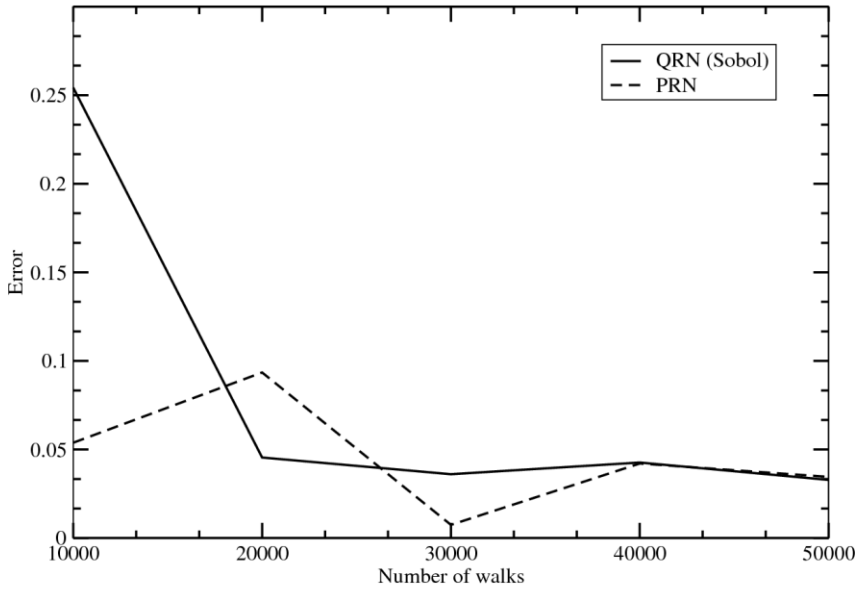
- $h^T A^i f$ in an (i+1)-dimensional integral
- Define the sets $G=[0,n)$, $G_i=[i-1,i)$, $i=1,\ldots,n$, and the piece wise continuous functions $f(x)=f_i$, $x\in G_i$, $i=1,\ldots n$, $a(x,y)=a_{ij}$, $x\in G_i$, $y\in G_j$, $i,j=1,\ldots,n$, $h(x)=h_i$, $x\in G_i$, $i=1,\ldots,n$.
- We choose $p(x)=p_i$, $x\in G_i$, $(\Sigma p_i=1)$, $p(x,y)= p_{ij}$, $x\in G_i$, $y\in G_j$, $(\Sigma p_{ij}=1)$, then:
- $p(y_0,y_1,\ldots, y_i)=p(y_0)p(y_0,y_1)\ldots p(y_i,y_i)$
- $h^T A^i f = \int_{G_0} \ldots \int_{G_i} p_i(y_0,y_1,\ldots, y_i)\, h(y_0)\, a(y_0,y_1)\ldots a(y_{i-1},y_i)\ldots f(y_i)\, dy_0 dy_1 \ldots dy_i)$
- $h_N^T A_N^i f_N - 1/N \Sigma\ldots \leq |h|^T |A|^i |f| D_N^*(y_0,y_1,\ldots, y_i)$

Numerical results

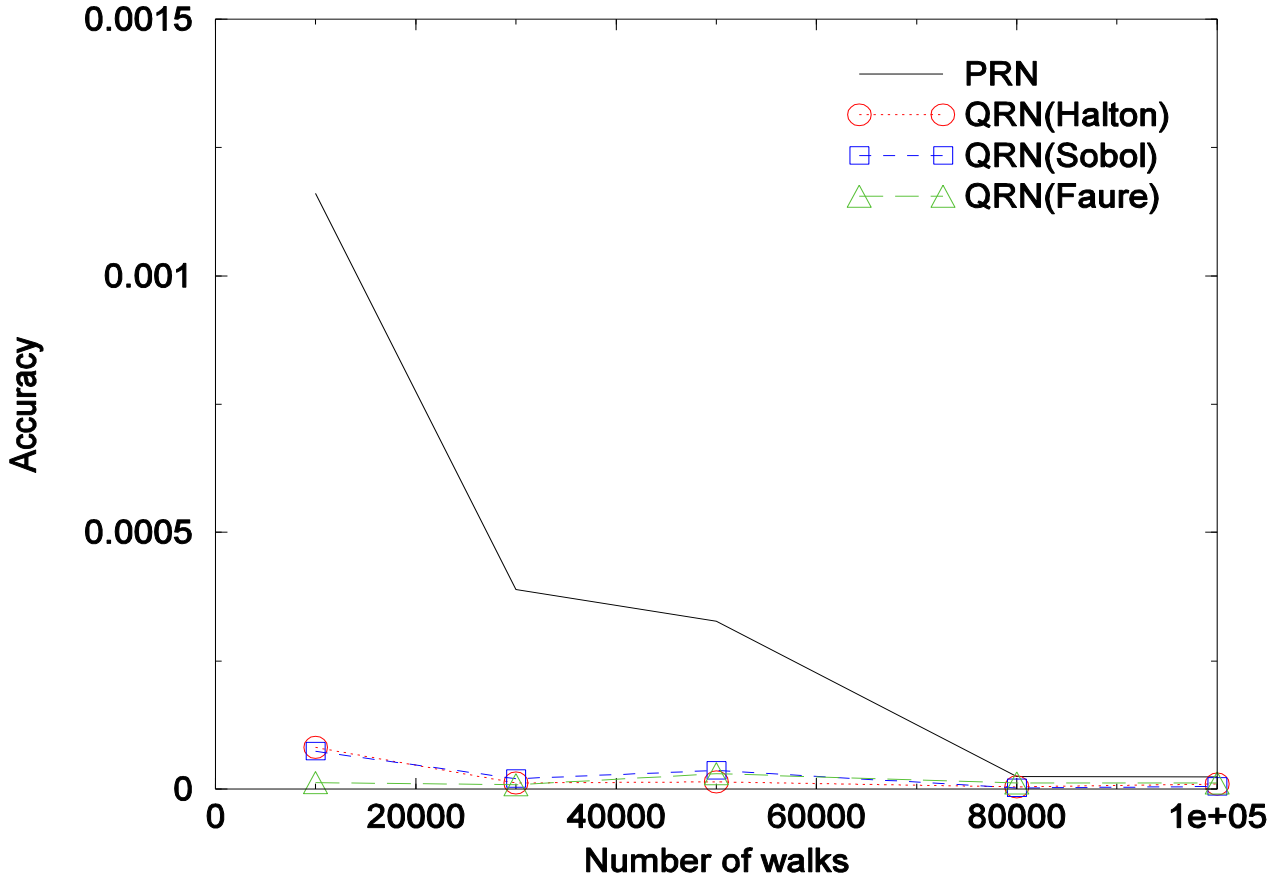
HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Accuracy vs number of walks for computing $h^T A^k f$ for k=5 and k=10 (n = 1280)

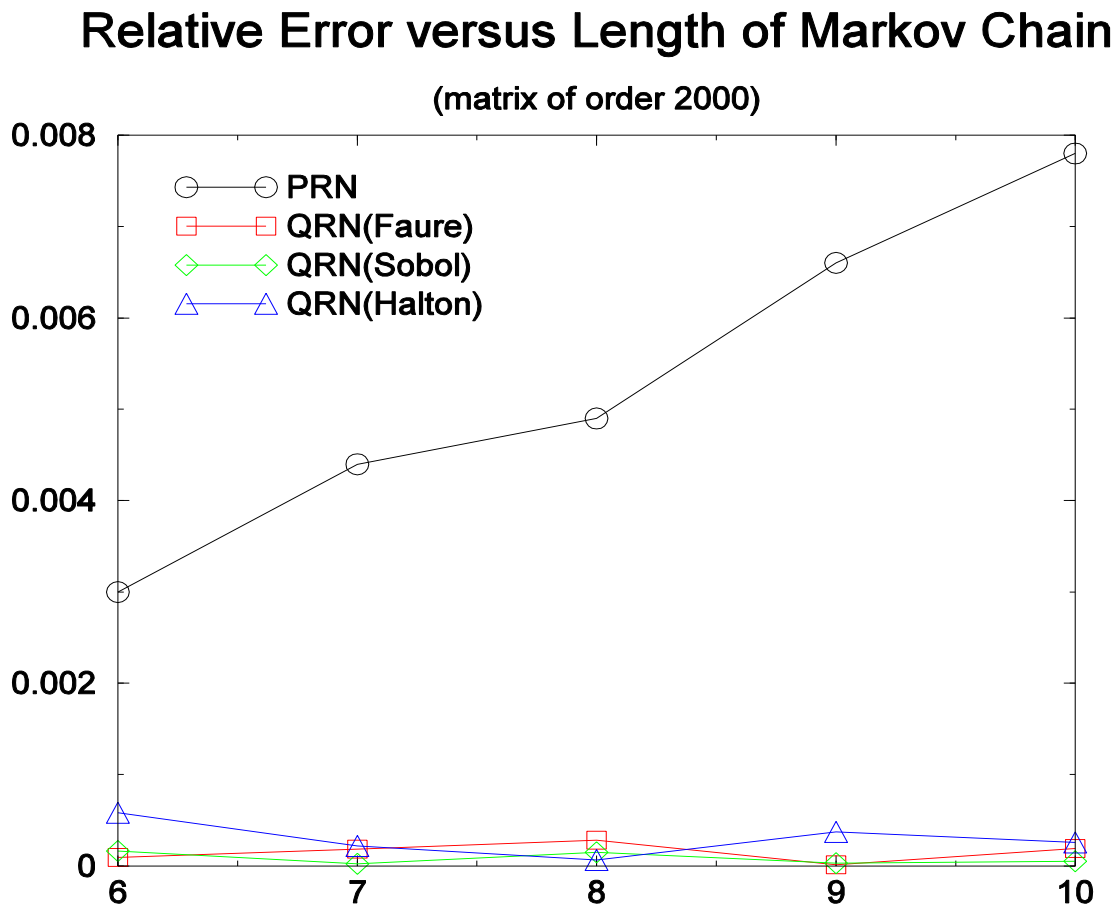




Numerical tests: solving SLAE (n=1024)



Numerical tests: Resolvent method for largest eigenvalue (n=1024)



Conclusion

HP-SEE
High-Performance Computing Infrastructure for South East Europe's Research Communities

- Randomized QMCMs are suitable for Markov chain based problems:
  - Improved 2D projections (necessary for Neumann series)
  - Automatic error estimation
- The simplified algorithms for scrambled Sobol and Halton sequences are 10 times faster than PRNs generators
- Show excellent parallel performance for integral (up to dimension 100) and integral equations on various computing environments (clusters, supercomputer, GPU)
- More experiments in LA problems have to be performed