Valuation of Mortgage-Backed Securities in a Distributed Environment

Overview

- What is a Mortgage-Backed Security
- Monte-Carlo valuation method
- Weighted Brownian Bridge discretization
- Dimensionality Reduction using ANOVA analysis
- Numerical Results
- Parallel Implementation in .NET

What is a MBS?

- Claim to the cashflows generated by a group of one or more mortgages
- MBS is created when a mortgage issuer (typically GNMA, FHLMC or FNMA) pools together a set of mortgages
- Issuer sells units to investors directly or through securities markets
- Ownership in a MBS entitles the owner to mortgage interest and principal payments minus fees for guaranteeing timely payment, even in case of default
- Owner of MBS is subjected to interest and prepayment risks but not default risk

Prepayment factors

- Refinancing incentives
  - Difference between borrower and market interest rates
  - Measured as Weighted Average Coupon / Pool Refinancing Rate
- Age of mortgage (Seasoning)
  - Prepayments increase as time passes
  - Homeowners aren't likely to move soon after purchasing a house
- Seasonality (Yearly trends)
  - People move more frequently during the summer months
- Premium burnout
  - Spike in refinancing is followed by burnout. Borrowers with poor credit history, declining house value, etc. are unaffected by further incentives
- Exogenous Factors
  - The three D's: Death, Divorce, Destruction. Poor credit history.

MBS Prepayment Models

- Classical (Brennan and Schwartz 1985)
  - Rational borrower minimizing PV of the mortgage
  - In practice borrowers are suboptimal agents. Model poorly matches actual prepayment rates.
- Reduced form (Caflsich, Morokoff and Owen 1997)
  - Prepayment model estimated from historical data
  - Prepayment is a random process
  - Borrowers may not be optimal in structural sense
- Optimal Recursive Refinancing (Longstaff 2002)
  - Refinancing when rates drop sufficiently
  - Key factors: cost of refinancing and borrower's credit

Model Details

- Interest rates – Brownian motion with no drift

$$i_k = K_0 e^{\varepsilon_k} i_{k-1} = K_0^k i_0 e^{(\varepsilon_1 + \cdots + \varepsilon_k)} \qquad \varepsilon_k \sim N(0, \sigma^2),\ K_0 = e^{-\sigma^2/2}$$

- Prepayment rate

$$w_k = K_1 + K_2 \arctan(K_3 i_k + K_4)$$
$$= K_1 + K_2 \arctan(K_3 K_0^k i_0 e^{(\varepsilon_1 + \cdots + \varepsilon_k)} + K_4)$$

- Present Value of MBS

$$PV = \sum_{k=1}^{M} u_k m_k$$

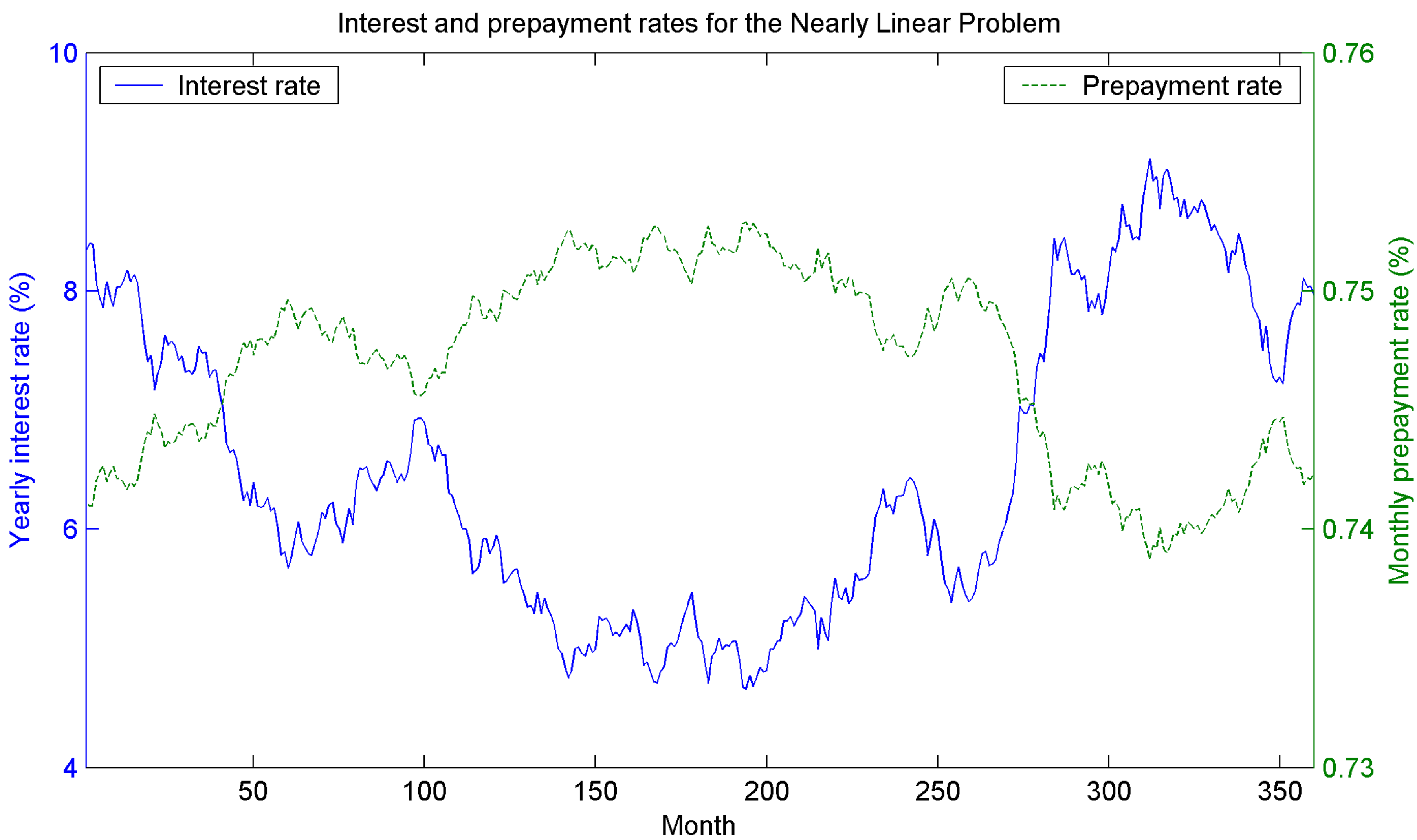
discount rate at month *k* u_k

cashflow at month *k* $m_k = g(w_k)$

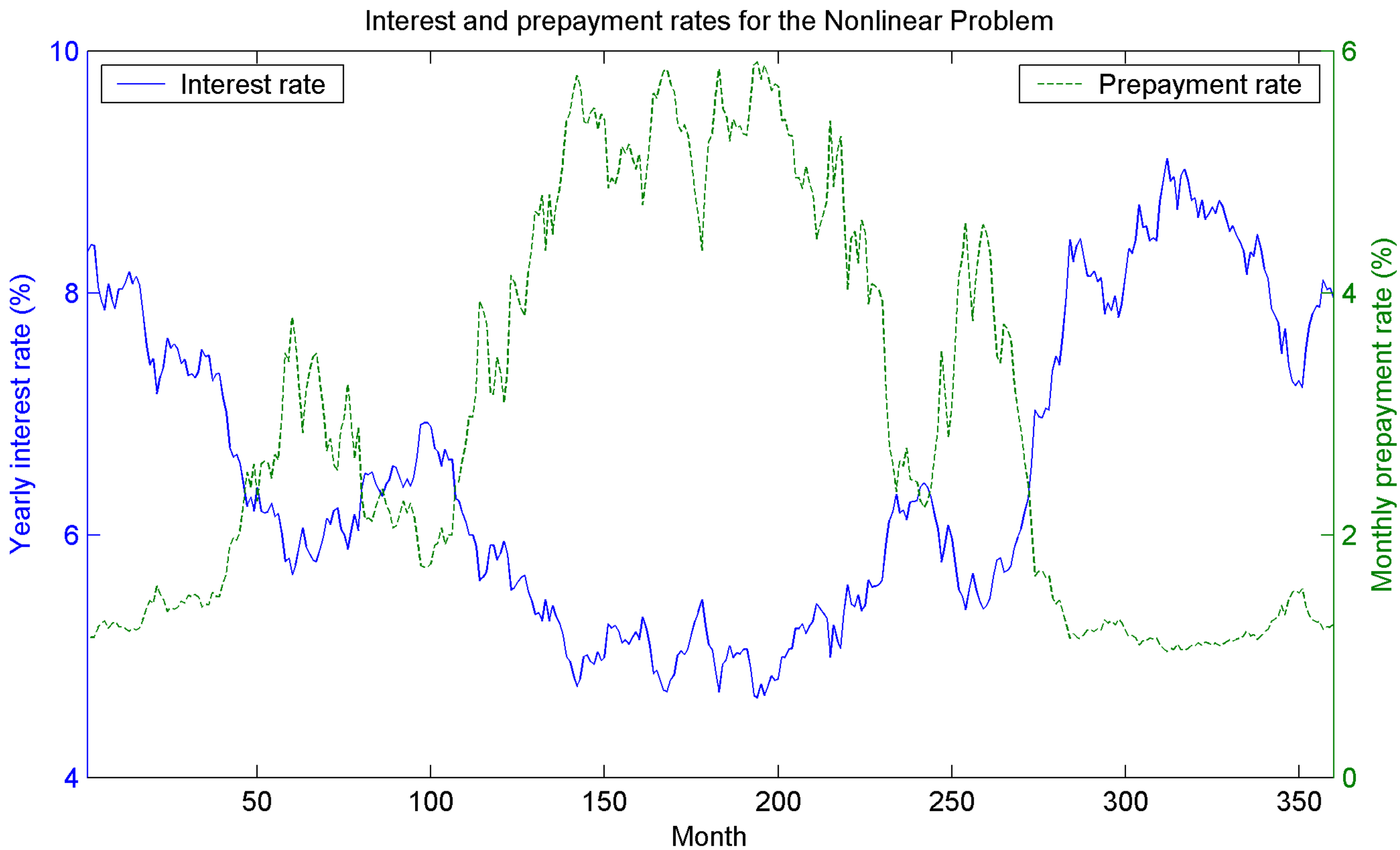
- Numerical examples
  - Nearly Linear problem

    $$(i_0, K_1, K_2, K_3, K_4, \sigma^2) = (0.07, 0.01, -0.005, 10, 0.5, 0.0004)$$
  - Nonlinear problem

    $$(i_0, K_1, K_2, K_3, K_4, \sigma^2) = (0.07, 0.04, 0.0222, -1500, 7.0, 0.0004)$$



Interest and prepayment rates for the Nearly Linear Problem
Interest rate
Prepayment rate
Yearly interest rate (%)
Monthly prepayment rate (%)
Month
10
8
6
4
0.76
0.75
0.74
0.73
50
100
150
200
250
300
350

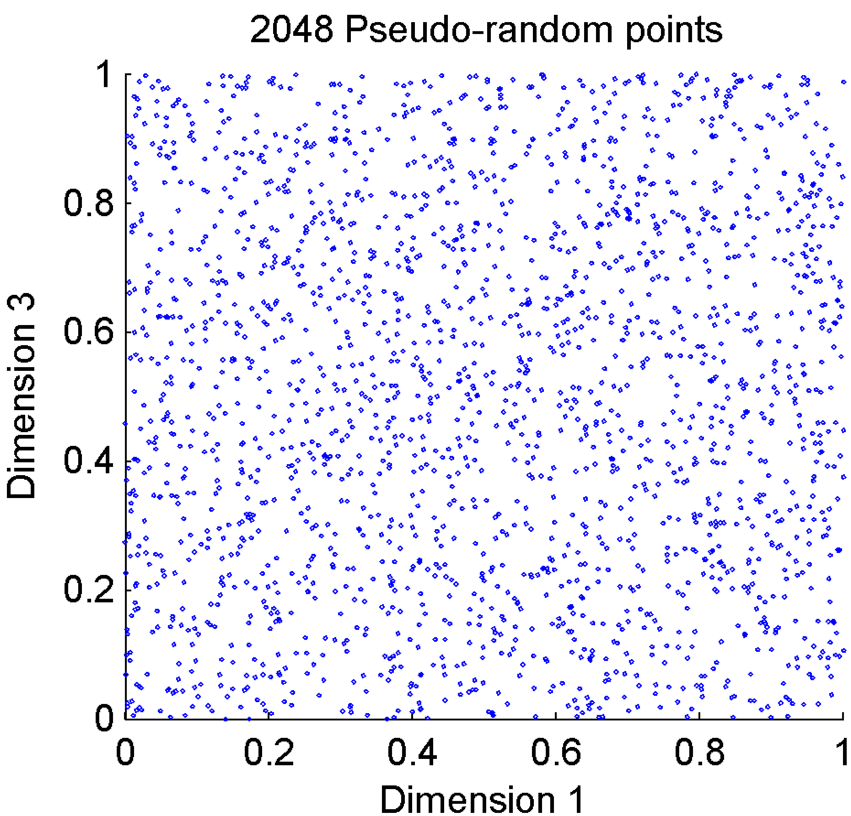


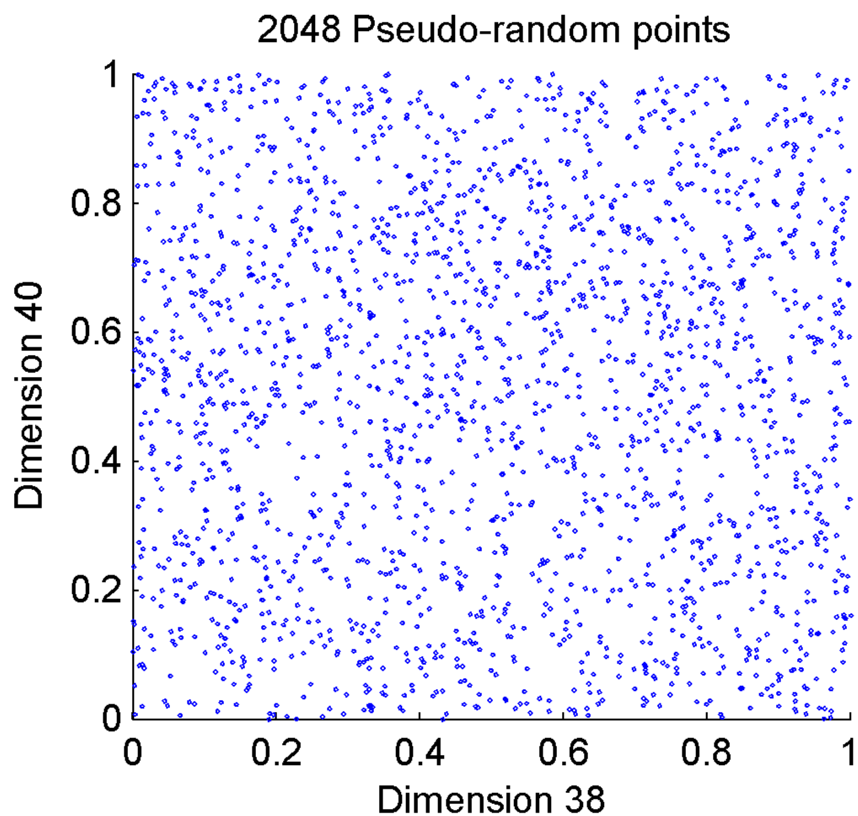
Interest and prepayment rates for the Nonlinear Problem
Interest rate
Prepayment rate
Yearly interest rate (%)
Monthly prepayment rate (%)
Month
10
8
6
4
6
4
2
0
50
100
150
200
250
300
350

Monte Carlo method

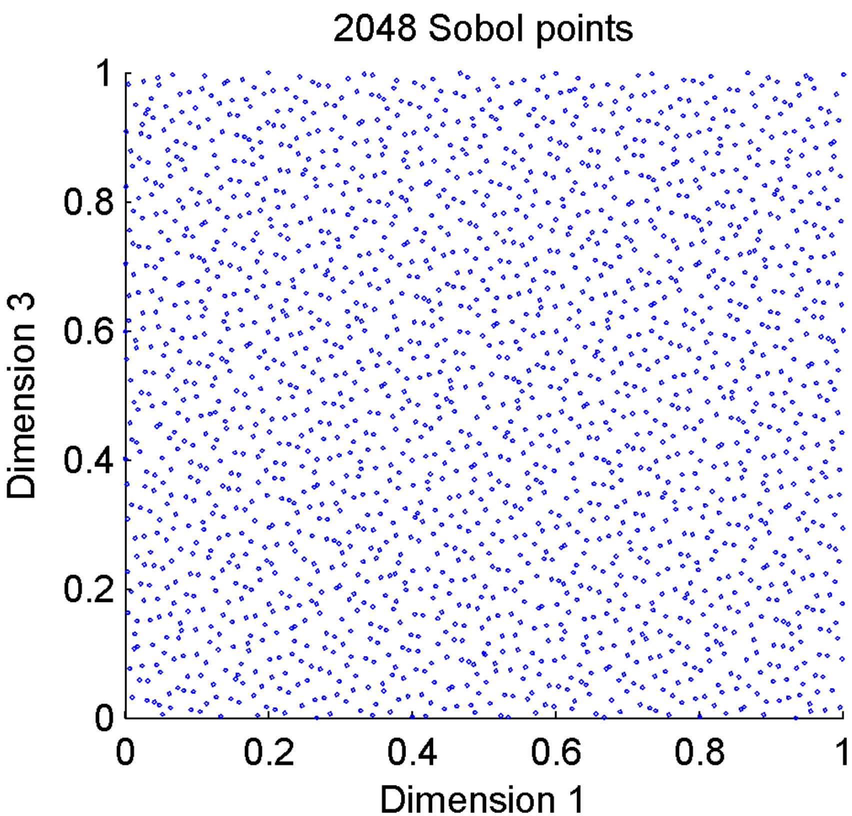
- We approximate $I = \int_D f(x)dx$ by $\hat{I}_N = \frac{1}{N}\sum_{i=1}^{N} f(X_i)$
- Monte Carlo
  - X_i^k are drawn from $U([0,1])$
  - Error is $O(N^{-1/2})$, independent of dimension *d*
- Quasi-Monte Carlo
  - Utilizes deterministic sequences to improve convergence
  - Error is $O(\log(N)^p / N)$, *N* grows exponentially with *d* (Morokoff, Caflisch 1994)
  - In high dimensions, low discrepancy sequences are no more uniform than random sequences and exhibit clustering

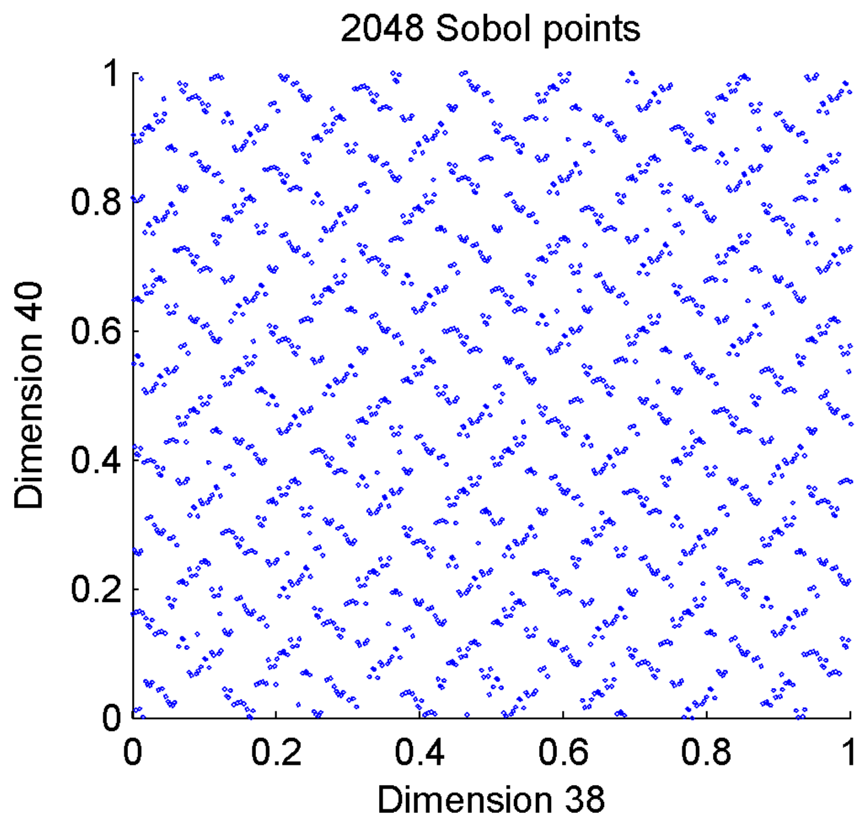
Pseudo-Random Numbers





Sobol Quasi-Random Numbers





Brownian Motion

- Let $X_i = \{X_i^0, X_i^1, \ldots, X_i^d\}$ be the discretized path
- Interest rates are modeled as Brownian motion

Distribution of *W(s)* given *W(r)* for *r<s:*

$$W(s) \sim N(W(r), s-r)$$

Distribution of *W(s)* given *W(r),W(t)* for *r<s<t*

$$W(s) \sim N((1-\beta)W(r) + \beta W(t), \beta(1-\beta)(t-r)) \quad \beta = \frac{s-r}{t-r}$$

- Discretized Brownian motion

$$X^j = X^i + \sqrt{(j-i)\Delta t} \cdot z \qquad z \sim N(0,1)$$

$$X^j = (1-\beta)X^i + \beta X^k + \sqrt{\beta(1-\beta)(k-i)\Delta t} \cdot z \quad \beta = \frac{j-i}{k-i}$$

Discretization Techniques

- Standard Discretization

  $X^0, X^1, X^2, ..., X^{d-1}, X^d$

  - Inexpensive to compute but requires exponential time to obtain good equidistribution on *d* dimensions
  - Substitution of parts of the path with pseudo-random numbers renders Quasi-MC ineffective

- Brownian Bridge Discretization

  $X^0, X^d, X^{d/2}, X^{d/4}, X^{3d/4}, X^{d/8}, X^{3d/8}, X^{5d/8}, X^{7d/8}, X^{d/16}...$

  - Concentrates most of path variance within first few components
  - Reduces effective dimension of the problem

Example: Importance of Ordering

- Vanilla European Call
  - Nominally, *d* dimensional integral
  - Value depends only on value of underlying stock at maturity
  - Optimal ordering would render X^d first

- Mortgage-Backed Securities
  - Early parts of the path are important
  - Declining PV of dollar & declining number of mortgages in the pool
  - Optimal ordering would render early parts of the path first

Weighted Brownian Bridge

- Standard Brownian Bridge 'utility' function

  $$U_{SBB}(t) = (t - t_a)(t - t_b)$$

- SBB discretization

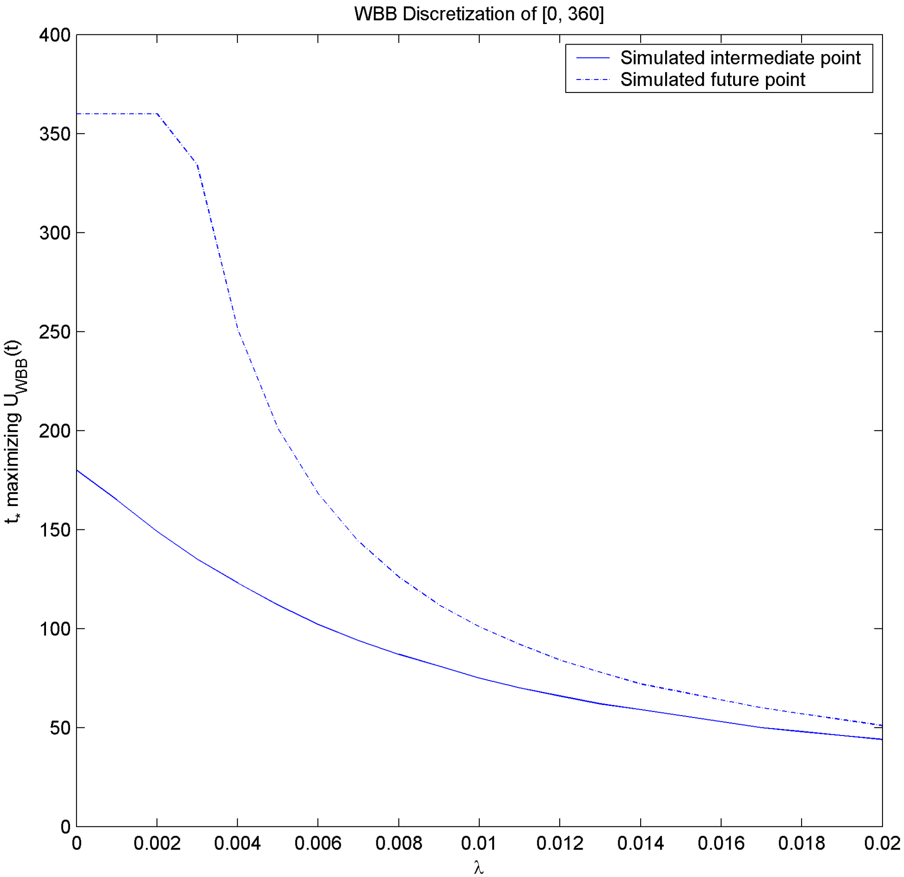
  $$X^0, X^{360}, X^{180}, X^{90}, X^{270}, X^{45}, \ldots$$

- Weighted Brownian Bridge 'utility' function

  $$U_{WBB}(t) = (t - t_a)(t - t_b)e^{-\lambda t}$$

- $WBB_{\lambda=0.01}$ discretization

  $$X^0, X^{101}, X^{21}, X^{174}, X^4, X^{38}, \ldots$$



Generalization of Path Discretization

- By varying λ can obtain other path discretization methods
  - Standard Discretization $\lambda \to \infty$
    Successive points maximize utility function
  - Standard Brownian Bridge Discretization $\lambda = 0$
    Midpoints maximize utility function

- How do we choose optimal λ ?
  - Choice of λ is integrand specific
  - Quantitative approach for estimating λ : ANOVA decomposition

ANOVA - ANalysis Of VAriance

- Using ANOVA decomposition:

$$f(x) = \sum_{u \subseteq \{1\ldots d\}} f_u(x)$$

$$f_u(x) = \int \left(f(x) - \sum_{v \subset u} f_v(x) \right) dx^{-u} = \int f(x) dx^{-u} - \sum_{v \subset u} f_v(x)$$

- Total variance decomposed into parts attributed to each subset

$$\sigma^2(f) = \sum_u \sigma^2(f_u)$$

- Numerically approximate f_u by integrating over x^{-u}

Dimension Distribution

- Effective dimension $\sum_{|u|<d_S} \sigma^2(f_u) \geq p \cdot \sigma^2(f)$
- Density function $v(s) = \sum_{|u|=s} \sigma^2(f_u) / \sigma^2(f)$
- Cumulative density function $F_v(s) = \sum_{t=1}^{s} v(t)$
  - Total variance of the function 'explained' by the first s dimensions
- Need a measure to describe the overall behavior of $F_v(s)$

Effective Dimension Order

- A function has effective dimension order r if

$$1 - F_v(s) \sim \alpha s^{-r}$$

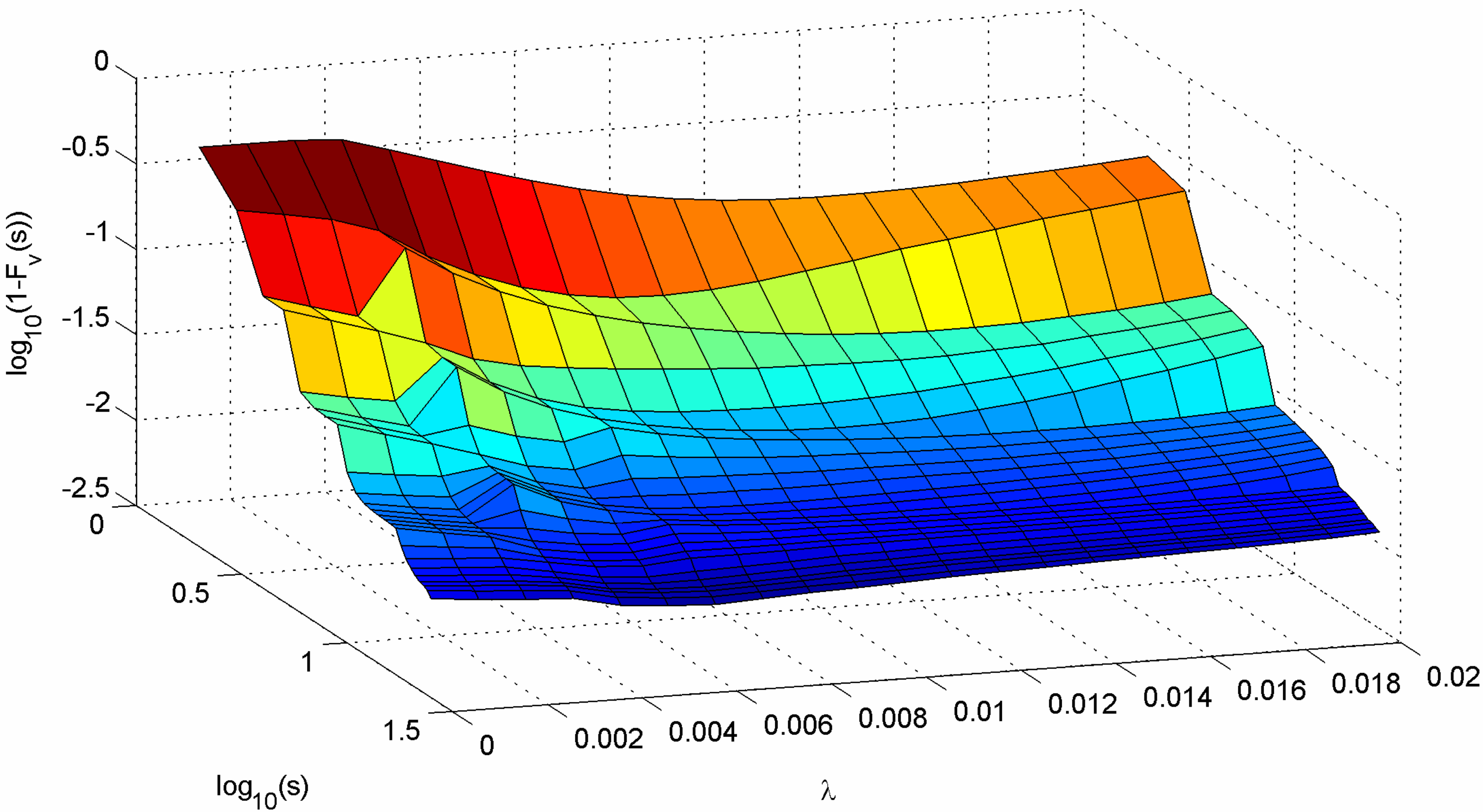
- Residual variability vs. s–Power Law

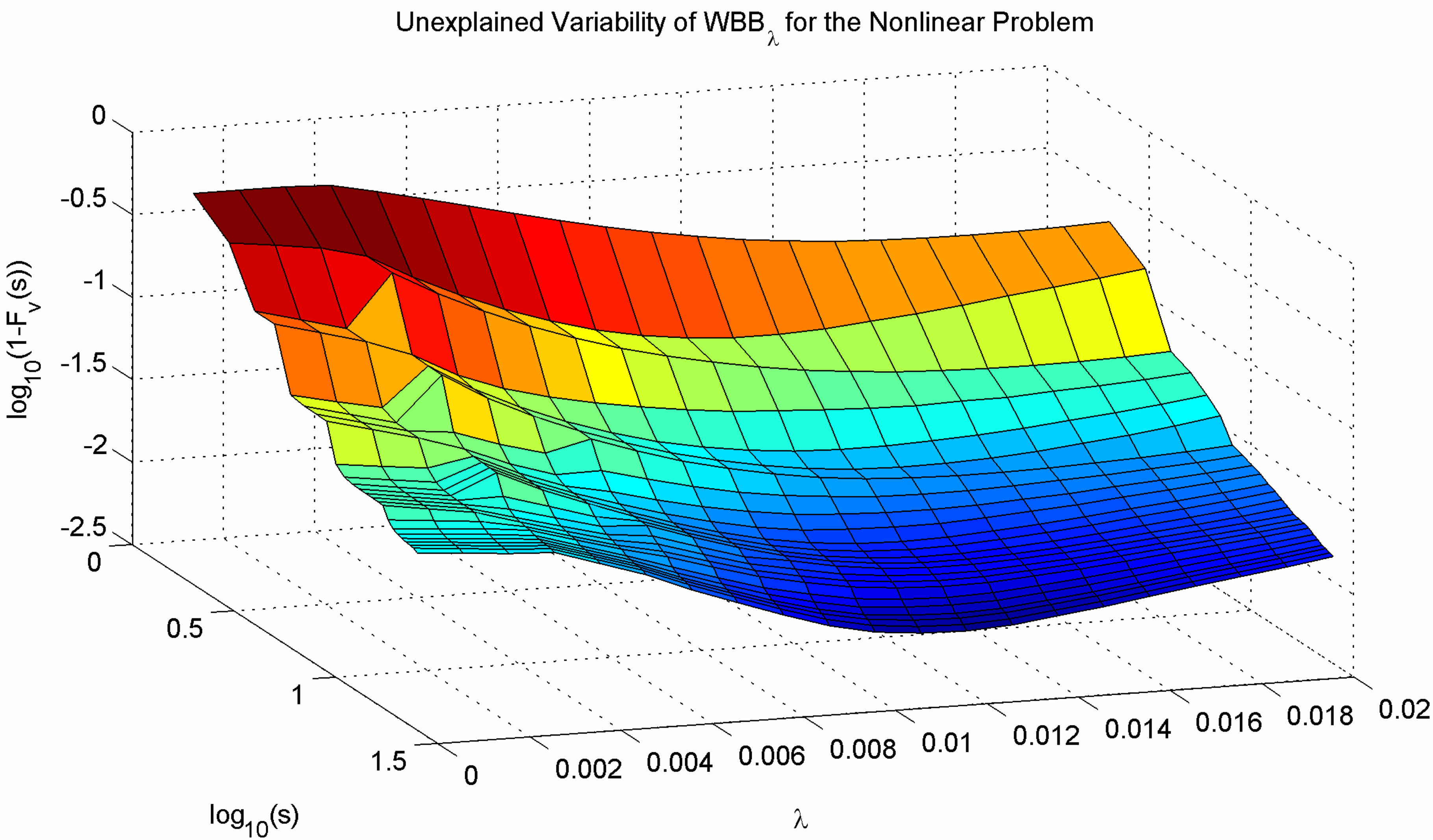
$$1 - F_v(s) \sim \alpha s^{-r} \Leftrightarrow \log(1 - F_v(s)) \sim \log \alpha - r \log s$$

  - Can estimate α, r using LLS
  - Constant term α is significant

- Exponential law $1 - F_v(s) \sim \alpha e^{-rs}$
  - Not supported by numerical experiments

Unexplained Variability of WBB_λ for the Nearly Linear Problem




Unexplained Variability of WBBλ for the Nonlinear Problem
log10(1-Fv(s))
0
-0.5
-1
-1.5
-2
-2.5
0
0.5
1
1.5
log10(s)
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0.02
λ

Effective Dimension Results

Effective Dimension of Nearly Linear Problem

	SBB	$WBB_{\lambda=0.005}$	$WBB_{\lambda=0.01}$	$WBB_{\lambda=0.015}$	$WBB_{\lambda=0.02}$
$ED_{p=0.99}$	25	21	23	25	25
$ED_{p=0.95}$	9	5	4	4	7
EDO	$0.71s^{-1.31}$	$0.36s^{-1.21}$	$0.17s^{-0.92}$	$0.17s^{-0.92}$	$0.26s^{-1.01}$

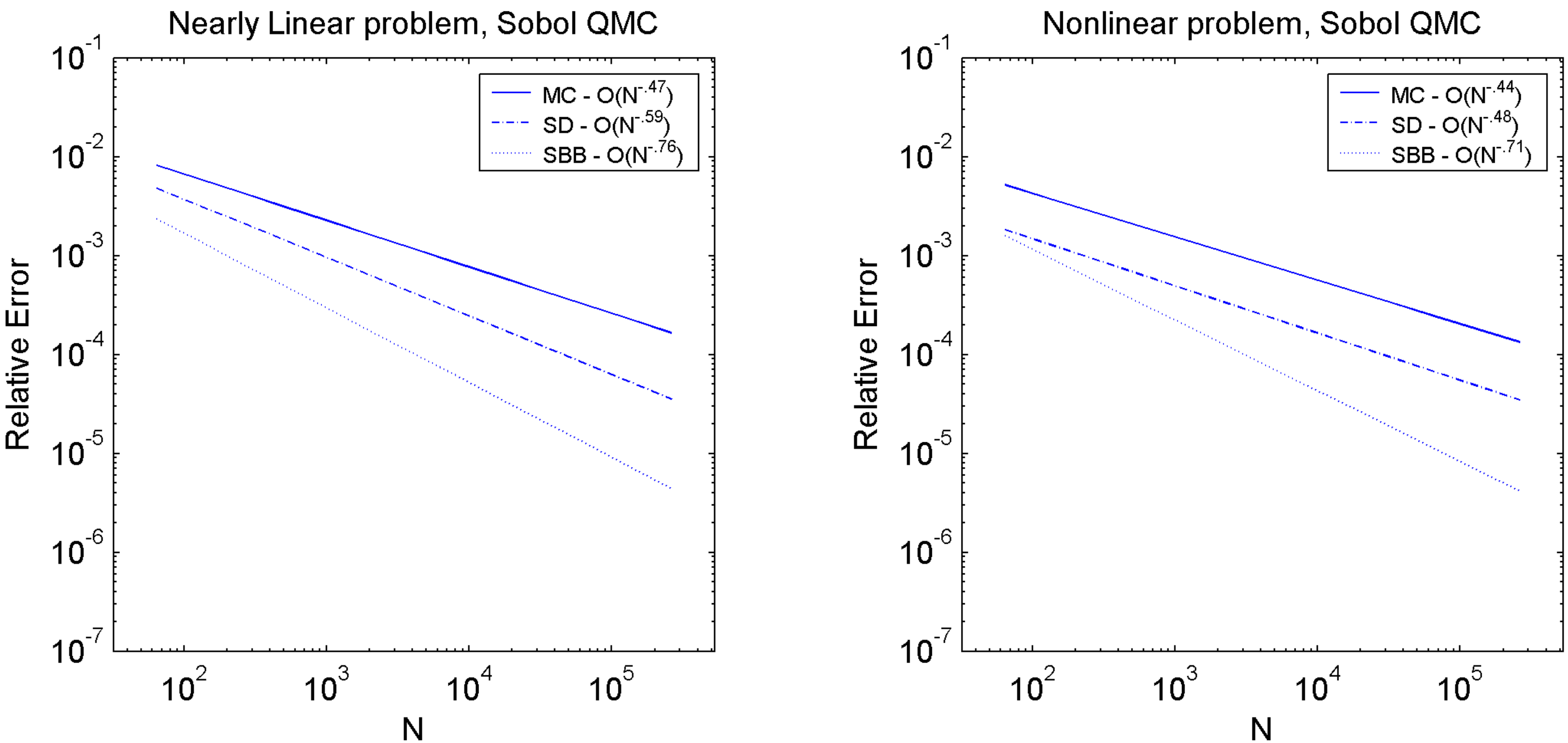
Effective Dimension of Non Linear Problem

	SBB	$WBB_{\lambda=0.005}$	$WBB_{\lambda=0.01}$	$WBB_{\lambda=0.015}$	$WBB_{\lambda=0.02}$
$ED_{p=0.99}$	25	25	20	21	25
$ED_{p=0.95}$	17	7	4	4	5
EDO	$0.73s^{-0.99}$	$0.40s^{-0.98}$	$0.21s^{-1.03}$	$0.16s^{-0.93}$	$0.17s^{-0.82}$

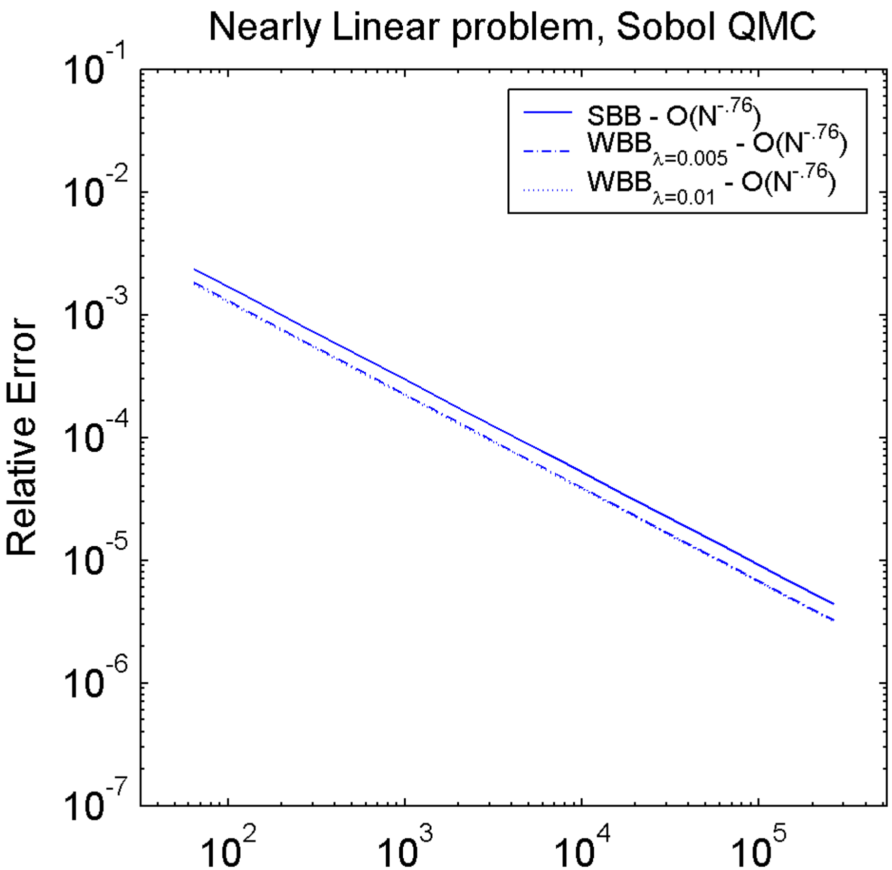
ANOVA Results

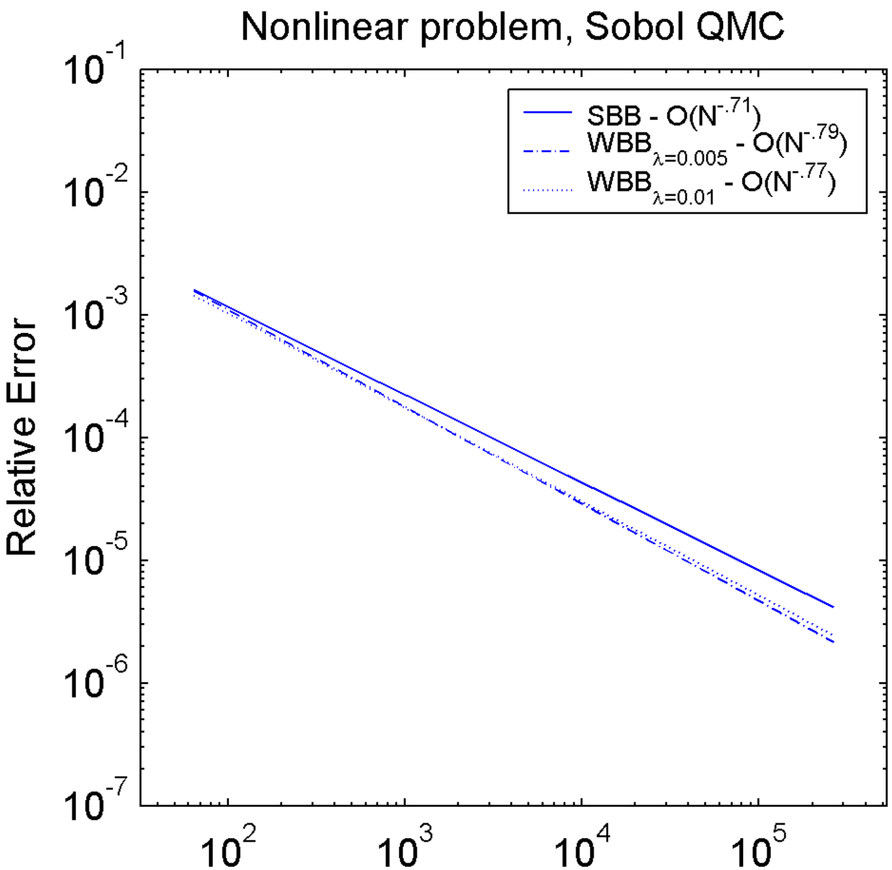
- ED varies greatly with the choice of *p*
- ED is not consistent
  - Caused by difference in slope of $\log(1 - F_v(s))$
- WBB has lower dimensionality than SBB
  - Especially effective on nonlinear problem
- Can infer ED from EDO $F_v(s) = p \Leftrightarrow s = (\frac{1-p}{\alpha})^{-\frac{1}{r}}$
- Relationship between dimension reduction and convergence of MC/QMC methods?

Numerical Results – MC, QMC



Numerical Results – SBB, WBB



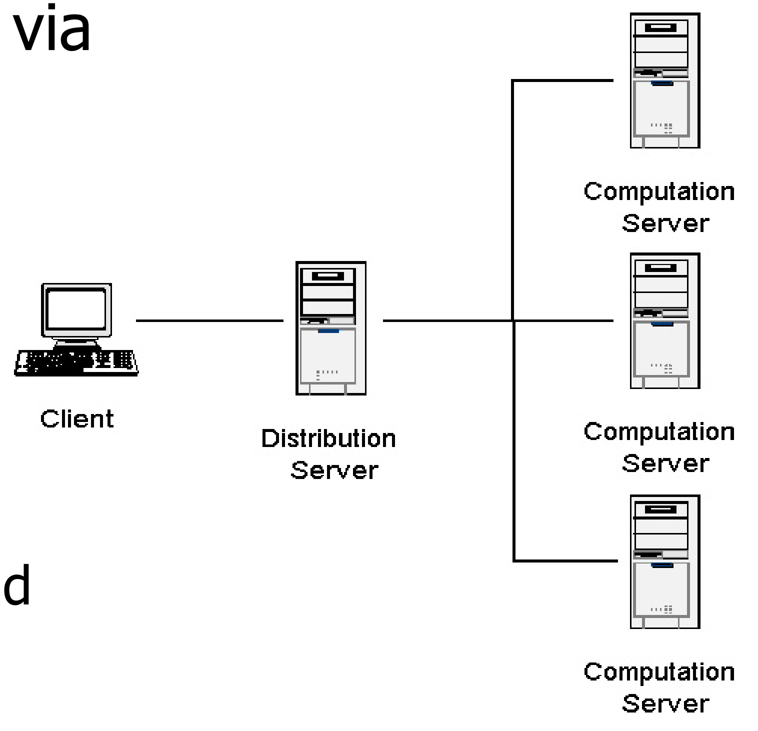


Summary of Numerical Results

- Monte Carlo method
  - MC error $\sim N^{-0.5}$
  - Quasi-MC error $\sim N^{-0.6}$
- Brownian Bridge discretization
  - SBB improves the order of convergence to $\sim N^{-0.7}$
  - WBB outperforms SBB for $\lambda \in [0.005, 0.015]$ with error $\sim N^{-0.8}$
  - Sobol sequence benefited most on linear problem vs Niederreiter sequence on nonlinear problem
- ANOVA Decomposition
  - Provides a framework for determining optimal parameter λ
  - Consistent with convergence results based on the parameter

Distributed Environment

- MC 'embarrassingly' parallel
- Master-Worker communication via Microsoft .NET Remoting
  - Ease of use
  - Transparency – proxy objects
  - Flexibility – channels
- Alternative HPC technologies
  - MPI, PVM, CORBA
  - Microsoft Application Center – load balancing and fault tolernace



Computation with MATLAB

- MATLAB rand for MC & NAG Library for QMC
- Supports Automation – can call MATLAB functions, code from C# .NET

```
MLApp.MLAppClass matlab = new MLApp.MLAppClass();
matlab.Execute("a = [1 2 3 4; 5 6 7 8]")
matlab.Execute("b = a + a")
matlab.GetFullMatrix("b", "base", ref real, ref imag);
```

- Leapfrog method for splitting the sequence
  - Processor k of a pool of M processors generates

$$X_k, X_{k+M}, X_{k+2M}, X_{k+3M}, X_{k+4M}, \ldots$$

Distributed Algorithm Speedup

- Timing results $T_{N,M}$

	N=1024	N=16384	N=32768	N=65536	N=1048576
M=1	1.584	4.374	7.248	13.201	188.547
M=2	0.942	2.316	3.677	6.619	94.413
M=4	0.776	1.487	2.114	3.439	47.418
M=8	0.869	1.156	1.572	2.330	24.690
M=16	0.860	0.819	0.992	1.380	12.721

- Efficiency $E_{N,M} = T_{N,1} / (T_{N,M} \cdot M)$

	N=1024	N=16384	N=32768	N=65536	N=1048576
M=1	1.000	1.000	1.000	1.000	1.000
M=2	0.841	0.944	0.986	0.997	0.999
M=4	0.510	0.735	0.857	0.960	0.994
M=8	0.228	0.473	0.576	0.708	0.955
M=16	0.115	0.334	0.457	0.598	0.926

Conclusion

- Weighted Brownian Bridge discretization
- Dimensionality Reduction using ANOVA analysis
- Parallel Implementation in .NET