Woudschoten Conference 2016

Introduction to quasi-Monte Carlo methods, with application to PDEs with random coefficients

Part 2

Application of QMC methods to PDEs with random coefficients

– a survey of analysis and implementation

Frances Kuo

f.kuo@unsw.edu.au

University of New South Wales, Sydney, Australia

∼ Based on a *J. FoCM* survey of the same title, written jointly with Dirk Nuyens (KU Leuven, Belgium) ∼

Outline

- Motivating example – a flow through a porous medium

- **Quasi-Monte Carlo (QMC) methods**
- **Component-by-component (CBC) construction**

- **Application of QMC theory to PDEs with random coefficients**
  - estimate the norm
  - choose the weights

- **Software**

  `http://people.cs.kuleuven.be/~dirk.nuyens/qmc4pde`

- Concluding remarks

Motivating example

Uncertainty in groundwater flow

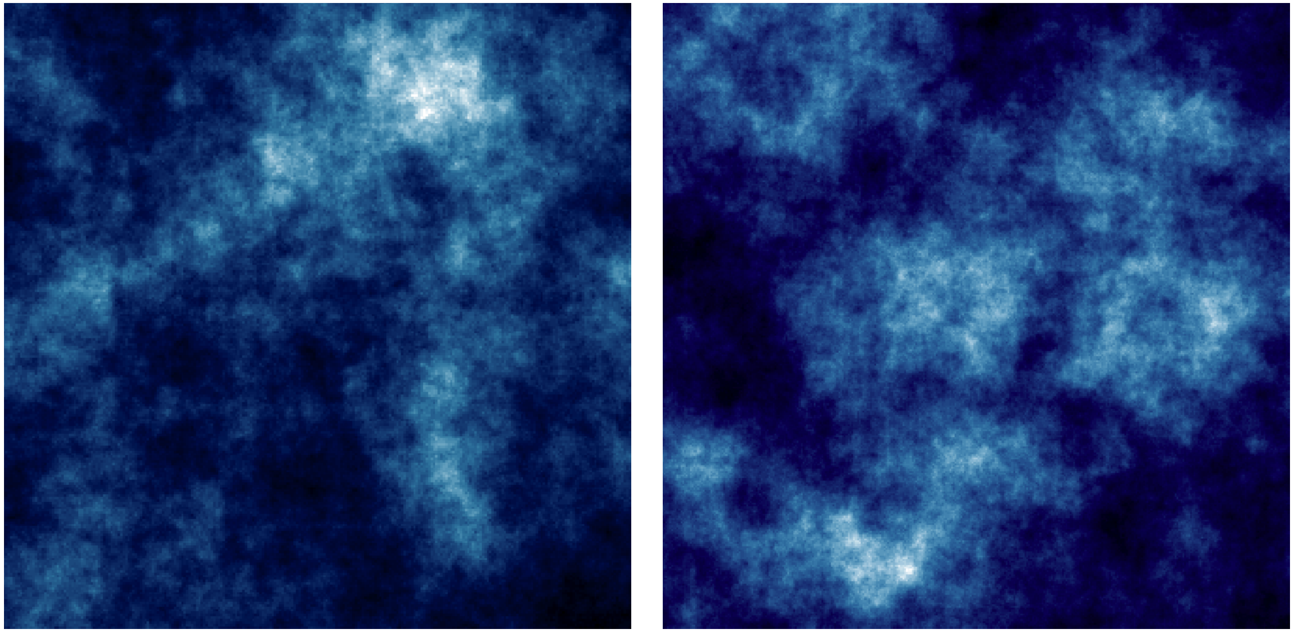
eg. risk analysis of radwaste disposal or CO_2 sequestration

Darcy's law $\quad \boldsymbol{q} + \boldsymbol{a}\,\vec{\boldsymbol{\nabla}}\boldsymbol{p} = \boldsymbol{\kappa}$

mass conservation law $\quad \boldsymbol{\nabla}\cdot\boldsymbol{q} = \boldsymbol{0}$

in $D \subset \mathbb{R}^d$, $d = 1, 2, 3$

together with boundary conditions



Uncertainty in $a(\boldsymbol{x},\omega)$ leads to uncertainty in $q(\boldsymbol{x},\omega)$ and $p(\boldsymbol{x},\omega)$

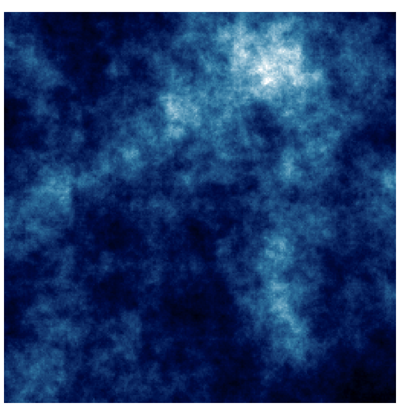
Expected values of quantities of interest

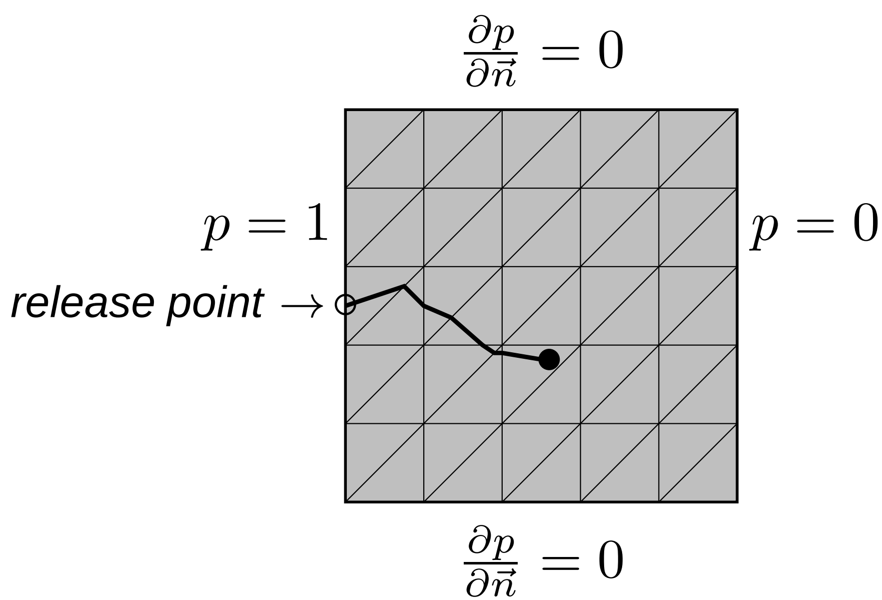
To compute the expected value of some quantity of interest:

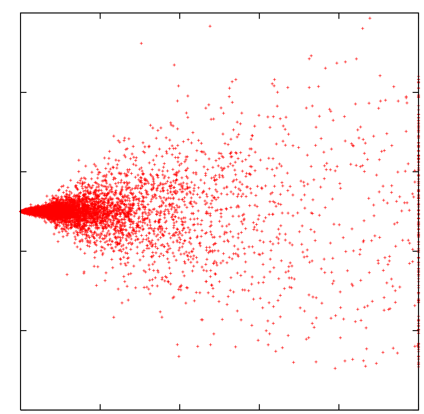
1. Generate a number of realizations of the random field
   (Some approximation may be required)
2. For each realization, solve the PDE using e.g. FEM / FVM / mFEM
3. Take the average of all solutions from different realizations

This describes Monte Carlo simulation.

Example: particle dispersion







Expected values of quantities of interest

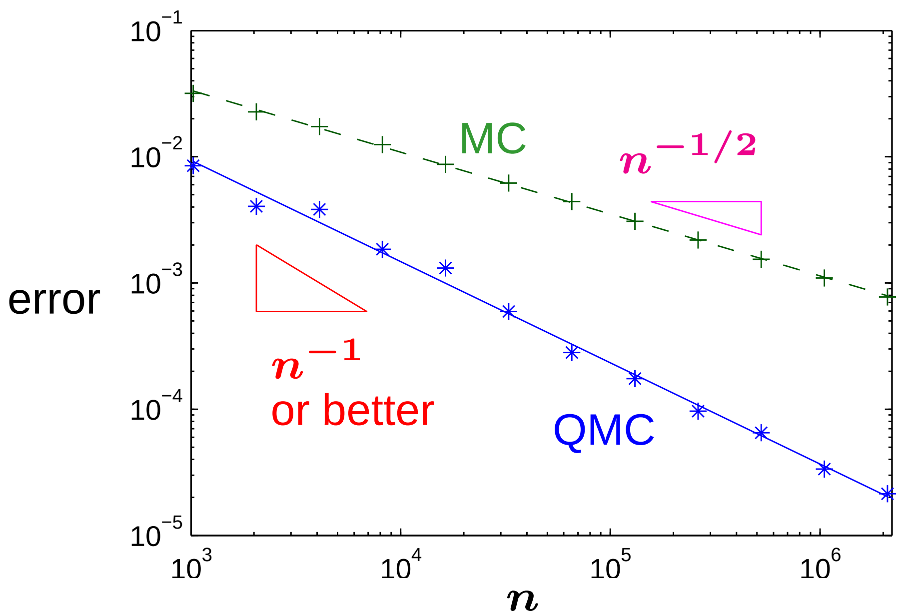
To compute the expected value of some quantity of interest:

1. Generate a number of realizations of the random field
   (Some approximation may be required)
2. For each realization, solve the PDE using e.g. FEM / FVM / mFEM
3. Take the average of all solutions from different realizations

This describes Monte Carlo simulation.

NOTE: **expected value** $=$ (high dimensional) **integral**

$\rightarrow$ use **quasi-Monte Carlo methods**



$s =$ stochastic dimension

MC v.s. QMC in the unit cube

$$\int_{[0,1]^s} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \;\approx\; \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{t}_i)$$

Monte Carlo method

$\boldsymbol{t}_i$ random uniform

$n^{-1/2}$ convergence

order of variables irrelevant

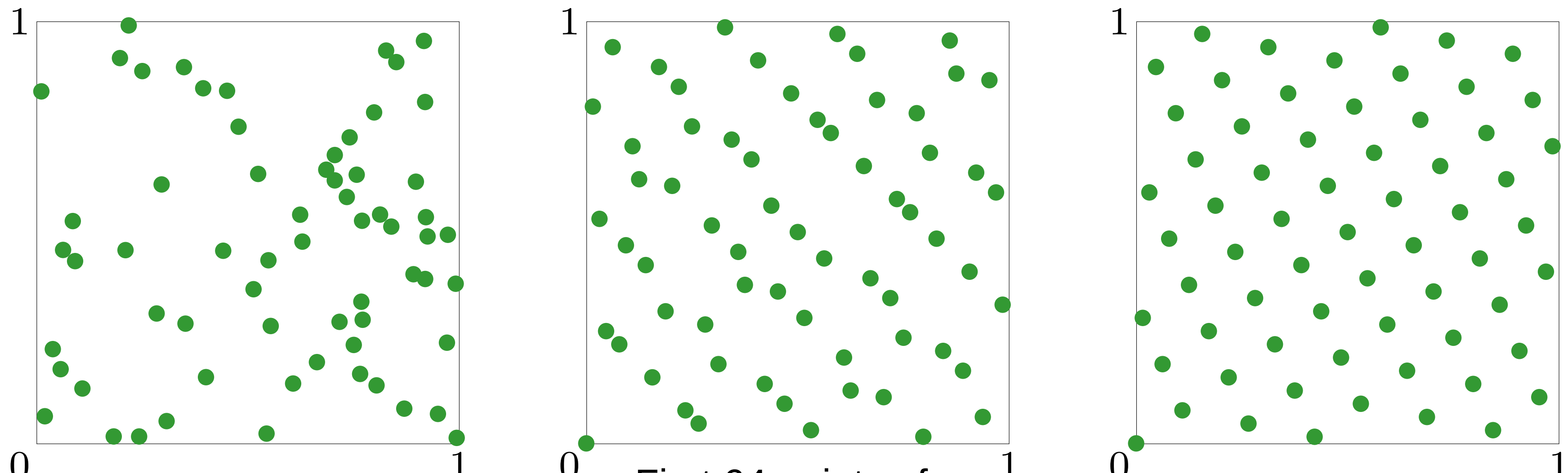
Quasi-Monte Carlo methods

$\boldsymbol{t}_i$ deterministic

close to n^{-1} convergence or better

more effective for earlier variables and lower-order projections

order of variables very important



64 random points

First 64 points of a 2D Sobol′ sequence

A lattice rule with 64 points

use *randomized* QMC methods for error estimation

QMC

Two main families of QMC methods:

- **(t,m,s)-nets** and **(t,s)-sequences**
- **lattice rules**

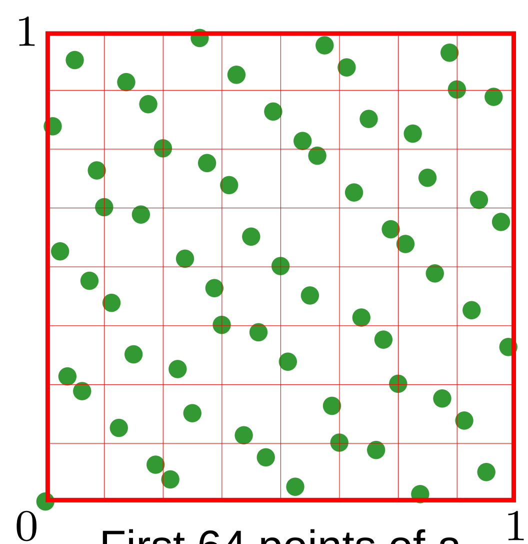
Niederreiter book (1992)
Sloan and Joe book (1994)

Dick and Pillichshammer book (2010)
Dick, K., Sloan Acta Numerica (2013)

Important developments:

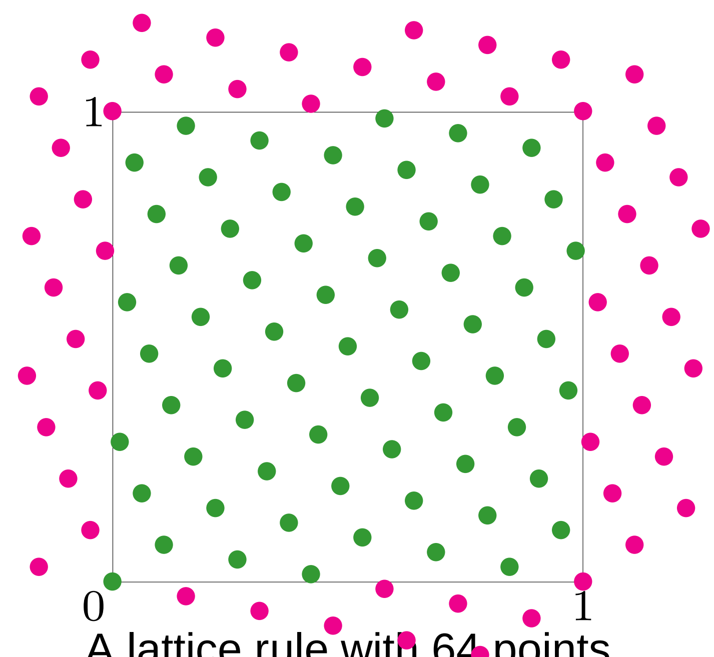
- ***component-by-component*** (***CBC***) ***construction, "fast" CBC***
- ***higher order digital nets***

Nuyens and Cools (2006)
Dick (2008)



First 64 points of a 2D Sobol′ sequence
(0,6,2)-net

Having the right number of points in various sub-cubes



A lattice rule with 64 points

A group under addition modulo $\mathbb{Z}$ and includes the integer points

Application of QMC to PDEs with random coefficients

[0] Graham, K., Nuyens, Scheichl, Sloan (J. Comput. Physics, 2011)
[1] K., Schwab, Sloan (SIAM J. Numer. Anal., 2012)
[2] K., Schwab, Sloan (J. FoCM, 2015)
[3] Graham, K., Nichols, Scheichl, Schwab, Sloan (Numer. Math., 2015)
[4] K., Scheichl, Schwab, Sloan, Ullmann (Math. Comp., to appear)
[5] Dick, K., Le Gia, Nuyens, Schwab (SIAM J. Numer. Anal., 2014)
[6] Dick, K., Le Gia, Schwab (SIAM J. Numer. Anal., 2016)
[7] Graham, K., Nuyens, Scheichl, Sloan (in progress)

Also

- Schwab (Proceedings of MCQMC 2012)
- Le Gia (Proceedings of MCQMC 2012)
- Dick, Le Gia, Schwab (in review, in review, in progress)
- Harbrecht, Peters, Siebenmorgen (Math. Comp., 2015)
- Gantner, Schwab (Proceedings of MCQMC 2014)
- Ganesh, Hawkings (SIAM J. Sci. Comput., 2015)
- Robbe, Nuyens, Vandewalle (in review)
- Scheichl, Stuart, Teckentrup (in review)
- Gilbert, Graham, K., Scheichl, Sloan (in progress)
- Graham, Parkinson, Scheichl (in progress)

etc.

Application of QMC to PDEs with random coefficients

Three different QMC theoretical settings:

[1,2] Weighted Sobolev space over $[0,1]^s$ and "randomly shifted lattice rules"

[3,4] Weighted space setting in $\mathbb{R}^s$ and "randomly shifted lattice rules"

[5,6] Weighted space of smooth functions over $[0,1]^s$ and (deterministic) "interlaced polynomial lattice rules"

- K., Nuyens (J. FoCM) – a survey of [1,2], [3,4], [5,6] in a unified view
- Accompanying software package
  `http://people.cs.kuleuven.be/~dirk.nuyens/qmc4pde/`

	Uniform	Lognormal KL expansion	Lognormal Circulant embedding
Experiments only			[0]*
First order, single-level	[1]	[3]*	[7]*
First order, multi-level	[2]	[4]*	
Higher order, single-level	[5]		
Higher order, multi-level	[6]*		

The * indicates there are accompanying numerical results

5-minute QMC primer (most important slide)

- Common features among all three QMC theoretical settings:
  - Separation in error bound

    (rms) **QMC error** $\leq$ (rms) **worst case error**$_\gamma$ $\times$ **norm of integrand**$_\gamma$

  - Weighted spaces
  - Pairing of QMC rule with function space
  - Optimal rate of convergence
  - Rate and constant independent of dimension
  - **Fast component-by-component (CBC) construction**
  - Extensible or embedded rules
- Application of QMC theory
  - **Estimate the norm** (critical step)
  - **Choose the weights**
  - Weights as input to the CBC construction

Lattice rules

Rank-1 lattice rules have points

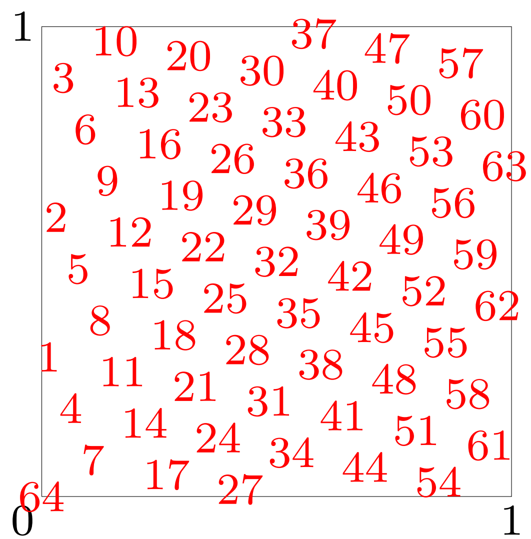
$$\boxed{\boldsymbol{t}_i = \mathrm{frac}\left(\frac{i}{n}\,\boldsymbol{z}\right), \qquad i = 1, 2, \ldots, n}$$

$\boldsymbol{z} \in \mathbb{Z}^s$ – the generating vector, with all components *coprime* to n

$\mathrm{frac}(\cdot)$ – means to take the fractional part of all components

$\sim$ quality determined by the choice of $\boldsymbol{z}$ $\sim$

$$n = 64 \quad \boldsymbol{z} = (1, 19) \quad \boldsymbol{t}_i = \mathrm{frac}\left(\frac{i}{64}(1, 19)\right)$$



A lattice rule with 64 points

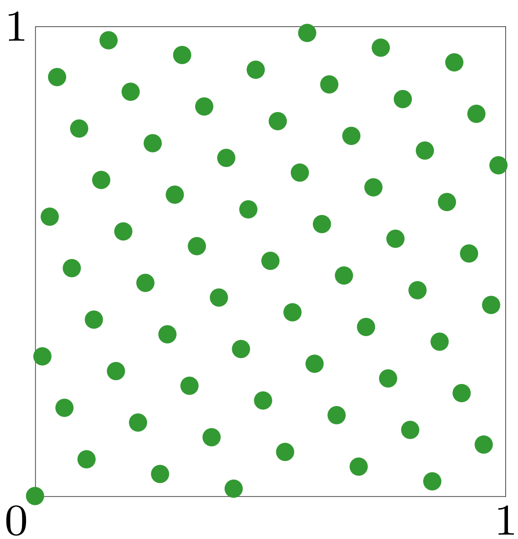
Randomly shifted lattice rules

Shifted rank-1 lattice rules have points

$$\boxed{\boldsymbol{t}_i = \mathrm{frac}\left(\frac{i}{n}\boldsymbol{z} + \boldsymbol{\Delta}\right), \qquad i = 1, 2, \ldots, n}$$

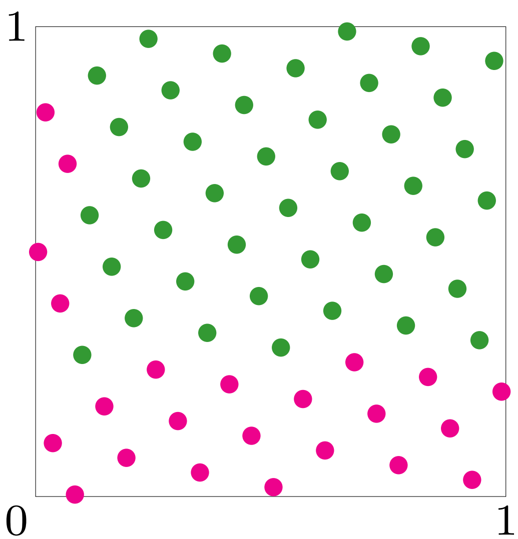
$\boldsymbol{\Delta} \in [0,1)^s$ – the shift

$\sim$ use a number of random shifts for error estimation $\sim$



A lattice rule with 64 points

shifted by $\boldsymbol{\Delta} = (0.1, 0.3)$



A shifted lattice rule with 64 points

Component-by-component construction

- Want to find $\boldsymbol{z}$ for which some error criterion is as small possible

  $\sim$ Exhaustive search is practically impossible - too many choices! $\sim$

- **CBC algorithm** [Korobov (1959); Sloan, Reztsov (2002); Sloan, K., Joe (2002),. . .]
  1. Set $z_1 = 1$.
  2. With z_1 fixed, choose z_2 to minimize the error criterion in 2D.
  3. With z_1, z_2 fixed, choose z_3 to minimize the error criterion in 3D.
  4. etc.

[K. (2003); Dick (2004)]

- Optimal rate of convergence $\mathcal{O}(n^{-1+\delta})$ in "weighted Sobolev space", independently of s under an appropriate condition on the weights

  $\sim$ Averaging argument: there is always one choice as good as average! $\sim$

[Nuyens, Cools (2006)]

- Cost of algorithm for "product weights" is $\mathcal{O}(n \log n\, s)$ using FFT

[Hickernell, Hong, L'Ecuyer, Lemieux (2000); Hickernell, Niederreiter (2003)]

- Extensible/embedded variants [Cools, K., Nuyens (2006)]

[Dick, Pillichshammer, Waterhouse (2007)]

---

http://people.cs.kuleuven.be/~dirk.nuyens/fast-cbc/

http://people.cs.kuleuven.be/~dirk.nuyens/qmc-generators/

Setting 1: standard QMC for unit cube

- **Worst case error bound**

$$\left| \int_{[0,1]^s} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} - \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{t}_i) \right| \le e_{\boldsymbol{\gamma}}^{\mathrm{wor}}(\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n)\,\|f\|_{\boldsymbol{\gamma}}$$

- **Weighted Sobolev space** [Sloan, Woźniakowski (1998)]

$$\|f\|_{\boldsymbol{\gamma}}^2 = \underbrace{\sum_{\mathfrak{u}\subseteq\{1:s\}}}_{2^s \text{ subsets}} \frac{1}{\underbrace{\gamma_{\mathfrak{u}}}_{\text{“weights”}}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|} f}{\partial \boldsymbol{y}_{\mathfrak{u}}}(\boldsymbol{y}_{\mathfrak{u}};\underbrace{0}_{\text{“anchor” at 0}}) \right|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}$$

2^s subsets

“weights”

“anchor” at 0 (also “unanchored”)

Mixed first derivatives are square integrable

Small weight $\gamma_{\mathfrak{u}}$ means that f depends weakly on the variables $\boldsymbol{y}_{\mathfrak{u}}$

- **Pair with randomly shifted lattice rules**

- **Choose weights to minimize the error bound** [K., Schwab, Sloan (2012)]

$$\underbrace{\left(\frac{2}{n}\sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} a_{\mathfrak{u}}\right)^{1/(2\lambda)}}_{\text{bound on worst case error (CBC)}} \underbrace{\left(\sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{b_{\mathfrak{u}}}{\gamma_{\mathfrak{u}}}\right)^{1/2}}_{\text{bound on norm}} \Rightarrow \gamma_{\mathfrak{u}} = \left(\frac{b_{\mathfrak{u}}}{a_{\mathfrak{u}}}\right)^{1/(1+\lambda)}$$

- **Construct points (CBC) to minimize the worst case error**

Setting 2: QMC integration over $\mathbb{R}^s$

- **Change of variables**

- **Norm with weight function**

- **Pair with randomly shifted lattice rules**

Setting 3: smooth integrands in the cube

- **Norm involving higher order mixed derivatives**
- **Pair with higher order digital nets**
  - Interlaced polynomial lattice rule

Uniform v.s. lognormal

$$-\nabla \cdot (a(\boldsymbol{x},\boldsymbol{y})\,\nabla u(\boldsymbol{x},\boldsymbol{y})) = \kappa(\boldsymbol{x}), \quad \boldsymbol{x} \in D \subset \mathbb{R}^d,\ d = 1,2,3$$
$$u(\boldsymbol{x},\boldsymbol{y}) = 0, \qquad \boldsymbol{x} \in \partial D$$

Uniform [Cohen, DeVore, Schwab (2011)]

- $a(\boldsymbol{x},\boldsymbol{y}) = \overline{a}(\boldsymbol{x}) + \sum_{j\ge1} y_j\,\psi_j(\boldsymbol{x}), \quad y_j \sim \mathcal{U}[-\frac{1}{2},\frac{1}{2}]$
- Goal: $\boxed{\int_{(-\frac{1}{2},\frac{1}{2})^{\mathbb{N}}} G(u(\cdot,\boldsymbol{y}))\,\mathrm{d}\boldsymbol{y}}$ G – bounded linear functional

Lognormal

- $a(\boldsymbol{x},\boldsymbol{y}) = \exp(Z(\boldsymbol{x},\boldsymbol{y}))$
- **Karhunen-Loève expansion**:

  $Z(\boldsymbol{x},\boldsymbol{y}) = \overline{Z}(\boldsymbol{x}) + \sum_{j\ge1} y_j\,\sqrt{\mu_j}\,\xi_j(\boldsymbol{x}), \quad y_j \sim \mathcal{N}(0,1)$
- Goal: $\boxed{\int_{\mathbb{R}^{\mathbb{N}}_*} G(u(\cdot,\boldsymbol{y}))\,\prod_{j\ge1}\phi_{\mathrm{nor}}(y_j)\,\mathrm{d}\boldsymbol{y}}$

Single-level v.s. multi-level (the uniform case)

$$I_\infty = \int_{(-\frac{1}{2},\frac{1}{2})^{\mathbb{N}}} G(u(\cdot,\boldsymbol{y}))\,\mathrm{d}\boldsymbol{y}$$

(1) dimension truncation
(2) FE discretization
(3) QMC quadrature

Single-level algorithm

- Deterministic: $\displaystyle \frac{1}{n}\sum_{i=1}^{n} G(u_h^s(\cdot,\boldsymbol{t}_i - \tfrac{\boldsymbol{1}}{\boldsymbol{2}}))$
- Randomized:

Multi-level algorithm

$I_\infty = (I_\infty - I_L) + \sum_{\ell=0}^{L}(I_\ell - I_{\ell-1}),\quad I_{-1} := 0$

[Heinrich (1998); Giles (2008)]

- Deterministic: $\displaystyle \sum_{\ell=0}^{L}\left(\frac{1}{n_\ell}\sum_{i=1}^{n_\ell} G((u_{h_\ell}^{s_\ell} - u_{h_{\ell-1}}^{s_{\ell-1}})(\cdot,\boldsymbol{t}_{\ell,i} - \tfrac{\boldsymbol{1}}{\boldsymbol{2}}))\right)$
- Randomized:

*Benefits of randomization: unbiased estimator, practical error estimate

Single-level v.s. multi-level (the lognormal case)

$$I_\infty = \int_{\mathbb{R}_*^{\mathbb{N}}} G(u(\cdot,\boldsymbol{y})) \prod_{j\ge 1} \phi_{\rm nor}(y_j)\, \mathrm{d}\boldsymbol{y}$$

(1) dimension truncation
(2) FE discretization
(3) QMC quadrature

Single-level algorithm

- Deterministic: $\displaystyle \frac{1}{n}\sum_{i=1}^{n} G(u_h^s(\cdot, \Phi^{-1}(\boldsymbol{t}_i)))$
- Randomized:

Multi-level algorithm

- Deterministic: $\displaystyle \sum_{\ell=0}^{L}\left(\frac{1}{n_\ell}\sum_{i=1}^{n_\ell} G((u_{h_\ell}^{s_\ell} - u_{h_{\ell-1}}^{s_{\ell-1}})(\cdot, \Phi^{-1}(\boldsymbol{t}_{\ell,i})))\right)$
- Randomized:

*Benefits of randomization: unbiased estimator, practical error estimate

Estimate the norm

Single-level algorithm

$$\|G(u_h^s)\|_{\boldsymbol{\gamma}}$$

Multi-level algorithm

$$\|G(u_{h_\ell}^{s_\ell} - u_{h_{\ell-1}}^{s_{\ell-1}})\|_{\boldsymbol{\gamma}}$$

TUTORIAL: Differentiate the PDE (the uniform case)

1. Variational formulation

$$\int_D a(\boldsymbol{x},\boldsymbol{y})\,\nabla u(\boldsymbol{x},\boldsymbol{y})\cdot\nabla w(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_D \kappa(\boldsymbol{x})\,w(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \forall w\in H_0^1(D)$$

2. Mixed derivatives in $\boldsymbol{y}$ with multi-index $\boldsymbol{\nu}\neq\boldsymbol{0}$

$$\int_D \partial^{\boldsymbol{\nu}}\Big(a(\boldsymbol{x},\boldsymbol{y})\,\nabla u(\boldsymbol{x},\boldsymbol{y})\cdot\nabla w(\boldsymbol{x})\Big)\,\mathrm{d}\boldsymbol{x} = 0 \quad \forall w\in H_0^1(D)$$

3. Leibniz product rule

$$\int_D \bigg(\sum_{\boldsymbol{m}\le\boldsymbol{\nu}}\binom{\boldsymbol{\nu}}{\boldsymbol{m}}(\partial^{\boldsymbol{m}}a)(\boldsymbol{x},\boldsymbol{y})\,\nabla(\partial^{\boldsymbol{\nu}-\boldsymbol{m}}u)(\boldsymbol{x},\boldsymbol{y})\cdot\nabla w(\boldsymbol{x})\bigg)\,\mathrm{d}\boldsymbol{x} = 0 \quad \forall w\in H_0^1(D)$$

4. Uniform case

$$a(\boldsymbol{x},\boldsymbol{y}) = \overline{a}(\boldsymbol{x}) + \sum_{j\ge1} y_j\,\psi_j(\boldsymbol{x}) \implies (\partial^{\boldsymbol{m}}a)(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} a(\boldsymbol{x},\boldsymbol{y}) & \text{if } \boldsymbol{m}=\boldsymbol{0},\\ \psi_j(\boldsymbol{x}) & \text{if } \boldsymbol{m}=\boldsymbol{e}_j,\\ 0 & \text{otherwise}\end{cases}$$

5. Separate out the $\boldsymbol{m}=\boldsymbol{0}$ term

$$\int_D a(\boldsymbol{x},\boldsymbol{y})\,\nabla(\partial^{\boldsymbol{\nu}}u)(\boldsymbol{x},\boldsymbol{y})\cdot\nabla w(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

$$= -\sum_{j\ge1}\nu_j\int_D \psi_j(\boldsymbol{x})\,\nabla(\partial^{\boldsymbol{\nu}-\boldsymbol{e}_j}u)(\boldsymbol{x},\boldsymbol{y})\cdot\nabla w(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \forall w\in H_0^1(D)$$

TUTORIAL: Differentiate the PDE (the uniform case)

5. Separate out the $\boldsymbol{m} = \mathbf{0}$ term

$$\int_D a(\boldsymbol{x},\boldsymbol{y})\, \nabla(\partial^{\boldsymbol{\nu}} u)(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla w(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$$

$$= -\sum_{j\geq 1} \nu_j \int_D \psi_j(\boldsymbol{x})\, \nabla(\partial^{\boldsymbol{\nu}-\boldsymbol{e}_j} u)(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla w(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \quad \forall w \in H_0^1(D)$$

6. Take $w = (\partial^{\boldsymbol{\nu}} u)(\cdot,\boldsymbol{y})$ and apply Cauchy-Schwarz

$$a_{\min} \int_D |\nabla(\partial^{\boldsymbol{\nu}} u)(\boldsymbol{x},\boldsymbol{y})|^2\, \mathrm{d}\boldsymbol{x}$$

$$\leq \sum_{j\geq 1} \nu_j\, \|\psi_j\|_{L_\infty} \left(\int_D |\nabla(\partial^{\boldsymbol{\nu}-\boldsymbol{e}_j} u)(\boldsymbol{x},\boldsymbol{y})|^2\, \mathrm{d}\boldsymbol{x}\right)^{1/2} \left(\int_D |\nabla(\partial^{\boldsymbol{\nu}} u)(\boldsymbol{x},\boldsymbol{y})|^2\, \mathrm{d}\boldsymbol{x}\right)^{1/2}$$

7. Cancel one common factor

$$\|\nabla(\partial^{\boldsymbol{\nu}} u)(\cdot,\boldsymbol{y})\|_{L_2} \leq \sum_{j\geq 1} \nu_j\, b_j\, \|\nabla(\partial^{\boldsymbol{\nu}-\boldsymbol{e}_j} u)(\cdot,\boldsymbol{y})\|_{L_2}, \qquad b_j = \frac{\|\psi_j\|_{L_\infty}}{a_{\min}}$$

8. Induction hypothesis

$$\|\nabla(\partial^{\boldsymbol{\nu}} u)(\cdot,\boldsymbol{y})\|_{L_2} \leq \sum_{j\geq 1} \nu_j\, b_j\, |\boldsymbol{\nu}-\boldsymbol{e}_j|!\, \boldsymbol{b}^{\boldsymbol{\nu}-\boldsymbol{e}_j} \frac{\|\kappa\|_{H^{-1}}}{a_{\min}} = |\boldsymbol{\nu}|!\, \boldsymbol{b}^{\boldsymbol{\nu}} \frac{\|\kappa\|_{H^{-1}}}{a_{\min}}$$

TUTORIAL: Estimate the norm (the uniform case)

8. Bound on mixed derivatives by induction

$$\|(\partial^{\boldsymbol{\nu}} u_h^s)(\cdot,\boldsymbol{y})\|_{H_0^1} = \|\nabla(\partial^{\boldsymbol{\nu}} u_h^s)(\cdot,\boldsymbol{y})\|_{L_2} \le |\boldsymbol{\nu}|!\, \boldsymbol{b}^{\boldsymbol{\nu}} \frac{\|\kappa\|_{H^{-1}}}{a_{\min}}, \quad b_j := \frac{\|\psi_j\|_{L_\infty}}{a_{\min}}$$

9. First order mixed derivatives

$$\left\| \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} u_h^s(\cdot,\boldsymbol{y}) \right\|_{H_0^1} \le |\mathfrak{u}|! \left(\prod_{j\in\mathfrak{u}} b_j \right) \frac{\|\kappa\|_{H^{-1}}}{a_{\min}}, \quad \mathfrak{u} \subseteq \{1:s\}$$

10. Linearity and boundedness of G

$$\left| \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} G(u_h^s(\cdot,\boldsymbol{y})) \right| = \left| G\left(\frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} u_h^s(\cdot,\boldsymbol{y}) \right) \right| \le \|G\|_{H^{-1}} \left\| \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} u_h^s(\cdot,\boldsymbol{y}) \right\|_{H_0^1}$$

11. Weighted norm

$$\|G(u_h^s)\|_{\boldsymbol{\gamma}} \le \frac{\|\kappa\|_{H^{-1}} \|G\|_{H^{-1}}}{a_{\min}} \left(\sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{(|\mathfrak{u}|!)^2 \prod_{j\in\mathfrak{u}} b_j^2}{\gamma_{\mathfrak{u}}} \right)^{1/2}$$

TUTORIAL: Choose the weights (the uniform case)

11. Weighted norm

$$\|G(u_h^s)\|_{\boldsymbol{\gamma}} \lesssim \left(\sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{(|\mathfrak{u}|!)^2 \prod_{j\in\mathfrak{u}} b_j^2}{\gamma_{\mathfrak{u}}} \right)^{1/2}, \quad b_j := \frac{\|\psi_j\|_{L_\infty}}{a_{\min}}$$

12. CBC error bound $\times$ bound on norm

$$\text{ms-error} \lesssim \left(\frac{2}{n} \sum_{\mathfrak{u}\subseteq\{1:s\}} {\gamma_{\mathfrak{u}}}^{\lambda} \, [\rho(\lambda)]^{|\mathfrak{u}|} \right)^{1/\lambda} \left(\sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{(|\mathfrak{u}|!)^2 \prod_{j\in\mathfrak{u}} b_j^2}{\gamma_{\mathfrak{u}}} \right) \quad \forall \lambda \in (\tfrac{1}{2}, 1]$$

13. Minimize the bound to yield **POD weights**

$$\gamma_{\mathfrak{u}} = \left(|\mathfrak{u}|! \prod_{j\in\mathfrak{u}} \frac{b_j}{\sqrt{\rho(\lambda)}} \right)^{2/(1+\lambda)}$$

14. Choose λ to ensure dimension independence

- If $\sum_{j\ge1} b_j^p < \infty$, take $\lambda = \begin{cases} \dfrac{1}{2 - \dfrac{2\delta}{p}} \text{ for } \delta \in (0, \frac{1}{2}) & \text{when } p \in (0, \frac{2}{3}], \\[2ex] \dfrac{p}{2-p} & \text{when } p \in (\frac{2}{3}, 1). \end{cases}$
- Convergence rate is $n^{-\min(\frac{1}{p}-\frac{1}{2},1-\delta)}$
- Implied constant is independent of s

Estimate the norm (the uniform case)

Single-level algorithm

$\|G(u_h^s)\|_{\gamma}$

$$b_j := \frac{\|\psi_j\|_{L_\infty}}{a_{\min}}$$

- **Lemma 1**

$$\|\nabla \partial^{\boldsymbol{\nu}} u(\cdot,\boldsymbol{y})\|_{L_2} \le |\boldsymbol{\nu}|!\, \boldsymbol{b}^{\boldsymbol{\nu}} \frac{\|\kappa\|_{H^{-1}}}{a_{\min}}$$

Multi-level algorithm

$\|G(u_{h_\ell}^{s_\ell} - u_{h_{\ell-1}}^{s_{\ell-1}})\|_{\gamma}$

$$\bar{b}_j := \frac{\max(\|\psi_j\|_{L_\infty}, \|\nabla\psi_j\|_{L_\infty})}{a_{\min}}$$

- **Lemma 2**

$\|\Delta \partial^{\boldsymbol{\nu}} u(\cdot,\boldsymbol{y})\|_{L_2}$

$\boxed{\text{Bounds of the form } (|\boldsymbol{\nu}|+a_1)!\,\bar{\boldsymbol{b}}^{\boldsymbol{\nu}}}$

- **Lemma 3**

$\|\nabla \partial^{\boldsymbol{\nu}} (u - u_h)(\cdot,\boldsymbol{y})\|_{L_2}$ $\mathcal{O}(h)$

- **Lemma 4**

$|\partial^{\boldsymbol{\nu}} G((u - u_h)(\cdot,\boldsymbol{y}))|$ $\mathcal{O}(h^2)$ by duality argument

Estimate the norm (the lognormal case)

Single-level algorithm

$\|G(u_h^s)\|_{\boldsymbol{\gamma}}$

- **Lemma 5**

Bounds of the form $T(\boldsymbol{y})\,(|\boldsymbol{\nu}|+a_1)!\,\boldsymbol{\beta}^{\boldsymbol{\nu}}$

Multi-level algorithm

$\|G(u_{h_\ell}^{s_\ell} - u_{h_{\ell-1}}^{s_{\ell-1}})\|_{\boldsymbol{\gamma}}$

- **Lemma 6**

- **Lemma 7**

- **Lemma 8**

Summary of analysis (the uniform case)

Assumptions: $\kappa \in H^{-1+t}$ $\quad \sum_{j\ge1} \|\psi_j\|_{L_\infty}^{p_0} < \infty$

$G \in H^{-1+t'}$ $\quad \sum_{j\ge1} \|\nabla\psi_j\|_{L_\infty}^{p_1} < \infty$ $\quad \sum_{j\ge1} \|\psi_j\|_{\mathcal{X}_t}^{p_t} < \infty$

Often $p_t = \dfrac{p_0}{1 - tp_0/d}$

First-order single-level

[K., Schwab, Sloan (2012)]

$$s^{-2(\frac{1}{p_0}-1)} + h^{t+t'} + n^{-\min(\frac{1}{p_0}-\frac{1}{2},1-\delta)}$$

First-order multi-level

[K., Schwab, Sloan (2015)]

$$s_L^{-2(\frac{1}{p_0}-1)} + h_L^{t+t'} + \sum_{\ell=0}^L n_\ell^{-\min(\frac{1}{p_1}-\frac{1}{2},1-\delta)}\left(s_{\ell-1}^{-(\frac{1}{p_0}-\frac{1}{p_1})} + h_{\ell-1}^{t+t'}\right)$$

Higher-order single-level

[Dick, K., Le Gia, Nuyens, Schwab (2014)]

$$s^{-2(\frac{1}{p_0}-1)} + h^{t+t'} + n^{-\frac{1}{p_0}}$$

Higher-order multi-level

[Dick, K., Le Gia, Schwab (2016)]

$$s_L^{-2(\frac{1}{p_0}-1)} + h_L^{t+t'} + \sum_{\ell=0}^L n_\ell^{-\frac{1}{p_t}}\left(s_{\ell-1}^{-(\frac{1}{p_0}-\frac{1}{p_t})} + h_{\ell-1}^{t+t'}\right)$$

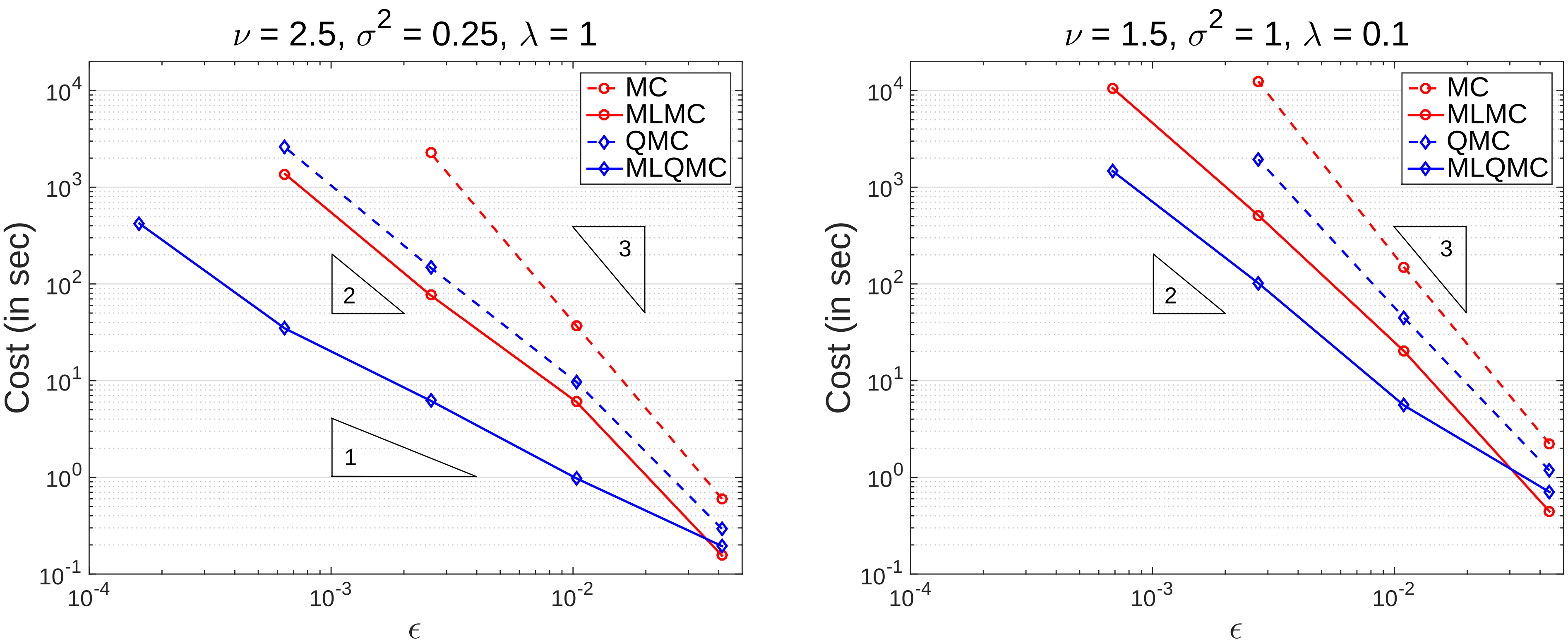
HO-ML does not always win!

High-level view of analysis (the uniform case)

- **QMC** (deterministic or randomized) **convergence** (rate and constant) is independent of truncation dimension
  - Achieved by working in the weighted function space setting
  - Need to estimate the regularity of the PDE solution with respect to $\boldsymbol{y}$
  - Function space weights are chosen to reduce the error bound
  - The chosen weights enter the fast CBC construction of QMC points
- **Combined error** – *truncation error, discretization error, quadrature error*
  - SL: sum of three errors
  - ML: multiplicative effect between FE and QMC, but need stronger regularity in $\boldsymbol{x}$, and regularity estimate simultaneously in $\boldsymbol{x}$ and $\boldsymbol{y}$
  - FO: randomly shifted lattice rules; root-mean-square error bound; Hilbert space setting
  - HO: interlaced polynomial lattice rules; deterministic error bound; non-Hilbert space setting (gain an extra factor of $n^{-1/2}$)
- **Error v.s. cost analysis** ... *depends crucially on the cost assumption*
  - SL: choose n, s, h to balance the three errors
  - ML: choose n_ℓ, s_ℓ, h_ℓ to minimize total cost for a fixed total error (Lagrange multipliers)

MLQMC numerical results (the lognormal case)

- $D = (0,1)^2$, $\kappa \equiv 1$, homogeneous Dirichlet boundary
- $G(u(\cdot,\boldsymbol{y})) = \frac{1}{D^*}\int_{D^*} u(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}$, $D^* = (\frac{3}{4},\frac{7}{8}) \times (\frac{7}{8},1)$
- Piecewise linear finite elements, off-the-shelf randomly shifted lattice rule
- Lognormal case with Matérn kernel: smoothness $\boldsymbol{\nu}$, variance $\boldsymbol{\sigma^2}$, correlation length scale $\boldsymbol{\lambda}$



[K., Scheichl, Schwab, Sloan, Ullmann (to appear)]

Software

http://people.cs.kuleuven.be/~dirk.nuyens/qmc4pde

Construction script in Python

- Input: generic bound on mixed derivatives (weights are set automatically)

$$|\partial^{\boldsymbol{\nu}} F(\boldsymbol{y})| \lesssim \big((|\boldsymbol{\nu}| + a_1)!\big)^{d_1} \prod_{j=1}^{s} (a_2 \mathscr{B}_j)^{\nu_j} \exp(a_3 \mathscr{B}_j |y_j|)$$

$$F(\boldsymbol{y}) = \begin{cases} G(u_h^s) & \text{single-level} \\ G(u_{h_\ell}^s - u_{h_{\ell-1}}^s) & \text{multi-level} \end{cases} \qquad \begin{cases} a_3 = 0 & \text{uniform case} \\ a_3 > 0 & \text{lognormal case} \end{cases}$$

- Output: generating vector for lattice sequence **`z.txt`**

```
## uniform case, 100-dim rule, 2^10 points, with specified bounds b:
./lat-cbc.py --s=100 --m=10 --d2=3 --b="0.1 * j**-3 / log(j+1)"

## lognormal case, 100-dim rule, 2^10 points, with algebraic decay:
./lat-cbc.py --s=100 --m=10 --a2="1/log(2)" --a3=1 --d2=3 --c=0.1
```

- Output: generating matrices for interlaced polynomial lattice rule **`Bs53.col`**

```
## 100-dim rule, 2^10 points, interlacing 3, with bounds from file:
./polylat-cbc.py --s=100 --m=10 --alpha=3 --a1=5 --b_file=bounds.txt
```

Software

http://people.cs.kuleuven.be/~dirk.nuyens/qmc4pde

Point generators in Matlab/Octave (also C++ and Python)

- Lattice sequences

```
load z.txt                       % load generating vector
latticeseq_b2('init0', z)        % initialize the procedural generator
Pa = latticeseq_b2(20, 512);     % first 512 20-dimensional points
Pb = latticeseq_b2(20, 512);     % next 512 20-dimensional points
```

- Interlaced polynomial lattice rules

```
load Bs53.col                       % load generating matrices
digitalseq_b2g('init0', Bs53)       % initialize the procedural generator
Pa = digitalseq_b2g(100, 512);      % first 512 100-dimensional points
Pb = digitalseq_b2g(100, 512);      % next 512 100-dimensional points
```

- Interlaced Sobol′ sequences

```
load sobol_alpha3_Bs53.col     % load generating matrices
digitalseq_b2g('init0', sobol_alpha3_Bs53) % initialize the procedural
Pa = digitalseq_b2g(50, 512);  % first 512 50-dimensional points
Pb = digitalseq_b2g(50, 512);  % next 512 50-dimensional points
```

Concluding remarks

- Three weighted space settings, fast CBC construction
- QMC error bound – rate and constant – can be independent of dimension

- Application of QMC theory: estimate the norm, choose the weights
- Pairing of method and setting: best rate with minimal assumption

- Theory versus practice: *generic QMC may work just fine!*

- Other cost reduction strategies: multi-index, MDM, . . .

- My wish list:
  - Sharper dimension truncation estimate for ML?
  - Weaker regularity requirement for higher order ML?
  - Error as a triple product?
  - Product weights instead of POD weights or SPOD weights?
  - Better choice of weight function for lognormal?
  - Higher order QMC for lognormal (integration over $\mathbb{R}^s$)?

Thanks to my collaborators

AU

Austria

Belgium

Ecuador

Germany

NZ

Poland

Switzerland

UK

USA