Computational Methods in Uncertainty Quantification

Robert Scheichl

Department of Mathematical Sciences
University of Bath
Romberg Visiting Professor at University of Heidelberg

HGS MathComp Compact Course
IWR, Universität Heidelberg, June 11-17 2015

Part 3

Lecture 3

Deterministic Representation – Classical Quadrature & New Ideas

- Stochastic collocation (and polynomial chaos) methods
- Deterministic representation
- Gauss quadrature
- Sparse grids
- Quasi-Monte Carlo quadrature
- Multilevel Quasi-Monte Carlo

RECALL: Computational Challenges

Simulating PDEs with Highly Heterogeneous Random Coefficients

$$-\nabla \cdot (k(\mathbf{x},\omega)\nabla p(\mathbf{x},\omega)) = f(\mathbf{x},\omega), \quad \mathbf{x} \in D \subset \mathbb{R}^d,\ \omega \in \Omega \text{ (prob. space)}$$

- **Sampling** from random field $\log k(\mathbf{x},\omega)$ (correlated Gaussian):
  - truncated Karhunen-Loève expansion of $\log k$ (see above)
  - matrix factorisation, e.g. circulant embedding (FFT)
  - via pseudodifferential "precision" operator (PDE solves)
- **High-Dimensional Quadrature** – (the central problem!):
  - Monte Carlo, **Quasi-Monte Carlo**
  - **stochastic Galerkin/collocation (+ sparse grids)**
- **Solve** large number of **multiscale** deterministic PDEs:
  - Efficient discretisation & FE error analysis (mesh size h)
  - Multigrid Methods, AMG, DD Methods

Weak Formulation & Finite Element Discretisation

Write PDE (subject to $p|_{\partial D} \equiv 0$) in weak form: $p(\cdot,\omega) \in H_0^1(D)$ s.t.

$$\int_D \nabla v \cdot (k(x,\omega)\nabla p(x,\omega))\,\mathrm{d}x = \int_D f(x,\omega) v \,\mathrm{d}x, \quad \forall v \in H_0^1(D).$$

- $\exists ! p(\cdot,\omega) \in H_0^1(D)$ a.s. in $\omega \in \Omega$ (subtle in lognormal case).

Weak Formulation & Finite Element Discretisation

Write PDE (subject to $p|_{\partial D} \equiv 0$) in weak form: $p(\cdot,\omega) \in H_0^1(D)$ s.t.

$$\int_D \nabla v \cdot (k(x,\omega)\nabla p(x,\omega)) \, \mathrm{d}x = \int_D f(x,\omega) v \, \mathrm{d}x, \quad \forall v \in H_0^1(D).$$

- $\exists! p(\cdot,\omega) \in H_0^1(D)$ a.s. in $\omega \in \Omega$ (subtle in lognormal case).
- Let $V_h \subset H_0^1(D)$ be the space of continuous, piecewise linear FEs w.r.t. a mesh $\mathscr{T}_h$ with mesh width $h > 0$.
- Find $p_h(\cdot,\omega) \in V_h$ that satisfies weak form for all $v_h \in V_h$.

Weak Formulation & Finite Element Discretisation

Write PDE (subject to $p|_{\partial D} \equiv 0$) in weak form: $p(\cdot,\omega) \in H_0^1(D)$ s.t.

$$\int_D \nabla v \cdot (k(x,\omega)\nabla p(x,\omega))\,\mathrm{d}x = \int_D f(x,\omega)v\,\mathrm{d}x, \quad \forall v \in H_0^1(D).$$

- $\exists! p(\cdot,\omega) \in H_0^1(D)$ a.s. in $\omega \in \Omega$ (subtle in lognormal case).
- Let $V_h \subset H_0^1(D)$ be the space of continuous, piecewise linear FEs w.r.t. a mesh $\mathscr{T}_h$ with mesh width $h > 0$.
- Find $p_h(\cdot,\omega) \in V_h$ that satisfies weak form for all $v_h \in V_h$.
- Write $p_h(x,\omega) := \sum_{i=1}^{M_h} P_i\varphi_i(x)$. Then this is equivalent to the random matrix system

$$A(\omega)\mathbf{P}(\omega) = \mathbf{F}(\omega)$$

with

$$A_{i,j}(\omega) := \int_D \nabla\varphi_j \cdot (k(x,\omega)\nabla\varphi_i)\,\mathrm{d}x, \quad F_i(\omega) := \int_D f(x,\omega)\varphi_i\,\mathrm{d}x$$

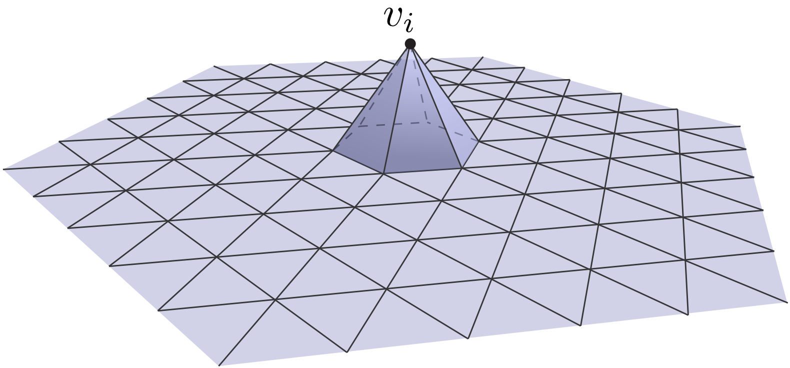
Finite Element Approximation

Short Primer

A short primer on Finite Element discretisation (spatially) on the blackboard . . .

Finite Element Approximation

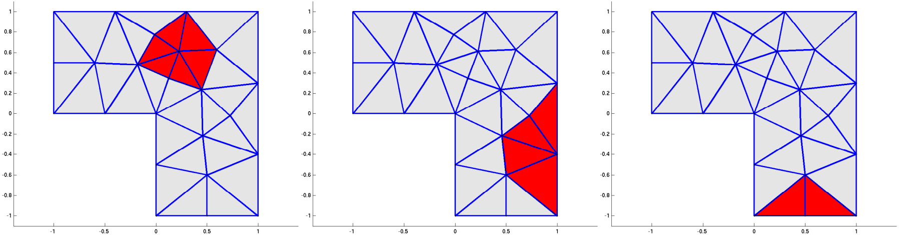
Nodal basis for linear triangles



A nodal basis function with its support.

Finite Element Approximation

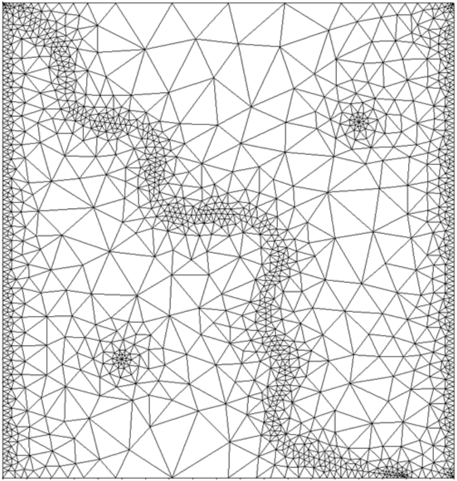
Nodal basis for linear triangles



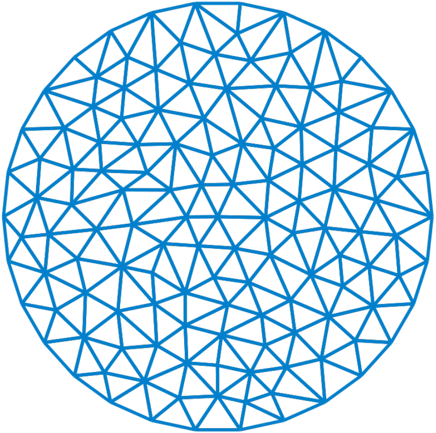
Triangulation of an L-shaped domain with the supports of several basis functions.

Finite Element Approximation

Triangulations



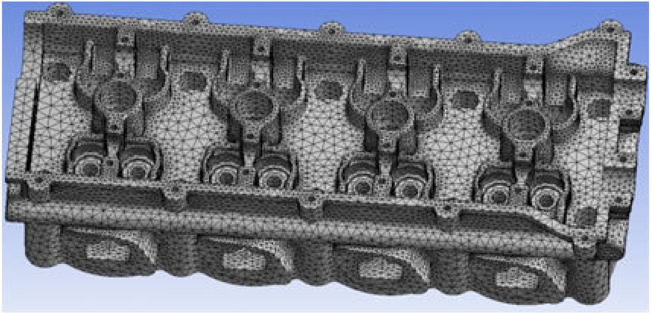
Triangular mesh on a square domain.



Triangular mesh on a polygonal approximation of a circle.

Finite Element Approximation

Triangulations



Tetrahedral mesh of complex 3D geometry (engine block).

Stochastic Collocation

Introduction

Collocation methods are a long-established technique for solving integral or differential equations and are based on requiring the equation under consideration to hold at a finite number of collocation points sufficient to determine an approximate solution in an appropriate finite-dimensional function space (typically global polynomials).

Stochastic Collocation

Introduction

Collocation methods are a long-established technique for solving integral or differential equations and are based on requiring the equation under consideration to hold at a finite number of collocation points sufficient to determine an approximate solution in an appropriate finite-dimensional function space (typically global polynomials).

They were first applied to solve PDEs with random inputs in [Xiu & Hesthaven, 2005] & [Babuška, Nobile & Tempone, 2007]

- Like MC, they reduce to a series of uncoupled deterministic subproblems for which legacy code can be used essentialy unmodified.

Stochastic Collocation

Introduction

Collocation methods are a long-established technique for solving integral or differential equations and are based on requiring the equation under consideration to hold at a finite number of collocation points sufficient to determine an approximate solution in an appropriate finite-dimensional function space (typically global polynomials).

They were first applied to solve PDEs with random inputs in [Xiu & Hesthaven, 2005] & [Babuška, Nobile & Tempone, 2007]

- Like MC, they reduce to a series of uncoupled deterministic subproblems for which legacy code can be used essentialy unmodified.
- Unlike MC, collocation can take advantage of smooth dependence of the solution on the random parameters to yield spectral convergence.

Stochastic Collocation

Introduction

Collocation methods are a long-established technique for solving integral or differential equations and are based on requiring the equation under consideration to hold at a finite number of collocation points sufficient to determine an approximate solution in an appropriate finite-dimensional function space (typically global polynomials).

They were first applied to solve PDEs with random inputs in [Xiu & Hesthaven, 2005] & [Babuška, Nobile & Tempone, 2007]

- Like MC, they reduce to a series of uncoupled deterministic subproblems for which legacy code can be used essentialy unmodified.
- Unlike MC, collocation can take advantage of smooth dependence of the solution on the random parameters to yield spectral convergence.
- Nonlinear problems pose no additional difficulty (unlike stoch. Galerkin)

Stochastic Collocation

Introduction

Collocation methods are a long-established technique for solving integral or differential equations and are based on requiring the equation under consideration to hold at a finite number of collocation points sufficient to determine an approximate solution in an appropriate finite-dimensional function space (typically global polynomials).

They were first applied to solve PDEs with random inputs in [Xiu & Hesthaven, 2005] & [Babuška, Nobile & Tempone, 2007]

- Like MC, they reduce to a series of uncoupled deterministic subproblems for which legacy code can be used essentialy unmodified.
- Unlike MC, collocation can take advantage of smooth dependence of the solution on the random parameters to yield spectral convergence.
- Nonlinear problems pose no additional difficulty (unlike stoch. Galerkin)
- If we are only interested in one or a few scalar quantities of interest stochastic collocation reduces to classical Gauss quadrature.

Deterministic parametric representation

Probabilistic Measures to (weighted) Lebesgue Measures

- Truncated KL expansion leads to a parametrisation by a vector of i.i.d. Gaussian random variables $\mathbf{y} := \{Y_j\}_{j=1}^{s}$.

Deterministic parametric representation

Probabilistic Measures to (weighted) Lebesgue Measures

- Truncated KL expansion leads to a parametrisation by a vector of i.i.d. Gaussian random variables $\mathbf{y} := \{Y_j\}_{j=1}^{s}$.
- Each Y_j has Gaussian density $\rho_j = \rho$ and image $Y_j(\Omega) = \mathbb{R}$. Identify

$$L^2_{\mathbb{P}}(\Omega) \simeq L^2_{\rho}(\mathbb{R}^s), \qquad \text{where} \qquad \boldsymbol{\rho} = \prod_{j=1}^{s} \rho_j \,.$$

random variables with probability measure $\mathbb{P}$ and bdd. 2nd moments are identified with square integrable Lebesgue-measurable fcts (w. weight ρ)

Deterministic parametric representation

Probabilistic Measures to (weighted) Lebesgue Measures

- Truncated KL expansion leads to a parametrisation by a vector of i.i.d. Gaussian random variables $\mathbf{y} := \{Y_j\}_{j=1}^s$.
- Each Y_j has Gaussian density $\rho_j = \rho$ and image $Y_j(\Omega) = \mathbb{R}$. Identify
$$L^2_{\mathbb{P}}(\Omega) \simeq L^2_{\rho}(\mathbb{R}^s), \qquad \text{where} \qquad \boldsymbol{\rho} = \prod_{j=1}^{s} \rho_j \,.$$

  random variables with probability measure $\mathbb{P}$ and bdd. 2nd moments are identified with square integrable Lebesgue-measurable fcts (w. weight ρ)
- Replace $k(x,\omega)$, $p(x,\omega)$ ... with $k(x,\mathbf{y})$, $p(x,\mathbf{y})$.

PDE becomes purely deterministic with high-dim'l parameter space:

$$-\nabla_x \cdot (k(x,\mathbf{y})\nabla_x\, p(x,\mathbf{y})) = f(x,\mathbf{y}), \qquad x \in D, \quad \text{for a.a.} \quad \mathbf{y} \in \mathbb{R}^s,$$

where $$\log k(x,\mathbf{y})) = \log k_0(x) + \sum_{j=1}^{s} \sqrt{\lambda_j}\,\phi_j(x)\, y_j \,.$$

Stochastic Collocation Method

Discretise in stochastic parameters via tensor-product polynomials

Write PDE in weak form: Find $p \in H_0^1(D) \times L_\rho^2(\mathbb{R}^s)$ s.t.

$$\int_D \nabla v \cdot (k(x,\mathbf{y}) \nabla p(x,\mathbf{y}))\, \mathrm{d}x = \int_D f(x,\mathbf{y}) v\, \mathrm{d}x, \quad \forall v \in H_0^1(D).$$

Stochastic Collocation Method

Discretise in stochastic parameters via tensor-product polynomials

Write PDE in weak form: Find $p \in H_0^1(D) \times L_\rho^2(\mathbb{R}^s)$ s.t.

$$\int_D \nabla v \cdot (k(x, \mathbf{y}) \nabla p(x, \mathbf{y}))\, \mathrm{d}x = \int_D f(x, \mathbf{y}) v \, \mathrm{d}x, \quad \forall v \in H_0^1(D).$$

- Let $\bigotimes_{j=1}^{s} \mathscr{P}_{q_j} \subset L_\rho^2(\mathbb{R}^s)$ be the tensor-product space of (global) polynomials of order q_j in dimension j.

Stochastic Collocation Method

Discretise in stochastic parameters via tensor-product polynomials

Write PDE in weak form: Find $p \in H_0^1(D) \times L_\rho^2(\mathbb{R}^s)$ s.t.

$$\int_D \nabla v \cdot (k(x,\mathbf{y})\nabla p(x,\mathbf{y}))\,\mathrm{d}x = \int_D f(x,\mathbf{y})v\,\mathrm{d}x, \quad \forall v \in H_0^1(D).$$

- Let $\bigotimes_{j=1}^s \mathscr{P}_{q_j} \subset L_\rho^2(\mathbb{R}^s)$ be the tensor-product space of (global) polynomials of order q_j in dimension j.
- Find $p_h \in V_h \times \bigotimes_{j=1}^s \mathscr{P}_{q_j}$ that satisfies weak form for all $v_h \in V_h$ at collocation points $\{\mathbf{y}_n\}_{n=1}^{N^{\mathrm{sc}}}$ (zeros of L_ρ^2–orthogonal polynomials)

Stochastic Collocation Method

Discretise in stochastic parameters via tensor-product polynomials

Write PDE in weak form: Find $p \in H^1_0(D) \times L^2_\rho(\mathbb{R}^s)$ s.t.

$$\int_D \nabla v \cdot (k(x,\mathbf{y}) \nabla p(x,\mathbf{y}))\, \mathrm{d}x = \int_D f(x,\mathbf{y}) v \,\mathrm{d}x, \quad \forall v \in H^1_0(D).$$

- Let $\bigotimes_{j=1}^s \mathscr{P}_{q_j} \subset L^2_\rho(\mathbb{R}^s)$ be the tensor-product space of (global) polynomials of order q_j in dimension j.
- Find $p_h \in V_h \times \bigotimes_{j=1}^s \mathscr{P}_{q_j}$ that satisfies weak form for all $v_h \in V_h$ at collocation points $\{\mathbf{y}_n\}_{n=1}^{N^{\mathrm{sc}}}$ (zeros of L^2_ρ–orthogonal polynomials)
- As in MC case, leads to a set of N^{sc} decoupled linear systems (in the case of Stochastic Galerkin methods, the systems are coupled!).

Stochastic Collocation Method

Discretise in stochastic parameters via tensor-product polynomials

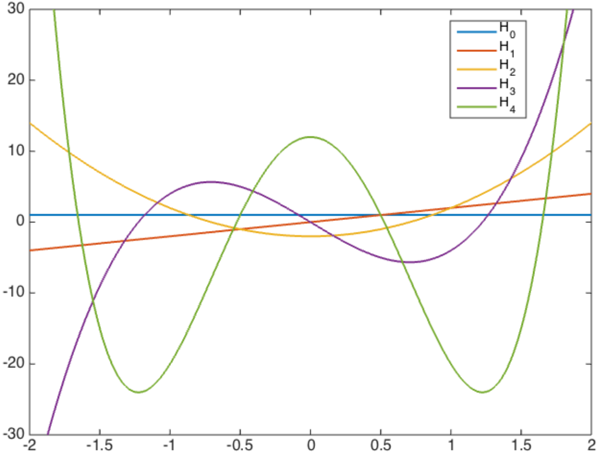
Write PDE in weak form: Find $p \in H_0^1(D) \times L_\rho^2(\mathbb{R}^s)$ s.t.

$$\int_D \nabla v \cdot (k(x, \mathbf{y}) \nabla p(x, \mathbf{y}))\, \mathrm{d}x = \int_D f(x, \mathbf{y}) v\, \mathrm{d}x, \quad \forall v \in H_0^1(D).$$

- Let $\bigotimes_{j=1}^s \mathscr{P}_{q_j} \subset L_\rho^2(\mathbb{R}^s)$ be the tensor-product space of (global) polynomials of order q_j in dimension j.
- Find $p_h \in V_h \times \bigotimes_{j=1}^s \mathscr{P}_{q_j}$ that satisfies weak form for all $v_h \in V_h$ at collocation points $\{\mathbf{y}_n\}_{n=1}^{N^{\mathrm{sc}}}$ (zeros of L_ρ^2–orthogonal polynomials)
- As in MC case, leads to a set of N^{sc} decoupled linear systems (in the case of Stochastic Galerkin methods, the systems are coupled!).
- **But** cost grows v. fast with dimension s & polynomial order q: $N^{\mathrm{sc}} = \mathcal{O}(q^s)$ (full tensor) and $N^{\mathrm{sc}} = \mathcal{O}\left(\frac{(s+q)!}{s!q!}\right)$ (total degree)
  This can be reduced via sparse grid techniques (e.g. Smolyak) $\boxed{\text{see below!}}$

Stochastic Collocation Method

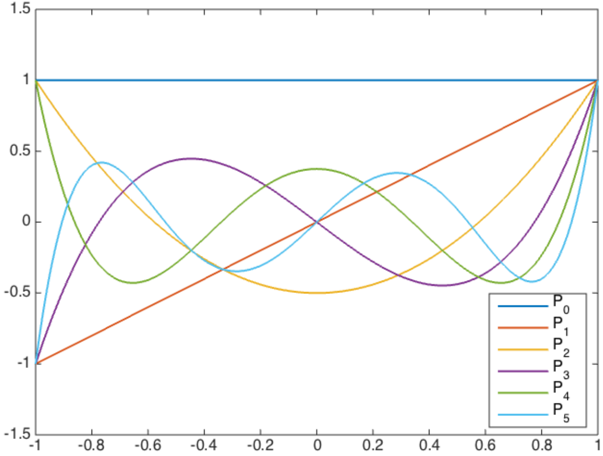
Hermite Polynomials (Gaussian RVs) & Legendre Polynomials (uniform RVs)



Hermite polynomials – orthogonal in $L^2_\rho(\mathbb{R})$

Stochastic Collocation Method

Hermite Polynomials (Gaussian RVs) & Legendre Polynomials (uniform RVs)



Legendre polynomials – orthogonal in $L^2(-1,1)$

Stochastic Collocation Method

Distinction between Quadrature and Interpolation

- Many terms floating around these days in UQ: *Stochastic Galerkin, Stochastic Collocation, Polynomial Chaos Expansions, Gauss Quadrature, Response Surfaces, Surrogates, . . .*
- Unfortunately most papers and books tend to overcomplicate matters and make them look all very daunting.

Stochastic Collocation Method

Distinction between Quadrature and Interpolation

- Many terms floating around these days in UQ: *Stochastic Galerkin, Stochastic Collocation, Polynomial Chaos Expansions, Gauss Quadrature, Response Surfaces, Surrogates, . . .*
- Unfortunately most papers and books tend to overcomplicate matters and make them look all very daunting.
- But essentially they are all based on classical quadrature and interpolation tools for the above high-dimensional problem.

Stochastic Collocation Method

Distinction between Quadrature and Interpolation

- Many terms floating around these days in UQ: *Stochastic Galerkin, Stochastic Collocation, Polynomial Chaos Expansions, Gauss Quadrature, Response Surfaces, Surrogates, . . .*
- Unfortunately most papers and books tend to overcomplicate matters and make them look all very daunting.
- But essentially they are all based on classical quadrature and interpolation tools for the above high-dimensional problem.
- We need to distinguish between
  - the case we have discussed so far, that is statistics (e.g. mean, variance, CDF) of scalar QoIs (**quadrature**)
  - and the case where we want to build a *surrogate model* or a *response surface* (**interpolation**)
- Classically, both of these tasks use very similar tools.

Stochastic Collocation Method

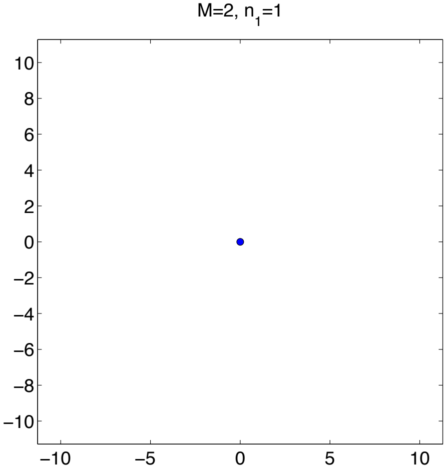
Short Primer on polynomial interpolation and Gauss quadrature

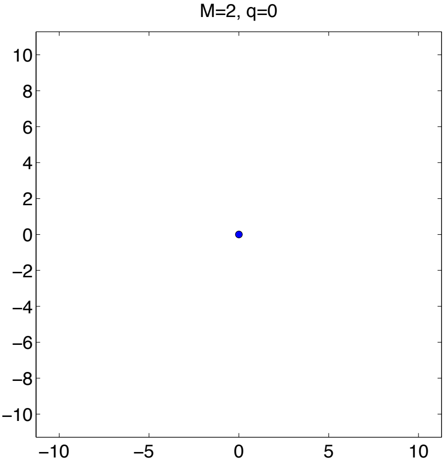
A short primer on polynomial interpolation and Gauss quadrature on the blackboard . . .

Tensor grid vs. Smolyak sparse grid

Based on Gauss-Hermite nodes in $\mathbb{R}^2$

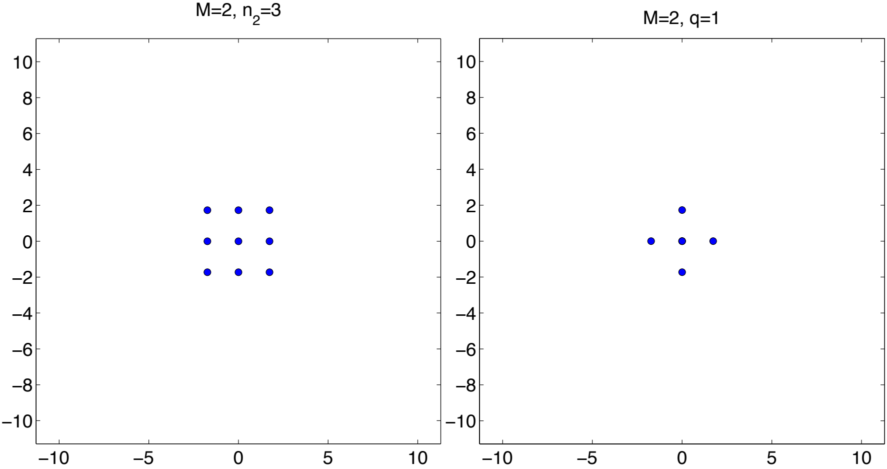
Now simply tensorise the rule in higher dimensions ...





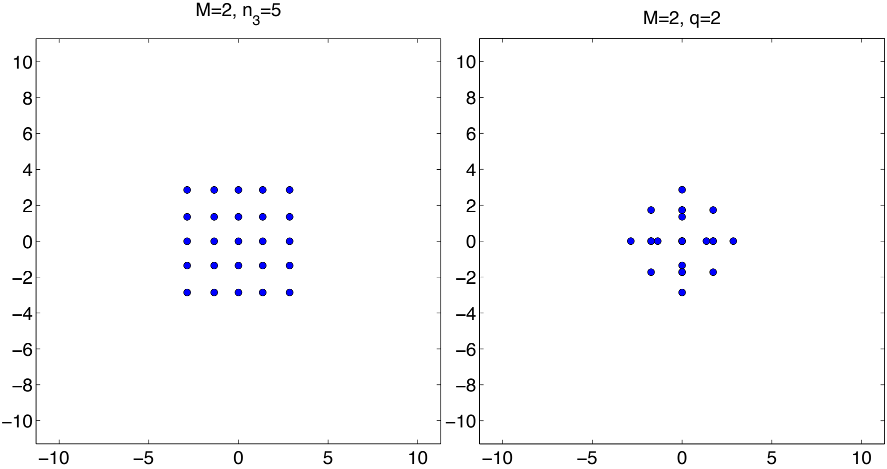
Tensor grid vs. Smolyak sparse grid

Based on Gauss-Hermite nodes in $\mathbb{R}^2$



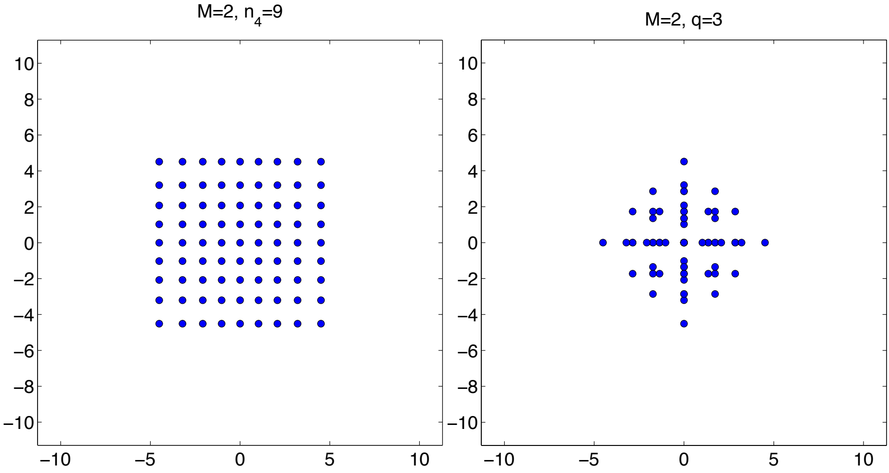
Tensor grid vs. Smolyak sparse grid

Based on Gauss-Hermite nodes in $\mathbb{R}^2$



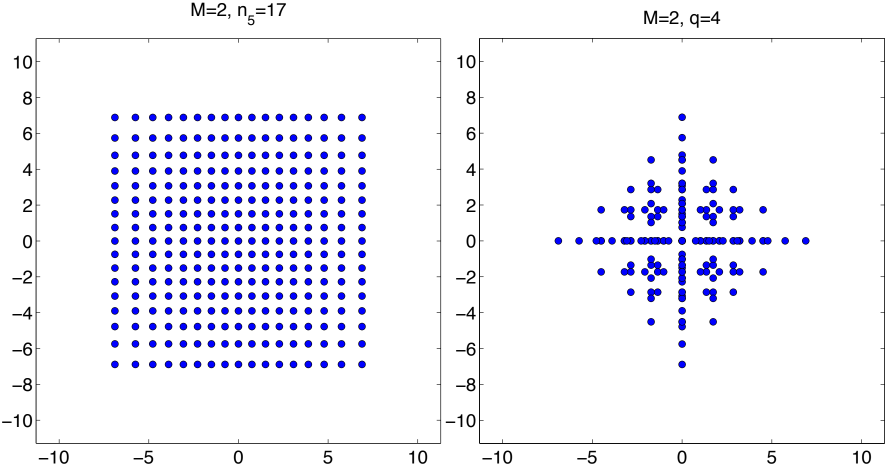
Tensor grid vs. Smolyak sparse grid

Based on Gauss-Hermite nodes in $\mathbb{R}^2$



Tensor grid vs. Smolyak sparse grid

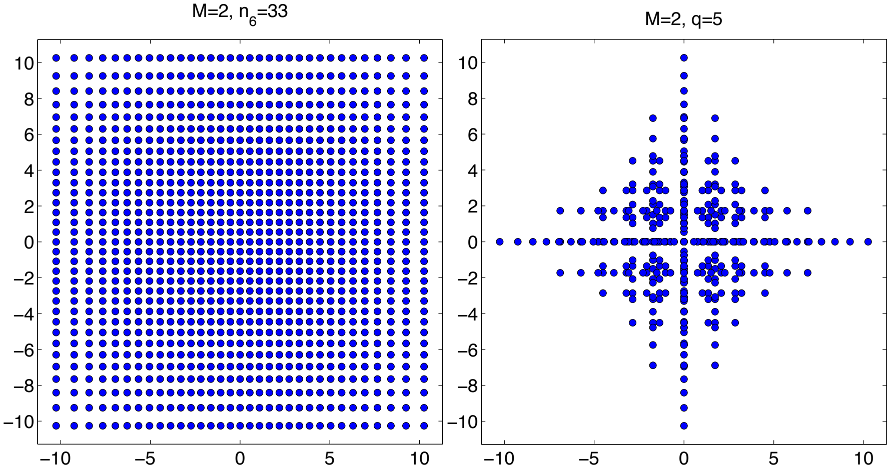
Based on Gauss-Hermite nodes in $\mathbb{R}^2$



Tensor grid vs. Smolyak sparse grid

Based on Gauss-Hermite nodes in $\mathbb{R}^2$

We see the sparse grid has significantly fewer points (even in 2D)!



Stochastic Collocation – The Quadrature Case

Predator-prey example

As an example where we use the polynomial expansion simply as the basis for a quadrature rule, let us recall the predator-prey example:

- The solution operator $\mathscr{G}$ there was mapping the $\mathrm{U}(\Gamma)$ RV $\mathbf{u}_0$ to the RV $Q = u_1(T)$ (with unknown distribution) with $\Gamma = \overline{\mathbf{u}}_0 + [-\varepsilon, \varepsilon]^2$.

Stochastic Collocation – The Quadrature Case

Predator-prey example

As an example where we use the polynomial expansion simply as the basis for a quadrature rule, let us recall the predator-prey example:

- The solution operator $\mathscr{G}$ there was mapping the $\mathrm{U}(\Gamma)$ RV $\mathbf{u}_0$ to the RV $Q = u_1(T)$ (with unknown distribution) with $\Gamma = \overline{\mathbf{u}}_0 + [-\varepsilon, \varepsilon]^2$.
- We can identify $L^2_{\mathbf{P}}(\Omega)$ with $L^2(\Gamma)$, i.e.

$$\mathbf{E}\left[u_1(T)\right] = \frac{1}{|\Gamma|} \int_\Gamma \mathscr{G}(\mathbf{u}_0)\, \mathrm{d}\mathbf{u}_0$$

Stochastic Collocation – The Quadrature Case

Predator-prey example

As an example where we use the polynomial expansion simply as the basis for a quadrature rule, let us recall the predator-prey example:

- The solution operator $\mathscr{G}$ there was mapping the $\mathrm{U}(\Gamma)$ RV $\mathbf{u}_0$ to the RV $Q = u_1(T)$ (with unknown distribution) with $\Gamma = \overline{\mathbf{u}}_0 + [-\varepsilon, \varepsilon]^2$.
- We can identify $L^2_{\mathbf{P}}(\Omega)$ with $L^2(\Gamma)$, i.e.

$$\mathbf{E}\left[u_1(T)\right] = \frac{1}{|\Gamma|} \int_\Gamma \mathscr{G}(\mathbf{u}_0)\, \mathrm{d}\mathbf{u}_0$$

- To approximate $\mathbf{E}\left[u_1(T)\right]$ we approximate $\mathscr{G}$ with $\mathscr{G}_M$, the Euler method with M time steps, and the integral with a (2d-tensorised) Gauss-Legendre quadrature rule (scaled from $[-1,1]$ to $[-\varepsilon,\varepsilon]$):

$$\mathbf{E}\left[u_1(T)\right] \approx \frac{1}{2\varepsilon^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\varepsilon w_i)(\varepsilon w_j) \mathscr{G}_M\big(\overline{\mathbf{u}}_0 + [\varepsilon x_i, \varepsilon x_j]\big)$$

Stochastic Collocation – The Quadrature Case

Predator-prey example

As an example where we use the polynomial expansion simply as the basis for a quadrature rule, let us recall the predator-prey example:

- The solution operator $\mathscr{G}$ there was mapping the $\mathrm{U}(\Gamma)$ RV $\mathbf{u}_0$ to the RV $Q = u_1(T)$ (with unknown distribution) with $\Gamma = \overline{\mathbf{u}}_0 + [-\varepsilon, \varepsilon]^2$.
- We can identify $L^2_{\mathbf{P}}(\Omega)$ with $L^2(\Gamma)$, i.e.

$$\mathbf{E}\left[u_1(T)\right] = \frac{1}{|\Gamma|} \int_\Gamma \mathscr{G}(\mathbf{u}_0)\, \mathrm{d}\mathbf{u}_0$$

- To approximate $\mathbf{E}\left[u_1(T)\right]$ we approximate $\mathscr{G}$ with $\mathscr{G}_M$, the Euler method with M time steps, and the integral with a (2d-tensorised) Gauss-Legendre quadrature rule (scaled from $[-1,1]$ to $[-\varepsilon, \varepsilon]$):

$$\mathbf{E}\left[u_1(T)\right] \approx \frac{1}{2\varepsilon^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\varepsilon w_i)(\varepsilon w_j) \mathscr{G}_M\left(\overline{\mathbf{u}}_0 + [\varepsilon x_i, \varepsilon x_j]\right)$$

- The map is very smooth and so the convergence is exponential.

Stochastic Collocation Methods

Exercise 7

Exercise 7

(a) The Matlab function `g_data(n)` (which is provided) computes the Gauss-Legendre quadrature points and weights for the interval $(-1, 1)$. Transform and tensorise this set of points and hence write a program that evaluates $\mathscr{G}_M$ at the Gauss points for user-defined values of M and n and then evaluates the integral (resp. expected value). Study the convergence with respect to n. What do you observe?

(b) You are also given a set of model codes for the lognormal diffusion problem in 1D. Study the codes and experiment with the different methods. In particular, study the stochastic collocation code (based on Gauss-Legendre points mapped to the entire real line via the inverse CDF for the normal distribution). How fast does it converge for this problem? How does the cost grow with dimension s? Compare to the different Monte Carlo codes (MC, QMC, MLMC, MLQMC).

Stochastic Collocation – The Interpolation Case

The real reason **stochastic collocation** was invented was to construct response surfaces (can then be used for Bayesian inference, control, design).

Stochastic Collocation – The Interpolation Case

The real reason **stochastic collocation** was invented was to construct response surfaces (can then be used for Bayesian inference, control, design).

Here to approximate a parameter-dependent object $u = u(\boldsymbol{\xi})$ with values in an abstract space V, fix a finite-dimensional subspace $V_N = \mathrm{span}\{u_1, \ldots, u_N\} \subset V$ and set

$$u(\boldsymbol{\xi}) \approx u_N(\boldsymbol{\xi}) = \sum_{j=1}^{N} u_j \, \psi_j(\boldsymbol{\xi})$$

with coefficient functions $\psi_j : \Gamma \to \mathbb{R}$ determined by a fixed set of

collocation points $\quad \{\boldsymbol{\xi}_j\}_{j=1}^{N} \subset \Gamma.$

Stochastic Collocation – The Interpolation Case

The real reason **stochastic collocation** was invented was to construct response surfaces (can then be used for Bayesian inference, control, design).

Here to approximate a parameter-dependent object $u = u(\boldsymbol{\xi})$ with values in an abstract space V, fix a finite-dimensional subspace $V_N = \mathrm{span}\{u_1, \dots, u_N\} \subset V$ and set

$$u(\boldsymbol{\xi}) \approx u_N(\boldsymbol{\xi}) = \sum_{j=1}^{N} u_j \, \psi_j(\boldsymbol{\xi})$$

with coefficient functions $\psi_j : \Gamma \to \mathbb{R}$ determined by a fixed set of

collocation points $\quad \{\boldsymbol{\xi}_j\}_{j=1}^N \subset \Gamma.$

Simplest choice for ψ_j: Lagrange basis of multivariate (global) polynomials with respect to a system of unisolvent nodes

$$\boldsymbol{\Xi} := \{\boldsymbol{\xi}_j\}_{j=1}^N \subset \Gamma \, .$$

Stochastic Collocation – The Interpolation Case

Lagrange interpolant

Given a univariate nodal sequence of distinct nodes

$$\chi_k = \{\xi_1^{(k)}, \dots, \xi_{n_k}^{(k)}\}, \qquad k \in \mathbb{N},$$

we denote by $\{\ell_j^{(k)}\}_{j=1}^{n_k}$ the associated Lagrange basis, i.e., the uniquely determined polynomials of degree $n_k - 1$ satisfying

$$\ell_j^{(k)}(\xi_i^{(k)}) = \delta_{i,j}, \qquad j = 1, \dots, n_k.$$

We introduce the univariate interpolation operator

$$I_k : f \mapsto I_k f = \sum_{j=1}^{n_k} f\big(\xi_j^{(k)}\big)\, \ell_j^{(k)} \in \mathscr{P}_{n_k - 1}$$

Stochastic Collocation – The Interpolation Case

Lagrange interpolant

Given a univariate nodal sequence of distinct nodes

$$\chi_k = \{\xi_1^{(k)}, \dots, \xi_{n_k}^{(k)}\}, \qquad k \in \mathbb{N},$$

we denote by $\{\ell_j^{(k)}\}_{j=1}^{n_k}$ the associated Lagrange basis, i.e., the uniquely determined polynomials of degree $n_k - 1$ satisfying

$$\ell_j^{(k)}(\xi_i^{(k)}) = \delta_{i,j}, \qquad j = 1, \dots, n_k.$$

We introduce the univariate interpolation operator

$$I_k : f \mapsto I_k f = \sum_{j=1}^{n_k} f(\xi_j^{(k)})\, \ell_j^{(k)} \in \mathscr{P}_{n_k-1}$$

The tensor-product interpolation operator is then defined as

$$\mathscr{I}_k := I_k \otimes \cdots \otimes I_k : u \mapsto \sum_{|\boldsymbol{\alpha}|_\infty \le n_k} u(\boldsymbol{\xi}_{\boldsymbol{\alpha}})\, \ell_{\alpha_1}^{(k)} \cdot \ldots \cdot \ell_{\alpha_M}^{(k)},$$

where $|\boldsymbol{\alpha}|_\infty = \max_{m=1}^{M} |\alpha_m|$ (i.e. total degree interpolation).

Stochastic Collocation – The Interpolation Case

Example: Elliptic PDE with random coefficients

The fully discrete problem of the elliptic PDE with random coefficients is obtained by approximating the semidiscrete solution $u_h : \Gamma \to V_h$ (where V_h is the FE space) by

$$u_h(\mathbf{x}, \boldsymbol{\xi}) \approx u_{h,p}(\mathbf{x}, \boldsymbol{\xi}) := (\mathscr{I}_p u_h)(\mathbf{x}, \boldsymbol{\xi})\,.$$

Here $\mathscr{I}_p$ is the tensor-product interpolant constructed from univariate Lagrange interpolants of degree p, i.e., based on $p + 1$ distinct nodes in each variable.

Stochastic Collocation – The Interpolation Case

Example: Elliptic PDE with random coefficients

The fully discrete problem of the elliptic PDE with random coefficients is obtained by approximating the semidiscrete solution $u_h : \Gamma \to V_h$ (where V_h is the FE space) by

$$u_h(\mathbf{x}, \boldsymbol{\xi}) \approx u_{h,p}(\mathbf{x}, \boldsymbol{\xi}) := (\mathscr{I}_p u_h)(\mathbf{x}, \boldsymbol{\xi}) \,.$$

Here $\mathscr{I}_p$ is the tensor-product interpolant constructed from univariate Lagrange interpolants of degree p, i.e., based on $p+1$ distinct nodes in each variable.

This entails solving a (deterministic) version of the random PDE for each of the tensor-product interpolation nodes:

Find $u_h(\boldsymbol{\xi}_{\boldsymbol{\alpha}}) \in V_h$ for all $\boldsymbol{\xi}_{\boldsymbol{\alpha}} \in \boldsymbol{\Xi}$ such that

$$\int_D k(\mathbf{x}, \boldsymbol{\xi}_{\boldsymbol{\alpha}}) \nabla u_h(\mathbf{x}, \boldsymbol{\xi}_{\boldsymbol{\alpha}}) \cdot \nabla v_h(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \int_D f(\mathbf{x}, \boldsymbol{\xi}_{\boldsymbol{\alpha}}) v_h(\mathbf{x}) \, \mathrm{d}\mathbf{x} \qquad \forall v_h \in V_h.$$

Curse of Dimensionality (large s)

- **Stochastic collocation methods**
  - cost grows v. fast with dimension s & polynomial order q (faster than exponential) → #stochastic DOFs $\mathscr{O}\left(\frac{(s+q)!}{s!q!}\right)$
  - lower # with sparse grids (Smolyak) but still **exponential** in s!

Curse of Dimensionality (large s)

- **Stochastic collocation methods**
  - cost grows v. fast with dimension s & polynomial order q (faster than exponential) → #stochastic DOFs $\mathcal{O}\left(\frac{(s+q)!}{s!q!}\right)$
  - lower # with sparse grids (Smolyak) but still **exponential** in s!
- **Stochastic Galerkin methods** (not discussed)
  - Huge coupled problems; block dense in general; preconditioners?
  - Anisotropic sparse grids or adaptive **best N-term approximation** can make them dimension independent (in special cases), but needs **a lot of smoothness**!

Curse of Dimensionality (large s)

- **Stochastic collocation methods**
  - cost grows v. fast with dimension s & polynomial order q (faster than exponential) → #stochastic DOFs $\mathcal{O}\left(\frac{(s+q)!}{s!q!}\right)$
  - lower # with sparse grids (Smolyak) but still **exponential** in s!
- **Stochastic Galerkin methods** (not discussed)
  - Huge coupled problems; block dense in general; preconditioners?
  - Anisotropic sparse grids or adaptive **best N-term approximation** can make them dimension independent (in special cases), but needs **a lot of smoothness**!
- Another deterministic alternative: **Quasi-Monte Carlo methods**
  - Faster than MC ($\mathcal{O}(N^{-1})$), but in general cost grows w. s again.
  - Using weighted (repr. kernel) Hilbert spaces, can be made **dimension independent**; requires also **(some) smoothness**!

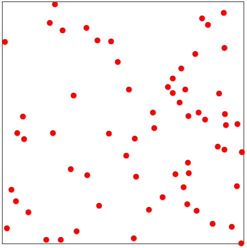
Quasi-Monte Carlo Quadrature

Reducing the number of sample points

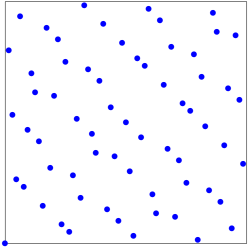
$$I_s(F) := \int_{[0,1]^s} F(\mathbf{y})\,\mathrm{d}\mathbf{y} \;\approx\; \frac{1}{N}\sum_{i=1}^{N} F\big(\mathbf{y}^{(i)}\big) =: Q_s^N(F) \quad \text{(equal weights)}$$

Monte Carlo: $\mathbf{y}^{(n)}$ unif. random
$\mathcal{O}(N^{-1/2})$ convergence
(order of variables irrelevant)

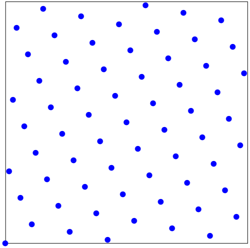
QMC: $\mathbf{y}^{(n)}$ deterministic
close to $\mathcal{O}(N^{-1})$ convergence
(order of variables **v. important**)



64 random points



64 Sobol′ points



64 lattice points

Quasi-Monte Carlo Quadrature

Numerical results for lognormal problem – Test cases and components

Covariance

$$r(\mathbf{x},\mathbf{x}') = \sigma^2 \exp\Big(- \|\mathbf{x}-\mathbf{x}'\|_1/\lambda\Big) \qquad (\ \|\cdot\|_2 \text{ similar})$$

	Case 1	Case 2	Case 3	Case 4	Case 5
σ^2	1	1	1	3	3
λ	1	0.3	0.1	1	0.1

Mixed FEM (RT0 + p.w. const): Uniform grid $h = 1/m$ on $(0,1)^2$

Sampling: circulant embedding, dimension $s = \mathcal{O}(m^2)$ (v. large)

("discrete KL-expansion" via FFT)

QMC Method: randomised QMC with N Sobol' points

$$\mathbb{E}[\mathcal{G}(p)] \approx \int_{[0,1]^s} \mathcal{G}\Big(p_h^s(\cdot, \mathbf{\Phi}^{-1}(\mathbf{y}))\Big)\mathrm{d}\mathbf{y} \approx \frac{1}{N}\sum_{i=1}^{N} \mathcal{G}\Big(p_h^s(\cdot, \mathbf{\Phi}^{-1}(\mathbf{y}^{(i)}))\Big)$$

with $\mathbf{\Phi} : \mathbb{R}^s \to [0,1]^s$ the cumulative normal distribution function.

Quasi-Monte Carlo Quadrature

Numerical results for lognormal problem – Algorithm profile

Time (in sec) on modest laptop for $N = 1000$, CASE 1:

(similar for other cases)

m	s	Setup	Φ^{-1}	FFTW	PDE Solve	TOT
33	4.1 (+3)	0.00	1.0	0.22	4.5	5.9
65	1.7 (+4)	0.01	3.9	1.2	16.5	22
129	6.6 (+4)	0.06	15	5.1	67	92
257	2.6 (+5)	0.15	62	31	290	400
513	1.0 (+6)	0.6	258	145	1280	1750
Order	m^2	m^2	m^2	$m^2 \log m$	$m^2 \log m$	$m^2 \log m$

Quasi-Monte Carlo Quadrature

Numerical results for lognormal problem – Algorithm profile

Time (in sec) on modest laptop for $N = 1000$, CASE 1:

(similar for other cases)

m	s	Setup	Φ^{-1}	`FFTW`	PDE Solve	TOT
33	4.1 (+3)	0.00	1.0	0.22	4.5	5.9
65	1.7 (+4)	0.01	3.9	1.2	16.5	22
129	6.6 (+4)	0.06	15	5.1	67	92
257	2.6 (+5)	0.15	62	31	290	400
513	1.0 (+6)	0.6	258	145	1280	1750
Order	m^2	m^2	m^2	$m^2 \log m$	$m^2 \log m$	$m^2 \log m$

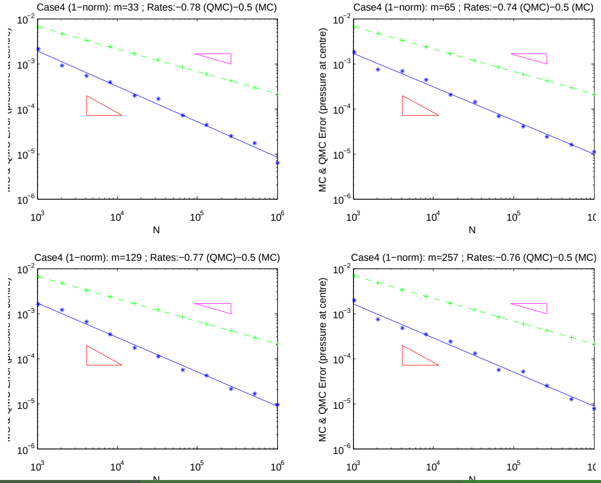
One mixed FE (saddle point system) solve with $\approx 1.3(+6)$ DOF ≈ 1.3s (in 2010)!

Quasi-Monte Carlo Quadrature

Numerical results – Dimension independence (increasing *m* and hence *s*)

Quadrature error for mean pressure at centre (CASE 4)

(no FE error, MC in green, QMC in blue)



Quasi-Monte Carlo Quadrature

Numerical results for lognormal problem – Robustness (varying σ^2 and λ)

Expected value of effective permeability (here FE error present)

h needed to obtain a discretization error $< 10^{-3}$

N needed to obtain (Q)MC error $< 0.5 \times 10^{-3}$ (95% confidence)

σ^2	λ	$1/h$	N (QMC)	N (MC)	CPU (QMC)	CPU (MC)
1	1	17	1.2(+5)	1.9(+7)	0.05 h	8 h
1	0.3	129	3.3(+4)	3.9(+6)	0.9 h	110 h
1	0.1	513	1.2(+4)	5.9(+5)	6.5 h	330 h
3	1	33	4.3(+6)	3.6(+8)	9 h	750 h
3	0.1	513	3.0(+4)	5.8(+5)	20 h (×5)	390 h (×8)

(last line calculated with twice the tolerance!)

Quasi-Monte Carlo Quadrature

Numerical results for lognormal problem – Robustness (varying σ^2 and λ)

Expected value of effective permeability (here FE error present)

h needed to obtain a discretization error $< 10^{-3}$

N needed to obtain (Q)MC error $< 0.5 \times 10^{-3}$ (95% confidence)

σ^2	λ	$1/h$	N (QMC)	N (MC)	CPU (QMC)	CPU (MC)
1	1	17	1.2(+5)	1.9(+7)	0.05 h	8 h
1	0.3	129	3.3(+4)	3.9(+6)	0.9 h	110 h
1	0.1	513	1.2(+4)	5.9(+5)	6.5 h	330 h
3	1	33	4.3(+6)	3.6(+8)	9 h	750 h
3	0.1	513	3.0(+4)	5.8(+5)	20 h (×5)	390 h (×8)

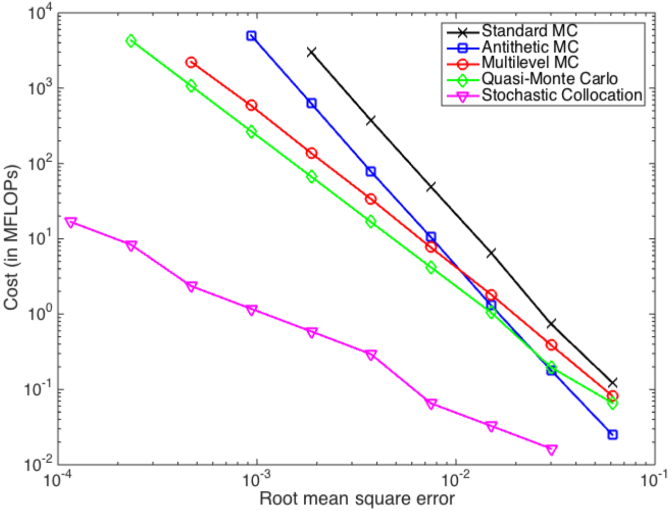
(last line calculated with twice the tolerance!)

Smaller λ needs smaller h but also smaller N (ergodicity).

Strong superiority of QMC in all cases.

Quasi–Monte Carlo Quadrature

Predator-prey problem (discretisation error + quadrature error)



Quasi–Monte Carlo Quadrature

How do they work?

Starting point: equal-weight quadrature rule $Q_s^N(F) := \dfrac{1}{N}\sum_{i=1}^{N} F(\mathbf{y}^{(i)})$

How to choose $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)}$?

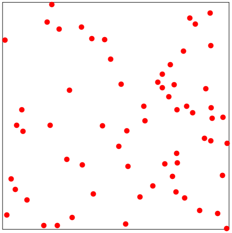
Quasi–Monte Carlo Quadrature

How do they work?

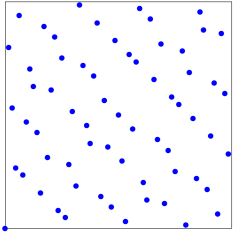
Starting point: equal-weight quadrature rule $Q_s^N(F) := \frac{1}{N}\sum_{i=1}^{N} F(\mathbf{y}^{(i)})$

How to choose $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)}$?

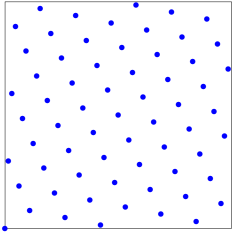
- **Low discrepancy points**: Sobol (1950s), Faure, Niederreiter (1980s), Dick ...
- **Lattice rules**: Korobov, Hlawka, Hua, Wang (50s), Sloan...



64 random points



64 Sobol′ points



64 lattice points

Quasi–Monte Carlo Quadrature

How do they work? [Kuo, Schwab, Sloan, ANZIAM J 2011]

Choose the Hilbert space $\mathscr{W}_s := (H^1(0,1))^s$ with norm

$$\|F\|^2_{\mathscr{W}_s} := \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \int_{[0,1]^{|\mathfrak{u}|}} \left(\frac{\partial^{|\mathfrak{u}|} F}{\partial \mathbf{y}_{\mathfrak{u}}} (\mathbf{y}_{\mathfrak{u}}; \mathbf{1}) \right)^2 \mathrm{d}\mathbf{y}_{\mathfrak{u}} \,,$$

i.e. all mixed first derivatives w.r.t. the parameters are bounded.

Quasi–Monte Carlo Quadrature

How do they work? [Kuo, Schwab, Sloan, ANZIAM J 2011]

Choose the Hilbert space $\mathscr{W}_s := (H^1(0,1))^s$ with norm

$$\|F\|^2_{\mathscr{W}_s} := \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \int_{[0,1]^{|\mathfrak{u}|}} \left(\frac{\partial^{|\mathfrak{u}|} F}{\partial \mathbf{y}_{\mathfrak{u}}} (\mathbf{y}_{\mathfrak{u}}; \mathbf{1}) \right)^2 \mathrm{d}\mathbf{y}_{\mathfrak{u}} \,,$$

i.e. all mixed first derivatives w.r.t. the parameters are bounded.

- Due to linearity of error in F we have

$$|I_s(F) - Q_s^N(F)| \;\leq\; e^{\text{wor}} \left(\{\mathbf{y}^{(i)}\}, \mathscr{W}_s \right) \, \|F\|_W$$

with

$$e^{\text{wor}} \left(\{\mathbf{y}^{(i)}\}, \mathscr{W}_s \right) \;:=\; \sup_{\|F\|_{\mathscr{W}_s} \leq 1} |I_s(F) - Q_s^N(F)|$$

the worst case error (related to discrepancy of the point set).

Quasi–Monte Carlo Quadrature

How do they work? [Kuo, Schwab, Sloan, ANZIAM J 2011]

- $\mathscr{W}_s$ is a reproducing kernel Hilbert space with kernel

$$K(\mathbf{y},\mathbf{z}) := \prod_{i=1}^{s} \left(1 + \min(1 - y_i, 1 - z_i)\right).$$

Quasi–Monte Carlo Quadrature

How do they work? [Kuo, Schwab, Sloan, ANZIAM J 2011]

- $\mathscr{W}_s$ is a reproducing kernel Hilbert space with kernel

$$K(\mathbf{y}, \mathbf{z}) := \prod_{i=1}^{s} \left(1 + \min(1 - y_i, 1 - z_i)\right).$$

- It is an easy exercise to show that $e^{\mathrm{wor}}\left(\{\mathbf{y}^{(i)}\}, \mathscr{W}_s\right)$ can be written down explicitly in terms of $K(\mathbf{y}, \mathbf{z})$.

Quasi–Monte Carlo Quadrature

How do they work? [Kuo, Schwab, Sloan, ANZIAM J 2011]

- $\mathscr{W}_s$ is a reproducing kernel Hilbert space with kernel

$$K(\mathbf{y},\mathbf{z}) := \prod_{i=1}^{s} \left(1 + \min(1-y_i, 1-z_i)\right).$$

- It is an easy exercise to show that $e^{\mathrm{wor}}\left(\{\mathbf{y}^{(i)}\}, \mathscr{W}_s\right)$ can be written down explicitly in terms of $K(\mathbf{y},\mathbf{z})$.

- As in the classical analysis, this leads to the dimension-dependent bound for standard QMC points sets

$$e^{\mathrm{wor}}\left(\{\mathbf{y}^{(i)}\}, \mathscr{W}_s\right) \lesssim \frac{(\log N)^s}{N}$$

  which is unsatisfactory because it only starts to decay when N is exponetially large in s $\longrightarrow$ Introduce weights in $\mathscr{W}_s$, ... [Sloan, Woźniakoski, 1998 & 2001]

Quasi–Monte Carlo Quadrature

How do they work? [Kuo, Schwab, Sloan, ANZIAM J 2011]

Choose instead weighted Hilbert space $\mathscr{W}_{s,\boldsymbol{\gamma}} := H^1_{\gamma_1}(0,1) \times \ldots \times H^1_{\gamma_s}(0,1)$ with norm

$$\|F\|^2_{\mathscr{W}_s} := \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left(\frac{\partial^{|\mathfrak{u}|} F}{\partial \mathbf{y}_{\mathfrak{u}}} (\mathbf{y}_{\mathfrak{u}}; \mathbf{1}) \right)^2 \mathrm{d}\mathbf{y}_{\mathfrak{u}} \ ,$$

e.g. $\gamma_{\mathfrak{u}} = \prod_{i=1}^s \gamma_i$ (product weights) or $\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|} \prod_{i=1}^s \gamma_i$ (POD weights).

Quasi–Monte Carlo Quadrature

How do they work? [Kuo, Schwab, Sloan, ANZIAM J 2011]

Choose instead weighted Hilbert space $\mathscr{W}_{s,\boldsymbol{\gamma}} := H^1_{\gamma_1}(0,1) \times \ldots \times H^1_{\gamma_s}(0,1)$ with norm

$$\|F\|^2_{\mathscr{W}_s} := \sum_{\mathfrak{u}\subseteq\{1,\ldots,s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left(\frac{\partial^{|\mathfrak{u}|} F}{\partial \mathbf{y}_{\mathfrak{u}}}(\mathbf{y}_{\mathfrak{u}}; \mathbf{1}) \right)^2 \mathrm{d}\mathbf{y}_{\mathfrak{u}} \ ,$$

e.g. $\gamma_{\mathfrak{u}} = \prod_{i=1}^s \gamma_i$ (product weights) or $\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|} \prod_{i=1}^s \gamma_i$ (POD weights).

- Now under some **decay** (or summability) conditions on the weights it is possible to show (for certain rules)

  $$e^{\text{wor}}\left(\{\mathbf{y}^{(i)}\}, \mathscr{W}_{s,\boldsymbol{\gamma}} \right) \lesssim N^{-1+\delta}$$

  for some $0 < \delta \leq 1/2$; decay depends on smoothness of F, in particular on the size of the mixed first derivatives.

Quasi–Monte Carlo Quadrature

How do they work? [Kuo, Schwab, Sloan, ANZIAM J 2011]

Choose instead weighted Hilbert space $\mathscr{W}_{s,\boldsymbol{\gamma}} := H^1_{\gamma_1}(0,1) \times \ldots \times H^1_{\gamma_s}(0,1)$ with norm

$$\|F\|^2_{\mathscr{W}_s} := \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left(\frac{\partial^{|\mathfrak{u}|} F}{\partial \mathbf{y}_{\mathfrak{u}}} (\mathbf{y}_{\mathfrak{u}}; \mathbf{1}) \right)^2 \mathrm{d}\mathbf{y}_{\mathfrak{u}} \,,$$

e.g. $\gamma_{\mathfrak{u}} = \prod_{i=1}^s \gamma_i$ (product weights) or $\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|} \prod_{i=1}^s \gamma_i$ (POD weights).

- Now under some **decay** (or summability) conditions on the weights it is possible to show (for certain rules)

  $$e^{\text{wor}} \left(\{\mathbf{y}^{(i)}\}, \mathscr{W}_{s,\boldsymbol{\gamma}} \right) \lesssim N^{-1+\delta}$$

  for some $0 < \delta \leq 1/2$; decay depends on smoothness of F, in particular on the size of the mixed first derivatives.

- Hence ordering of coordinates is **crucial** for dimension independence. There are no point sets that are equally good in all coordinates.

Quasi-Monte Carlo Lattice Rule (of rank 1)

[Sloan & Joe, Lattice Methods for Multiple Integration, OUP, 1994]

Given a *generating vector* $\mathbf{z}_{\mathrm{gen}} \in \{1, \ldots, N-1\}^s$ and a *random shift* $\boldsymbol{\Delta} \sim U\big[(0,1)^s\big]$:

$$\mathbf{z}^{(i)} := \mathrm{frac}\left(\frac{i\,\mathbf{z}_{\mathrm{gen}}}{N} + \boldsymbol{\Delta}\right), \qquad i = 1, \ldots, N$$

The random shift makes estimator unbiased (!) and is very convenient for analysis and for adaptive error control

Quasi-Monte Carlo Lattice Rule (of rank 1)

[Sloan & Joe, Lattice Methods for Multiple Integration, OUP, 1994]

Given a *generating vector* $\mathbf{z}_{\text{gen}} \in \{1, \ldots, N-1\}^s$ and a *random shift* $\boldsymbol{\Delta} \sim U\big[(0,1)^s\big]$:

$$\mathbf{z}^{(i)} := \operatorname{frac}\left(\frac{i\,\mathbf{z}_{\text{gen}}}{N} + \boldsymbol{\Delta}\right), \qquad i = 1, \ldots, N$$

The random shift makes estimator unbiased (!) and is very convenient for analysis and for adaptive error control

- Efficient **component-by-component (CBC)** construction available – controlled by weights $\gamma_{\mathfrak{u}}$. (see [Sloan, Reztsov, Kuo, Joe, 2002] and www.maths.unsw.edu.au/∼fkuo)
- For **infinite dimensions and improper integrals**, need extra weight function ψ^2 in $\|\cdot\|_{\mathscr{W}_{s,\gamma}}$. [Kuo, Sloan, Wasilkowski, Waterhouse, 2010], [Kuo, Nicholls, 2014]

Quasi-Monte Carlo Lattice Rule (of rank 1)

Bounding $\|F\|_{\mathscr{W}_{s,\gamma}}$ in the lognormal model problem

- To show $\mathscr{G}(p_h^s) \in \mathscr{W}_{s,\gamma}$, we first bound the mixed 1st derivatives of p_h^s w.r.t. parameters in any finite subset $\mathfrak{u} \subset \mathbb{N}$:

$$\left| \frac{\partial^{|\mathfrak{u}|} p_h^s}{\partial \mathbf{y}_{\mathfrak{u}}}(\cdot, \mathbf{y}) \right|_{H^1(D)} \le \frac{\|f\|_{H^{-1}(D)}}{k_{\min}(\mathbf{y})} \frac{|\mathfrak{u}|!}{\ln 2^{|\mathfrak{u}|}} \left(\prod_{j \in \mathfrak{u}} \sqrt{\mu_j} \, \|\phi_j\|_{L^\infty(D)} \right)$$

Quasi-Monte Carlo Lattice Rule (of rank 1)

Bounding $\|F\|_{\mathscr{W}_{s,\gamma}}$ in the lognormal model problem

- To show $\mathscr{G}(p_h^s) \in \mathscr{W}_{s,\gamma}$, we first bound the mixed 1st derivatives of p_h^s w.r.t. parameters in any finite subset $\mathfrak{u} \subset \mathbb{N}$:

$$\left| \frac{\partial^{|\mathfrak{u}|} p_h^s}{\partial \mathbf{y}_{\mathfrak{u}}}(\cdot, \mathbf{y}) \right|_{H^1(D)} \leq \frac{\|f\|_{H^{-1}(D)}}{k_{\min}(\mathbf{y})} \, \frac{|\mathfrak{u}|!}{\ln 2^{|\mathfrak{u}|}} \left(\prod_{j \in \mathfrak{u}} \sqrt{\mu_j} \, \|\phi_j\|_{L^\infty(D)} \right)$$

- Assume $\mathscr{G}(p_h^s)$ linear. If KL-eigenvalues μ_j decay suff'ly fast we can find weights $\gamma_{\mathfrak{u}}$ s.t. $\mathscr{G}(p_h^s) \in \mathscr{W}_{s,\gamma}$.

Theorem [Graham, Kuo, Nichols, RS, Schwab, Sloan, 2014]

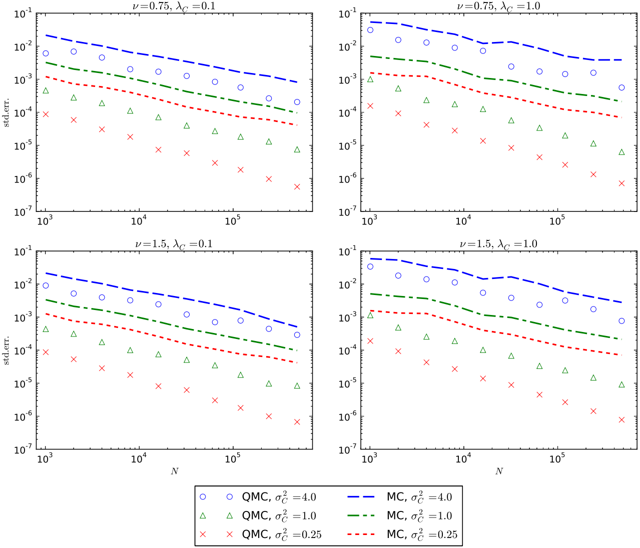
$$\mathbb{E}[\mathscr{G}(p_h^s)] - Q_N^s(\mathscr{G}(p_h^s)) = \mathscr{O}(N^{-1/2}) \quad \text{if} \quad \mu_j \|\phi_j\|^2_{L^\infty(D)} = O(j^{-2-\delta})$$

$$\mathbb{E}[\mathscr{G}(p_h^s)] - Q_N^s(\mathscr{G}(p_h^s)) = \mathscr{O}(N^{-1+\delta}) \quad \text{if} \quad \mu_j \|\phi_j\|^2_{L^\infty(D)} = O(j^{-3})$$

Optimal rates (**provable**) for Matérn with $\nu > \frac{3}{2}d$.

Quasi-Monte Carlo Lattice Rule (of rank 1)

Quadrature Error (1D, Matérn covariance, rank-1 lattice rule)



Quasi-Monte Carlo Lattice Rule (of rank 1)

Quadrature Error (1D, Matérn covariance, rank-1 lattice rule)

Rates

ν	σ^2	$\lambda = 0.1$	$\lambda = 1.0$
0.75	0.25	0.82	0.89
	1.00	0.64	0.83
	4.00	0.60	0.63
1.5	0.25	0.80	0.86
	1.00	0.66	0.73
	4.00	0.58	0.55

Quasi-Monte Carlo Lattice Rule (of rank 1)

Quadrature Error (1D, Matérn covariance, rank-1 lattice rule)

Rates

ν	σ^2	$\lambda = 0.1$	$\lambda = 1.0$
0.75	0.25	0.82	0.89
	1.00	0.64	0.83
	4.00	0.60	0.63
1.5	0.25	0.80	0.86
	1.00	0.66	0.73
	4.00	0.58	0.55

Higher order QMC methods (polynomial lattice rules): $\mathcal{O}(N^{-k})$
[Dick, Pillichshammer, 2007] – but requires again more smoothness from F

Quasi-Monter Carlo Methods

Exercise 8

Exercise 8

(a) Use the file `lattice-38005-1024-1048576.5000.txt` from Kuo's webpage `web.maths.unsw.edu.au/~fkuo/lattice/index.html` that contains a generating vector for a rank-1 lattice rule with equal weights $\gamma_j = 0.05$ to construct a set of QMC points on the unit square $[0,1]^2$. Randomise and use this set to approximate $\mathbf{E}[u_1(T)]$ in the predator-prey example and compare the convergence of this QMC rule with the convergence of your other codes.

(b) As part of the model codes for the lognormal diffusion problem in 1D you will also find a QMC code there. Experiment also with that code.

Multilevel Quasi-Monte Carlo

Combining approaches and gains – Complexity theorem

- QMC acceleration **complimentary** to ML variance reduction**!**

Multilevel Quasi-Monte Carlo

Combining approaches and gains – Complexity theorem

- QMC acceleration **complimentary** to ML variance reduction**!**

  [Giles, Waterhouse '09] (SDE), [Kuo, Schwab, Sloan '12] (uniform affine), [Harbrecht et al, '13] (**lognormal**, but **not** tractable & no effic. gains)

Multilevel Quasi-Monte Carlo

Combining approaches and gains – Complexity theorem

- QMC acceleration **complimentary** to ML variance reduction**!**

  [Giles, Waterhouse '09] (SDE), [Kuo, Schwab, Sloan '12] (uniform affine), [Harbrecht et al, '13] (**lognormal**, but **not** tractable & no effic. gains)

Theorem (Multilevel QMC) [Kuo, RS, Schwab, Sloan, Ullmann, to be submitted]

Assume FE error $\mathcal{O}(M_\ell^{-\alpha})$, Cost/sample $\mathcal{O}(M_\ell^{\gamma})$ (as above) **and**

$$\mathbb{V}_\Delta\Big[Q^s_{N_\ell}(\mathcal{G}(p_\ell)) - Q^s_{N_\ell}(\mathcal{G}(p_{\ell-1}))\Big] = \mathcal{O}(N_\ell^{-\eta} M_\ell^{-\beta}), \quad \text{with } 1 \le \eta < 2.$$

There exist L, $\{N_\ell\}_{\ell=0}^L$ (computable on the fly) to obtain MSE $< \varepsilon^2$ with

$$\mathrm{Cost}(\widehat{Q}_L^{\mathrm{MLQ}}) \;=\; \mathcal{O}\left(\varepsilon^{-\frac{2}{\eta}-\max\left(0,\frac{\eta\gamma-\beta}{\eta\alpha}\right)}\right) \qquad + \text{ possible log's}$$

Multilevel Quasi-Monte Carlo

Discussion and setup for numerical test case

- If QMC optimal (i.e. $\eta \approx 2$), if $\beta \approx 2\alpha$ and $\gamma \approx 1$ (e.g. via AMG) then

$$\mathrm{Cost}(\widehat{Q}_L^{\mathrm{MLQ}}) \;=\; \mathcal{O}\left(\varepsilon^{-\max\left(1,\frac{d}{\alpha}\right)}\right)$$

- **Better than MLMC** complexity $\mathcal{O}(\varepsilon^{-\max\left(2,\frac{d}{\alpha}\right)})$ for $\alpha \geq d/2$.

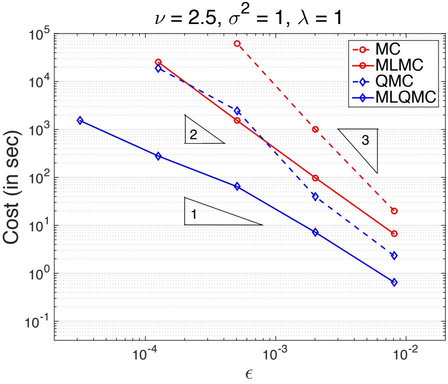
- **Optimal** for $\alpha \leq d$! In that case the cost is $\mathcal{O}(\varepsilon^{-1})$.

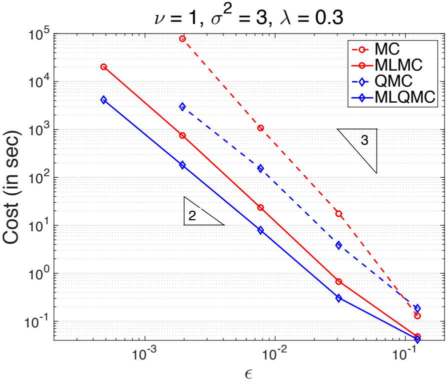
- **Also:** Multilevel stochastic collocation [Teckentrup, Jantsch, Webster, Gunzburger, 2014]

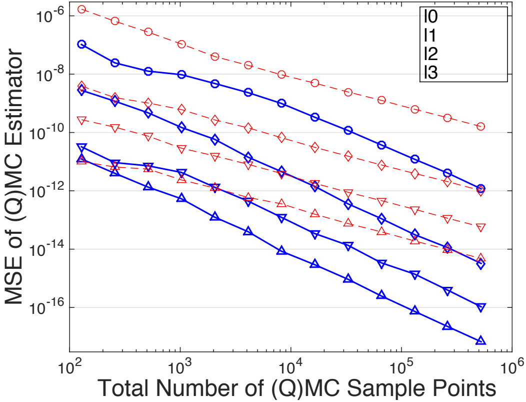
- **Numerical experiment:**
  - $D = (0,1)^2$; stand. FEs; $Q = \frac{1}{|D^*|}\int_{D^*} p\,\mathrm{d}x$
  - Matérn cov.; truncated KLE w. $s \sim h^{-2/\nu}$;
  - randomised lattice rule with $\gamma_j = 1/j^2$.

Multilevel Quasi-Monte Carlo

Numerical Experiments







Convergence of MSE of the QMC/MC estimators for $\mathbb{E}[F_\ell - F_{\ell-1}]$

$(\nu = 1.5,\ \lambda = 1,\ \sigma^2 = 1,\ s_L = 27,\ h_0 = 1/8)$