High-Dimensional Approximation and Applications in Uncertainty Quantification

I. Introduction and Motivation

Prof. Dr. Robert Scheichl
r.scheichl@uni-heidelberg.de

Dr. Alexander Gilbert
a.gilbert@uni-heidelberg.de

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Institut für Angewandte Mathematik, Universität Heidelberg

Summer Semester 2020

1. Aims & Course Structure

Background

- **Mathematical modelling**, e.g. in the form of differential equations, is essential to understand, optimise, control or predict physical, biological and engineering processes.
- **Numerical methods** are central in solving these often very complex mathematical models.
- In the previous numerics modules, we have learned about efficient numerical methods, in particular for differential equations.
- These methods have reached high state of maturity & sophistication.
- **But** models have input data that are typically not known precisely
  (parameters, source term, domain shape, boundary conditions, etc...)
- **It is of great importance to determine these parameters, their influence on the solution & uncertainties due to their variability.**

To find (and analyse) efficient numerical methods for these tasks is still a **very active field** of research and will be the **focus of this course**.

Aims

1. Motivate the interest in **high dimensional numerical approximation** via its huge importance in **Uncertainty Quantification (UQ)**.
2. Introduce one of the **key obstacles**, the so-called

   Curse of Dimensionality.
3. Introduce **Monte Carlo–type** methods that **circumvent** the curse of dimensionality for **high-dimensional quadrature** and **analyse** them,

   in particular **Multilevel Monte Carlo** and **quasi-Monte Carlo**.
4. How to **adapt polynomial-based** quadrature and approximation methods for high dimensional problems?

   Sparse Grids.
5. **Breaking the curse of dimensionality** with polynomial methods.
6. **A more general tool: Low-rank tensor approximation** methods.

Course Structure

- As in previous semesters, we will use **moodle** to provide you with lecture material and to communicate with you. The moodle site for this course (on the **new server**) is

  `https://moodle.uni-heidelberg.de/course/view.php?id=1595`

  Access key (=Einschreibeschlüssel): in first lecture
- **Please use the moodle "Discussion Forum" for any questions you have on the course and also answer each others questions.**
- These slides will also form the **lecture notes**.
- We will **pre-record** individual **lectures** and put the videos on moodle.
- We aim to also have one **problem sheet** per week. If you want **feedback** on your solution, please submit a scan/photo on moodle.
- The lecture on Tuesday at 9:15 will be used as both a **question-and-answer session** and a **tutorial** where we will present model solutions to the problem sheets.

Assessment

- There will be **no formal** admission requirements for the exam to this module (= **keine** Zulassungskriterien), but we **strongly recommend** that you attempt the problem sheets every week.
- Provided the measures for the Corona Crisis allow it, we aim for the **final exam** to be an **oral exam** (in person).

Literature & Other Resources

[1] G.J. Lord, C.E. Powell, T. Shardlow, *An Introduction to Computational Stochastic PDEs*, Cambridge University Press, Cambridge, UK, 2014.

[2] M.B. Giles, Multilevel Monte Carlo methods, *Acta Numer.* **24**, 259–328, 2015. https://doi.org/10.1017/S096249291500001X

[3] J. Dick, F.Y. Kuo, I. H. Sloan, High-dimensional integration: The quasi-Monte Carlo way, *Acta Numer.* **22**, 133–288, 2013. https://doi.org/10.1017/S0962492913000044

[4] H-J. Bungartz, M. Griebel, Sparse grids, *Acta Numer.* **13**, 147–169, 2004. https://doi.org/10.1017/S0962492904000182

[5] R. Ghanem, P. Spanos, *Stochastic Finite Elements*, Springer, New York, 1991.

[6] W. Hackbusch, Numerical tensor calculus, *Acta Numer.* **23**, 651–742, 2014. https://doi.org/10.1017/S0962492914000087

Additional Material:

Probability Primer. Some basic tools and concepts from probability theory.

FE Primer. The model PDE problem and its discretisation via finite elements.

Course Content

- Uncertainty Quantification and the "Curse of Dimensionality"
- Monte Carlo methods (in particular, multilevel Monte Carlo)
- Quasi-Monte Carlo methods
- Sparse grids — Quadrature and Approximation
- Best N-term approximation and adaptive methods
- Low-rank tensor approximation

2. What is Uncertainty Quantification?

What are the Challenges in UQ?

- What is uncertainty quantification (UQ) about?

- What is uncertainty?
- How can uncertainty be described?
- How can the effects of uncertainty be treated and quantified?
- A case study – radioactive waste disposal.
- Methods for solving the resulting mathematical problems.
- What are the challenges?

Please note that this first part is based on the lecture notes of "Mathematische Methoden der Unsicherheitsquantifizierung" at the **TU Chemnitz** by Prof. Oliver Ernst.

`https://www.tu-chemnitz.de/mathematik/numa/lehre/uq-2014`

What is 'uncertain'?

uncertain: *not able to be relied on; not known or definite.*

Oxford Collegiate Dictionary

uncertain: *not exactly known or decided; not definite or fixed; not known beyond doubt; not constant*

Merriam Webster Online Dictionary

uncertain: *not able to be accurately known or predicted; not precisely determined, established, or decided; liable to variation; changeable*

Collins Online Dictionary

A Poetic Description

There are known knowns;
there are things we know we know.

We also know there are known unknowns;
that is to say, we know there are some things we do not know.

But there are also unknown unknowns – the ones we don't know we don't know.

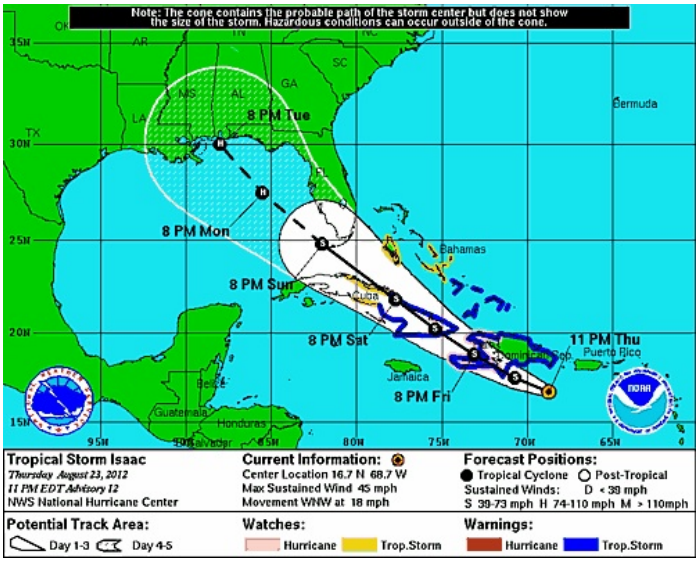
U. S. Secretary of Defence, Donald Rumsfeld
DoD News Briefing; Feb. 12, 2002

Uncertainty in Modern Life

Many aspects of modern life involve uncertainty:

- **Social systems:** military, finance, insurance industry, elections
- **Environmental systems:** weather, climate, seismic, subsurface geophysics
- **Engineering systems:** automobiles, aircraft, bridges, structures
- **Biological systems:** health and medicine, pharmaceuticals, gene expression, cancer research
- **Physical systems:** quantum physics, radioactive decay

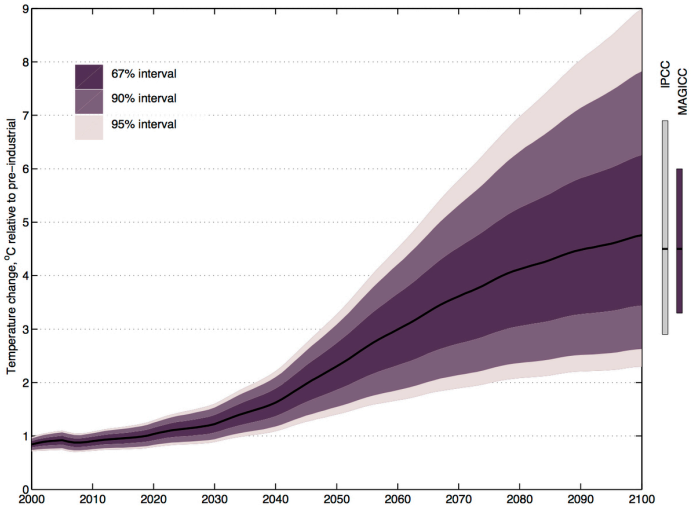
Uncertainty in Modern Life



Source: National Hurricane Center, USA

Predicted storm path with uncertainty cones.

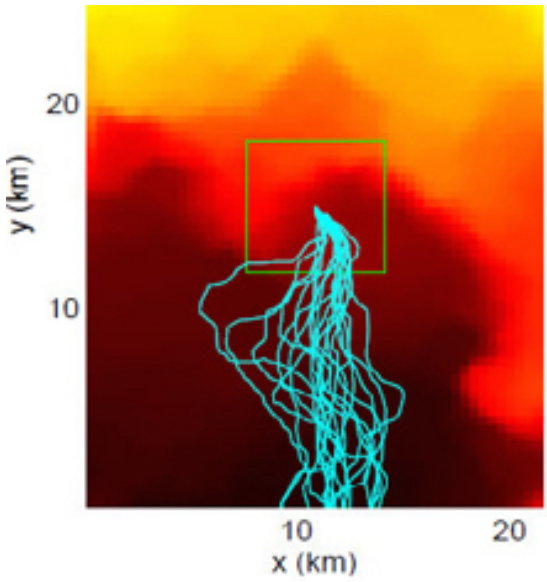
Uncertainty in Modern Life



Source: Brodman & Karoly, 2013

Global-mean temperature change for a business-as-usual emission scenario, relative to pre-industrial. Black line: median, shaded regions 67% (dark), 90% (medium) and 95% (light) confidence intervals.

Uncertainty in Modern Life



Source: K. A. Cliffe, 2012

Sample paths of groundwater-borne contaminant particles emanating from an underground radioactive waste disposal site.

Examples

Radioactive decay

- Radium-226: half-life of 1602 years
- Decays into Radon gas (Radon-222) by emitting alpha particles.
- Over a period of 1602 years, half the radium atoms in a given sample will decay.
- But we cannot say which half!

This kind of uncertainty seems to be "built into" the physical world.

Examples

Rolling dice

- Cube, 6 faces, numbered 1–6
- One or more thrown onto a table.
- For "fair dice", expect to see the numbers 1–6 appear equally often, provided the dice are thrown sufficiently many times.

How does this differ from radioactive decay?

Is this uncertainty also "built into" the physical world, or are we just not able to calculate what will happen when the dice are thrown?

Examples

Screening/testing for disease

- Incidence of disease among general population: 0.01 %
- Test has true positive rate (*sensitivity*) of 99.9 %.
- Same test has true negative rate (*specificity*) of 99.99 %.
- What is the chance that someone who tests positive actually has the disease?

Answer (using relative/conditional probabilities, Bayes' formula):

$$\begin{aligned}\mathbb{P}(\text{diseas}|\text{pos}) &= \frac{\mathbb{P}(\text{pos}|\text{diseas})\cdot\mathbb{P}(\text{diseas})}{\mathbb{P}(\text{pos}|\text{diseas})\cdot\mathbb{P}(\text{diseas})+\mathbb{P}(\text{pos}|\text{no diseas})\cdot\mathbb{P}(\text{no diseas})}\\ &= \frac{0.999\cdot 0.0001}{0.999\cdot 0.0001+(1-0.9999)\cdot(1-0.0001)} \approx 0.4998\end{aligned}$$

Frequentist View vs. Conditional Probabilities

Alternative answer (using natural frequencies):

- Think of random sample of 10,000 people.
- Of these, on average 1 will have the disease, 9,999 will not.
- Person who has the disease will almost certainly test positive.
- on average 1 of the 9,999 healthy people will test (falsely) pos.
- Thus, (roughly) only one out of every two positive patients actually has the disease.

In [Gigerenzer, 1996] medical practitioners were given the following information regarding mammography screenings for breast cancer:

incidence: 1 %; sensitivity: 80 %; specificity: 90 %.

When asked to quantify probability of a patient actually having breast cancer given a positive screening result (7.5%), 95 out of 100 physicians estimated this probability to lie above 75%.

Frequentist View vs. Conditional Probabilities

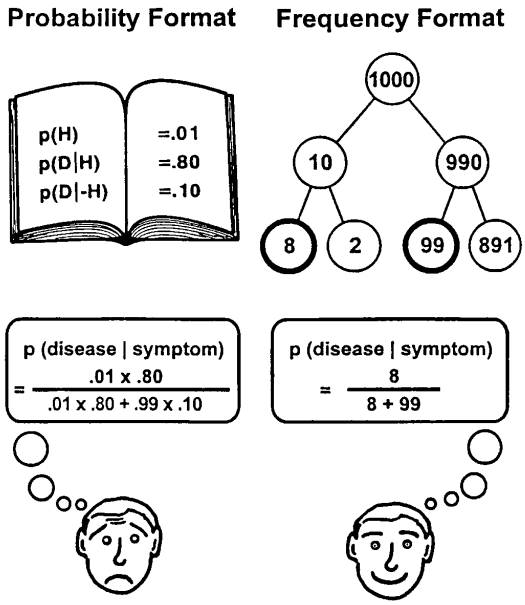


FIGURE 1. Bayesian computations are simpler when information is represented in a frequency format (right) than when it is represented in a probability format (left). p(H) = prior probability of hypothesis. H (breast cancer), p(D|H) = probability of data D (positive test) given H, and p(D|−H) = probability of D given −H (no breast cancer).

We see how crucial it is for its transparent communication how uncertainty is described.

Source: Gigerenzer, 1996

More Examples

Modeling biological systems

- From one view, biology is just very complicated physics and chemistry.
- But even the simplest biological systems are far too complicated to be understood from basic principles at the moment.
- Models are constructed that attempt to capture the essential features of what is happening, but often there are competing models and they may all fail in some way or other to predict the observed phenomena.
- In short, we don't really know what the model is!

How does this situation differ from the previous two?

More Examples

Unknown unknowns

- Obviously can't give a current example.
- Good example is the state of Physics at end of 19th century.

  There is nothing new to be discovered in physics now. All that remains is more and more precise measurement.

  Lord Kelvin, 1900

- Quantum mechanics and relativity theory were unknown unknowns.

It is easy to underestimate uncertainty.

Political Implications

Questions:[1]

1. How do we account for all the uncertainties in the complex models and analyses that inform decision makers?
2. How can those uncertainties be communicated simply but quantitatively to decision makers?
3. How should decision makers use those uncertainties when combining scientific evidence with more socio-economic considerations?
4. How can decisions be communicated so that the proper acknowledgment of uncertainty is transparent?

[1] posed on entry at the 2006 UK EPSRC Ideas Factory on the topic *Scientific Uncertainty and Decision Making for Regulatory and Risk Assessment Purposes*.

Communicating the Results

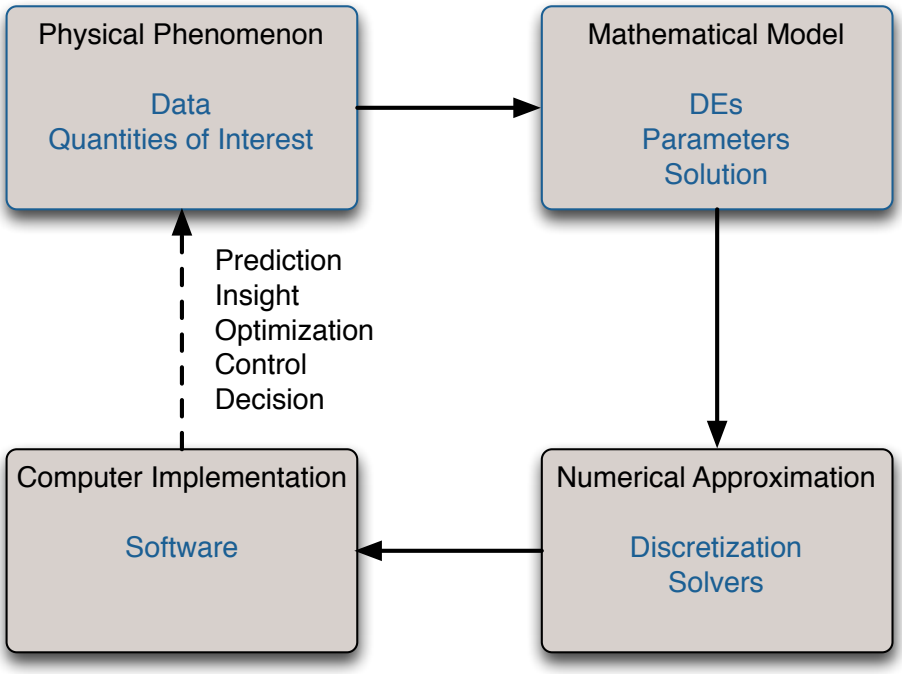
Climate change

> *The weight of evidence makes it clear that climate change is a real and present danger. The Exeter conference was told that whatever policies are adopted from this point on, the Earth's temperature will rise by 0.6F within the next 30 years. Yet those who think climate change just means Indian summers in Manchester should be told that the chances of the Gulf stream - the Atlantic thermohaline circulation that keeps Britain warm - shutting down are now thought to be greater than 50%.*
>
> *The Guardian, 2005*

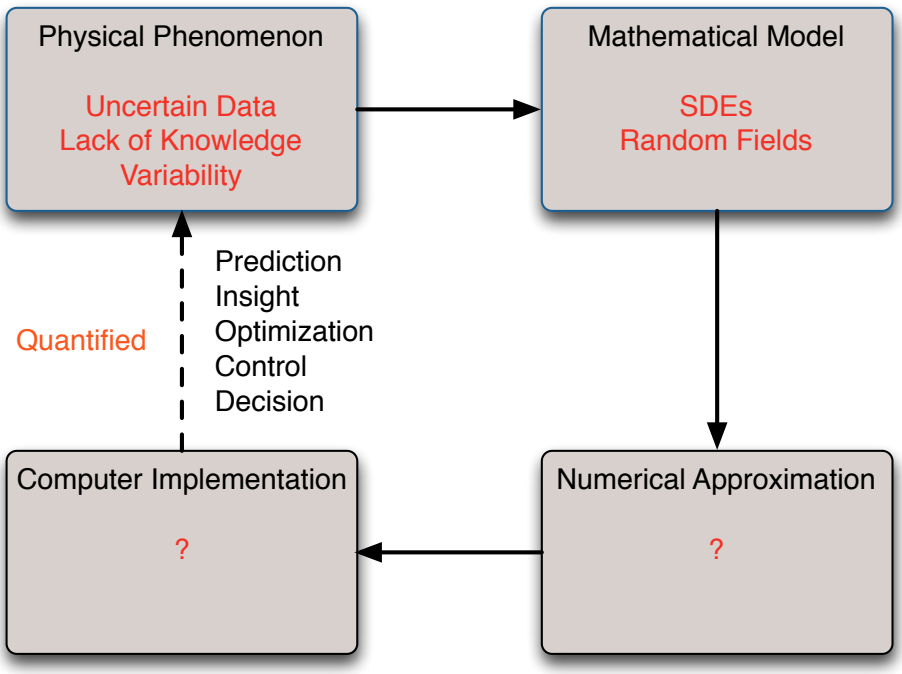
> *Most of the observed increase in globally-averaged temperatures since the mid-20th century is very likely due to the observed increase in anthropogenic GHG concentrations. It is likely there has been significant anthropogenic warming over the past 50 years averaged over each continent (except Antarctica).*
>
> *IPCC Fourth Assessment Summary for Policymakers*

What do these statements mean?

UQ and the Scientific Computing Paradigm



UQ and the Scientific Computing Paradigm



Key Tools: Efficient methods for **High-dimensional Approximation** !

Validation and Verification

What confidence can be assigned to a computer prediction of complex phenomena?

Validation: Determination of whether a mathematical model adequately represents physical/engineering phenomenon under study.
"Are we solving the right problem?"

Is this even possible? (cf. Carl Popper)

Verification: Determination of whether an algorithm and/or computer code correctly implements a given mathematical model.
"Are we solving the problem correctly?"

- code verification (software engineering)
- solution verification (a posteriori error estimation)

Aleatoric and Epistemic Uncertainty

Aleatoric: Uncertainty due to true intrinsic variability; cannot be reduced by additional experimentation, improvement of measuring devices, better model, etc.

Examples:

- rolling a die
- wind stress on a structure
- production variations

Epistemic: Uncertainty due to lack of knowledge or incomplete information.

Examples:

- turbulence modeling assumptions
- surrogate chemical kinetics
- probability distribution of a random quantity

Note: This distinction is not always meaningful or even possible.

3. Stochastic Modelling and Parametric PDEs

Stochastic Modelling

Many reasons for stochastic modelling (not all strictly UQ):

- **lack of data** (e.g. data assimilation for weather prediction)
- **data uncertainty** (e.g. uncertainty quantification in subsurface flow)
- **parameter identification** (e.g. Bayesian inference in engineering)
- **unresolvable scales** (e.g. atmospheric dispersion modelling)
- **high dimensionality** (e.g. stochastic simulation in systems biology)

Input: best knowledge about system (PDE), statistics of input parameters, measured ouput data with error statistics,...

Output: statistics of QoIs or of entire state space

often very sparse (or no) output data $\rightarrow$ need a good physical model!

- Data assimilation in NWP: data misfit, rainfall at some location
- Radioactive waste disposal: flow at repository, 'breakthrough' time
- Oil reservoir simulation: production rate
- Atmospheric dispersion: amount of ash over Heathrow
- Aeronautical engineering: certification of carbon fibre composite wing

The "Fruit Fly" of UQ

The most popular **model problem** in the UQ community is the steady-state diffusion problem with uncertain coefficient function:

$$-\nabla\cdot(a\nabla u) = f \qquad \text{on domain} \quad D \subset \mathbb{R}^d.$$

(an elliptic partial differential equation)

Rather than the PDE solution u (pressure, temperature, concentration, ...) we are typically interested in a functional Q of the solution. Such a functional is known as a quantity of interest (QoI).

Examples:

$$Q(u) = u(\mathbf{x}_0), \qquad Q(u) = \frac{1}{|D_0|}\int_{D_0} u(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

In what way might uncertainty in the coefficient a be addressed?

Worst Case Analysis

Introduce an ϵ-ball around a given function a_0 (in a suitable norm).

Examples:

$$S := \begin{cases} \{a \in C^0(D) : \|a - a_0\|_\infty \le \epsilon\}, \\ \{a \in C^1(D) : \|\nabla(a - a_0)\|_\infty \le \epsilon\}, \\ \{a \text{ constant in } D : |a - a_0| \le \epsilon\}. \end{cases}$$

Worst case analysis: determine uncertainty interval

$$I = [\inf_{a\in S} Q(u(a)), \sup_{a\in S} Q(u(a))].$$

The uncertainty range of Q is then the length of I.

This is a generalisation of interval analysis.

Probabilistic Model

But: In general, some coefficients $a \in S$ are more likely than others.

Probabilistic approach:

- Introduce probability measure on S.
- $Q(u(\cdot))$ as a (measurable) mapping from S to the output set $\{Q(u(a)) : a \in S\}$ induces a probability measure for the QoI. ("uncertainty propagation")
- **Big issue:** choice of distribution, too much subjective information?
- Some classical guidelines: Laplace's principle of insufficient reason, maximum entropy, etc.
- Choosing distribution based on data is point of departure for Bayesian inference (genuine "uncertainty quantification").

Other Models

- Evidence theory (generalisation of probabilistic model)
- Fuzzy sets (deterministic approach introduced by [Zadeh, 1965])
- Possibility theory
- Scenario analysis
- ...

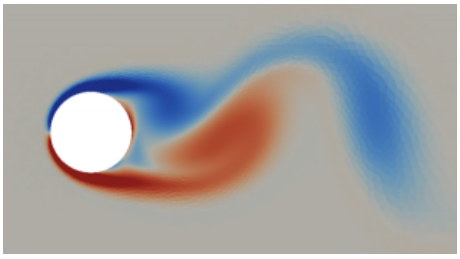
For the remainder we will focus on the probabilistic approach.

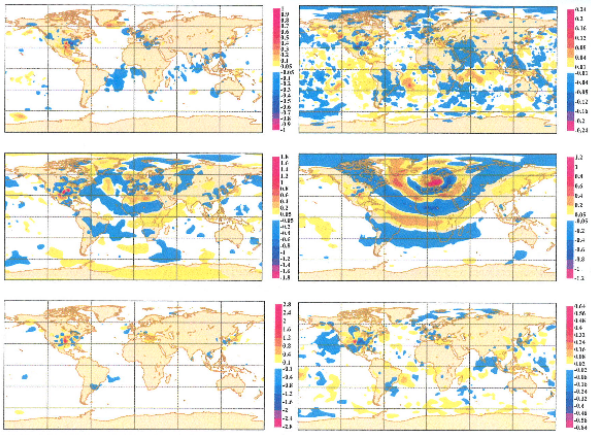
PDEs with Random Coefficients – Examples

- **Navier–Stokes** (e.g. flow around wing, weather forecasting):

$$\rho(\omega)\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = -\nabla p + \mu(\omega)\nabla^2\mathbf{v} + \mathbf{f}(x,\omega) \quad \text{in} \quad \Omega(\omega)$$

$$\text{subject to IC} \quad v(x,0) = v_0(x,\omega) \quad + \quad \text{BCs}$$





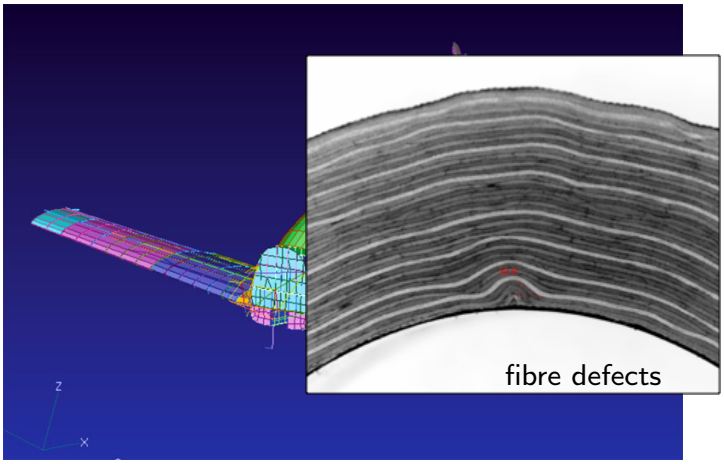
uncertain ICs → **data assimilation**

PDEs with Random Coefficients – Examples

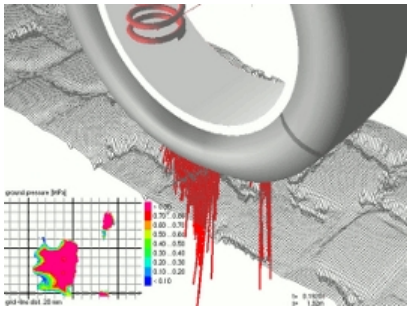
- **Structural Mechanics** (e.g. composites, tires or bone):

$$\nabla\cdot\left(\overline{\overline{C}}(x,\omega) : \frac{1}{2}\left[\nabla\mathbf{u} + \nabla\mathbf{u}^T\right]\right) + \mathbf{F}(x,\omega) = 0 \quad \text{in} \quad \Omega(\omega)$$

subject to BCs



contact on rough surface



PDEs with Random Coefficients – Examples

- **Neutron Transport**:

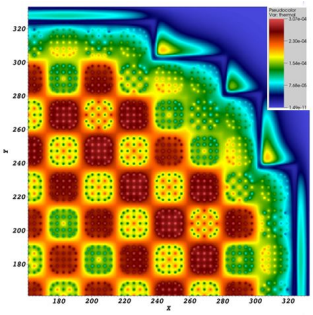
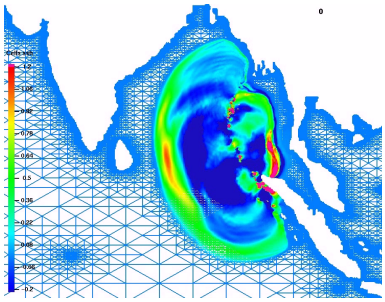


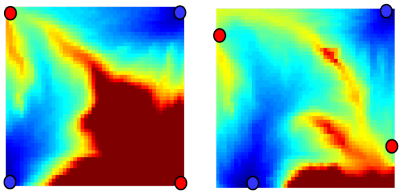
Figure 1 2D Slice of Thermal Flux Distribution near the Core Mid-plane

- **Tsunami Simulation**:



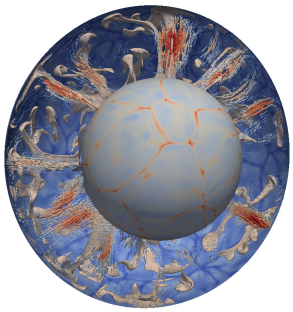
[Behrens et al]

- **Oil Reservoir Simulation**:



optimal well placement

- **Mantel Convection**:



[Gmeiner et al]

Stochastic Differential Equations (SDEs)

Atmospheric Dispersion (e.g. volcanic ash, radionuclides, ...)

Given large-scale atmospheric flow $\vec{v}(\vec{x},t)$, model turbulent dispersion of particles by a system of SDEs:

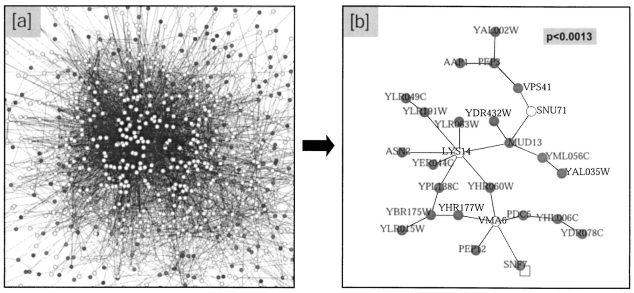
$$d\vec{U} = a(\vec{U},\vec{X},t)dt + b(\vec{X},t)d\vec{W}(t)$$
$$d\vec{X} = \big(\vec{v}(\vec{X},t) + \vec{U}(\vec{X},t)\big)dt$$

$\vec{U}(t)\ldots$ turbulent correction; $\vec{X}(t)\ldots$ particle position; $\vec{W}(t)\ldots$ Brownian motion

Similar models appear in **mathematical finance**. We will come back to that later!

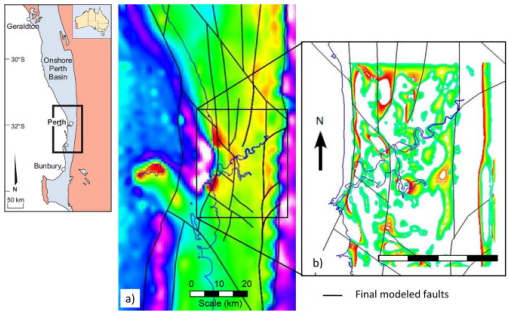
Stochastic Reaction Networks and Imaging

Gene Regulatory Networks
(direct stochastic simulation)



Source: Shannon et al, 2003

Geostatistics
(and other imaging applications)



Source: Corbel, Wellmann, 2015

4. A Case Study: Radioactive Waste Disposal

A Case Study: Radioactive Waste Disposal

- An area where UQ has played a central role in the past 25 years is the assessment of strategies and sites for the long-term storage of radioactive waste.
- Uncertainties arise from technological complexity as well as the long time scales to be considered.
- Many leading industrial countries (USA, UK, Germany) have scrapped previous plans for national long-term disposal sites and are re-evaluating their strategies.
- Consider a basic UQ problem which occurs in site assessment.

Background

- Radioactive waste is produced mainly by nuclear power plants (Other sources: medical, weapons, non-nuclear industries)
- Exposure to high radiation levels seriously harmful to humans and animals; long-term exposure to low-level radiation can cause cancer and other long-term health problems.
- Classification of waste "level":
  - *high (HLW)*: highly radioactive, produces heat, small amount
  - *intermediate (ILW)*: still very radioactive, no heat produced
  - *low (LLW)*: low radioactivity; packaging material, protective clothing, soil, concrete that has been exposed to radioactivity
- Quantities in storage (excl. LLW; source: `http://newmdb.iaea.org`)
  - Germany: 120,000 m^3 (2007)
  - France: 90,000 m^3 (2007)
  - UK: 350,000 m^3 (2007)
  - USA: 540,000 m^3 (2008)

Management Options

Since this problem has received serious consideration ($\approx$ 1970s), several options have been discussed

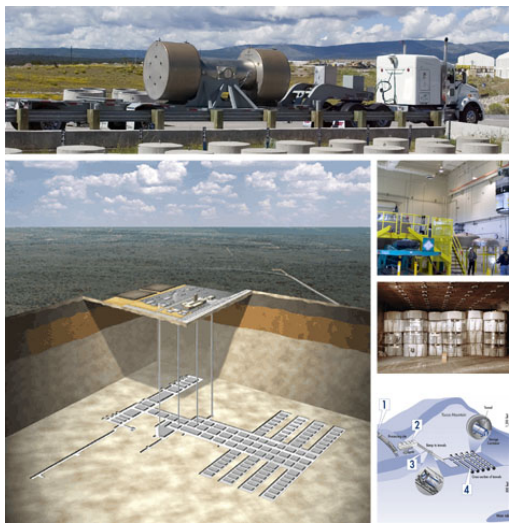
- Surface storage: current universal solution, not long-term, risky.
- Disposal at sea: banned by international treaty (London Convention)
- Disposal in space: too dangerous, prohibitive cost (but permanent)
- Transmutation: not yet proven, would mitigate but not solve problem
- Deep geological disposal: favoured by nearly all countries with a radioactive waste disposal programme

Deep Geological Disposal

- Storage in containers in tunnels, several hundred meters deep, in stable geological formations.
- Issue: retrievable or not?
- No human intervention required after final closure of repository.
- Several barriers: chemical, physical, geological.
- Substantial engineering challenge
  (containment must be assured for at least 10,000 years).
- Main escape route for radionuclides: groundwater pathway.
- **Assessing safety** of potential sites of utmost importance
  long timescales $\rightarrow$ **modelling essential!**
- **Key aspect:** How to quantify uncertainties in the models?

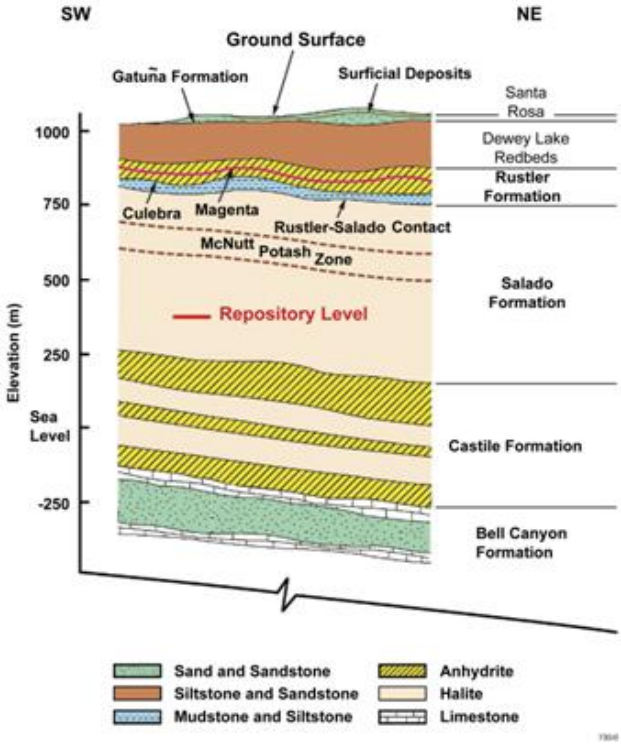
WIPP – Waste Isolation Pilot Plant

- US DOE repository for radioactive waste situated near Carlsbad, NM.
- Fully operational since 1999.
- Extensive site characterisation and performance assessment since 1976, also in course of compliance certification and recertification by US EPA (every 5 years).
- Lots of publicly available data.
- `http://www.wipp.energy.gov`



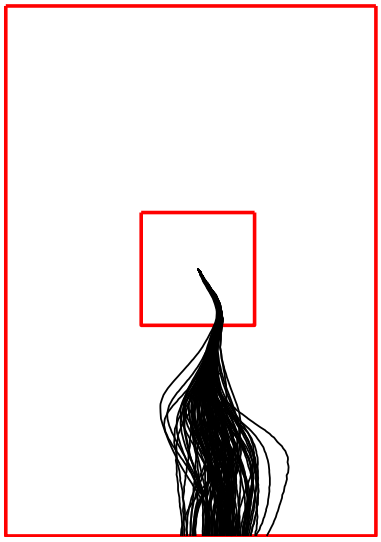
WIPP Geology

- Repository located at 655m depth in bedded evaporites (primarily halite, a salt).
- Most transmissive rock layer in the region is the Culebra Dolomite.
- In the event of an accidental breach, Culebra would be the principal pathway for transport of radionuclides away from the repository.



WIPP UQ Scenario

- One scenario at WIPP is a release of radionuclides by means of a borehole drilled into the repository.
- Radionuclides are released into the Culebra Dolomite and then transported by groundwater.
- Travel time from release point in the repository to the boundary of the region is an important quantity.
- To a good approximation the flow is two-dimensional.

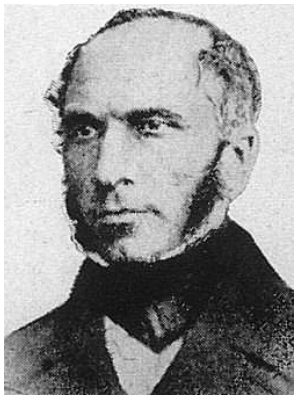


Darcy's Law

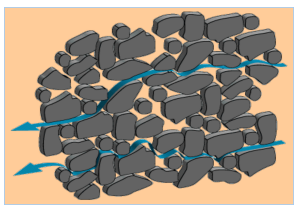
- The simplest mathematical model for flow through a porous medium (e.g. groundwater through an aquifer) is given by Darcy's Law:

$$\mathbf{q} = \frac{-\mathbf{k}}{\mu}\nabla p,$$

in which $\mathbf{q}$ is the volumetric flux or Darcy velocity (discharge per unit area in [m/s]), $\mathbf{k}$ is the permeability tensor, a material parameter describing how easily water flows through the medium, μ is the dynamic viscosity of the fluid and p is the hydraulic head (pressure) of the fluid.



- The hydraulic conductivity tensor is defined as $K := \mathbf{k}\rho g/\mu$, where g is the acceleration due to gravity and ρ the fluid density.
- The actual pore velocity with which the fluid particles move through the pores is obtained as $\mathbf{u} = \mathbf{q}/\phi$, where $\phi \in [0, 1]$ denotes the porosity of the medium.



Groundwater Flow Model

Stationary Darcy flow	$\mathbf{q} = -K\nabla p$	$\mathbf{q}$: Darcy flux
		K : hydraulic conductivity
		p : hydraulic head
mass conservation	$\nabla\cdot\,\mathbf{u} = 0$	$\mathbf{u}$: pore velocity
	$\mathbf{q} = \phi\mathbf{u}$	ϕ : porosity
transmissivity	$k = Kb$	b : aquifer thickness
particle transport	$\dot{\mathbf{x}}(t) = -\frac{k(\mathbf{x})}{b\phi}\nabla p(\mathbf{x})$	$\mathbf{x}$: particle position
	$\mathbf{x}(0) = \mathbf{x}_0$	$\mathbf{x}_0$: release location

Quantity of interest: $\log_{10}$ of particle travel time to reach boundary

UQ Problem – PDE with Random Coefficient

Primal form of Darcy equations (our "fruit fly"):

$$-\nabla\cdot[k(\mathbf{x})\nabla p(\mathbf{x})] = 0, \quad \mathbf{x}\in D, \qquad p = p_0 \text{ along } \partial D.$$

Model k as a random field (RF) $k = k(\mathbf{x},\omega)$, $\omega\in\Omega$, with respect to underlying probability space $(\Omega,\mathcal{A},\mathbb{P})$.

Modeling Assumptions (standard in 2D hydrogeology):

- T has finite mean and covariance

$$\overline{k}(\mathbf{x}) = \mathbb{E}\left[k(\mathbf{x},\cdot)\right], \qquad \mathbf{x}\in D,$$
$$\mathbf{Cov}_k(\mathbf{x},\mathbf{y}) = \mathbb{E}\left[\left(k(\mathbf{x},\cdot)-\overline{k}(\mathbf{x})\right)\left(k(\mathbf{y},\cdot)-\overline{k}(\mathbf{y})\right)\right], \qquad \mathbf{x},\mathbf{y}\in D.$$

- k is lognormal, i.e., $Z(\mathbf{x},\omega) := \log k(\mathbf{x},\omega)$ is a Gaussian RF.
- $\mathbf{Cov}_Z$ is stationary and isotropic, i.e., $\mathbf{Cov}_Z(\mathbf{x},\mathbf{y}) = c(\|\mathbf{x}-\mathbf{y}\|_2)$

Matérn Family of Covariance Kernels

$$c(\mathbf{x},\mathbf{y}) = c_{\boldsymbol{\theta}}(r) = \frac{\sigma^2}{2^{\nu-1}\,\Gamma(\nu)} \left(\frac{2\sqrt{\nu}\,r}{\lambda}\right)^{\nu} K_\nu \left(\frac{2\sqrt{\nu}\,r}{\lambda}\right), \quad r = \|\mathbf{x}-\mathbf{y}\|_2$$

K_ν : modified Bessel function of order ν

Parameters $\boldsymbol{\theta} = (\sigma^2, \lambda, \nu)$

σ^2 : variance

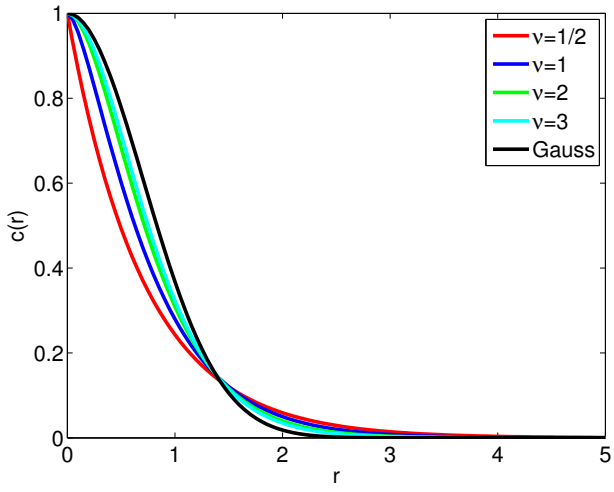
λ : correlation length

ν : smoothness parameter

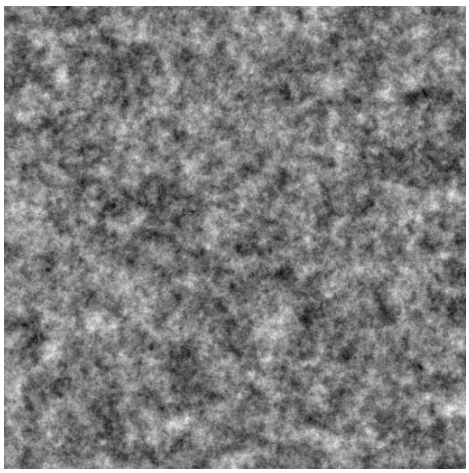
Special cases:

$\nu = \frac{1}{2}$:	$c(r) = \sigma^2 \exp(-\sqrt{2}r/\lambda)$	exponential covariance
$\nu = 1$:	$c(r) = \sigma^2 \left(\frac{2r}{\lambda}\right) K_1 \left(\frac{2r}{\lambda}\right)$	Bessel covariance
$\nu \to \infty$:	$c(r) = \sigma^2 \exp(-r^2/\lambda^2)$	Gaussian covariance

Matérn Covariance Functions



$\sigma^2 = 1,\ \lambda = 1$



$\sigma^2 = 8,\ \lambda = 1/64,\ \nu = 1/2$

Smoothness: Realisations $Z(\cdot,\omega) \in C^\eta(D)$ (Hölder), for any $\eta < \nu$.

Sampling from Z – Karhunen-Loève Expansion

Since $c(\mathbf{x}, \mathbf{y})$ is symmetric, positive semidefinite, continuous, the covariance operator

$$C : L^2(D) \to L^2(D), \qquad (Cu)(\mathbf{x}) = \int_D u(\mathbf{y}) c(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{y}$$

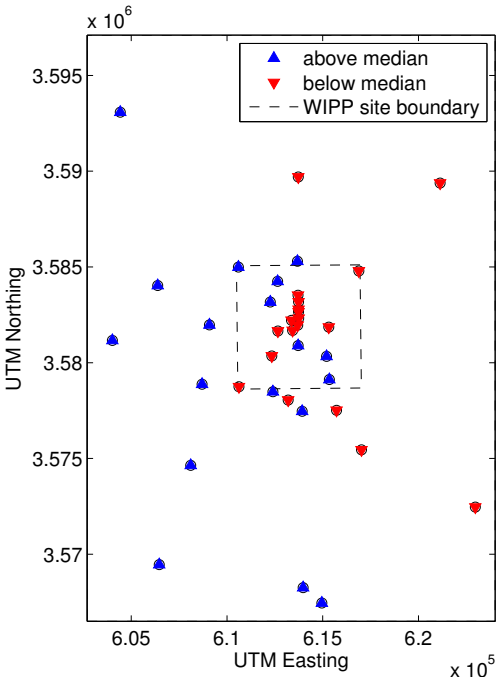
is selfadjoint, compact, nonnegative. Hence, its eigenvalues $\{\mu_m\}_{m \in \mathbb{N}}$ form a non-increasing sequence accumulating at most at 0.

Karhunen-Loève expansion (converges in $L^2_{\mathbb{P}}(\Omega; L^\infty(D))$):

$$Z(x, \omega) = \overline{Z}(\mathbf{x}) + \sum_{m=1}^{\infty} \sqrt{\mu_m} \, \phi_m(\mathbf{x}) \, Y_m(\omega)$$

where $\{\phi_m\}_{m \in \mathbb{N}}$ are normalised eigenfunctions and $Y_m \sim N(0, 1)$ i.i.d.

WIPP Data



- transmissivity measurements at 38 test wells
- use head measurements to obtain boundary data via statistical interpolation (kriging)
- constant layer thickness $b = 8$m
- constant porosity $\phi = 0.16$
- SANDIA Nat. Labs reports [Caufman et al., 1990]
  [La Venue et al., 1990]

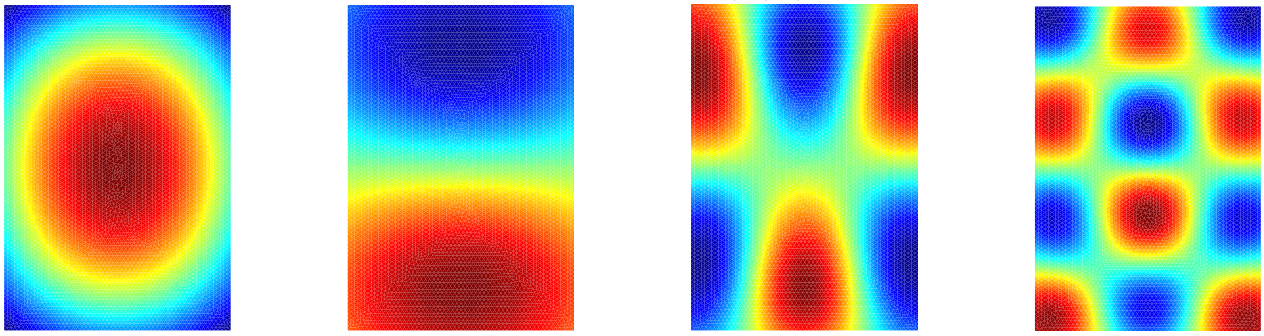
Probabilistic Model of Transmissivity

Calibrate statistical model to the transmissivity data:
e.g. [Ernst et al., 2014]

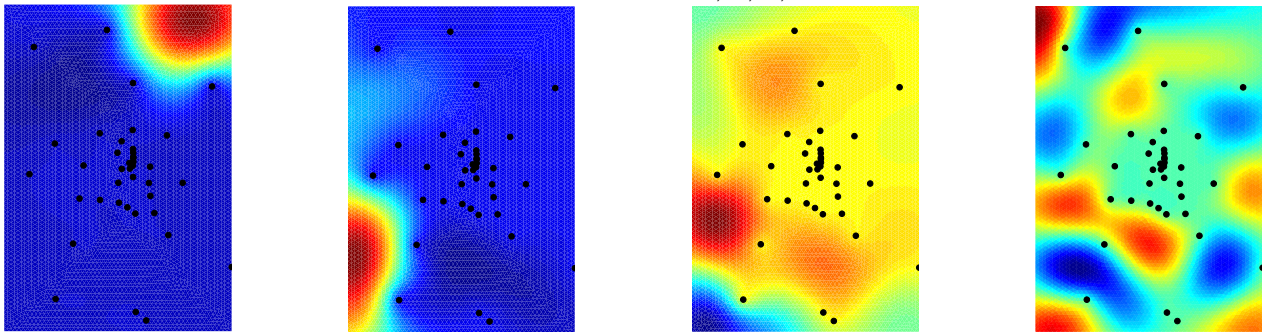
1. Estimate parameters σ, λ and ν via restricted maximum likelihood estimation (REML).
2. Condition resulting covariance structure of $Z = \log k$ on transmissivity measurements. (Low-rank modification of covariance operator.)
3. Approximate Z by truncated Karhunen-Loève expansion, i.e use only the leading s terms.

WIPP KL Modes – conditioned on 38 transmissivity observations

unconditioned, $m = 1, 2, 9, 16$



conditioned, $m = 1, 2, 9, 16$



5. Computational Challenges

Computational Challenges

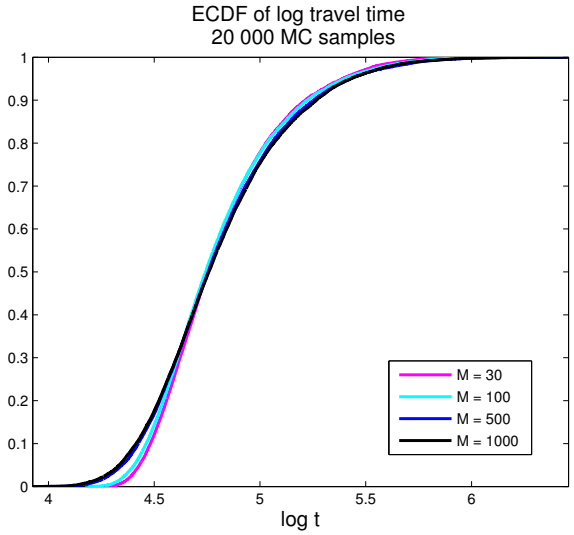
Simulating PDEs with Highly Heterogeneous Random Coefficients:

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

- **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 focus of this course!):
  - Monte Carlo, Quasi-Monte Carlo
  - Sparse Grids & stochastic Galerkin/collocation
- **Solve** large number of **multiscale** deterministic PDEs:
  - Efficient discretisation & FE error analysis (mesh size h)
  - Multigrid Methods, AMG, DD Methods

Why is it computationally so challenging?

- Low regularity (global): $k \in C^\eta,\ \eta < \nu < 1$ (Hölder) $\implies$ **fine** mesh $h \ll 1$
- Large σ^2 & exponential $\implies$ **high** contrast $k_{\max}/k_{\min} > 10^6$
- Small λ $\implies$ **multiscale** + **high stochastic dimension** $s > 100$



Source: Ernst et al, 2014 ($s = M$)

Standard Monte Carlo Quadrature

$$\underset{\text{random input}}{\mathbf{Y}(\omega) \in \mathbb{R}^s} \xrightarrow{\text{Model}(h)} \underset{\text{state vector}}{\mathbf{P}(\omega) \in \mathbb{R}^{M_h}} \xrightarrow{\text{Output}} \underset{\text{quantity of interest}}{Q_{h,s}(\omega) \in \mathbb{R}}$$

- **Here:** $\mathbf{Y}$ multivariate Gaussian for KL expansion; $\mathbf{P}$ numerical PDE solution; $Q_{h,s}$ a (non)linear functional of $\mathbf{P}$
- Real QoI $Q(\omega)$ inaccessible (exact PDE), but we can assume $\mathbb{E}[Q_{h,s}] \overset{h\to 0,\ s\to\infty}{\longrightarrow} \mathbb{E}[Q]$ and $|\mathbb{E}[Q_{h,s} - Q]| = \mathcal{O}(h^\alpha) + \mathcal{O}(s^{-\alpha'})$
- **Standard Monte Carlo** estimator for $\mathbb{E}[Q]$: More detail below!

$$\hat{Q}^{\text{MC}} := \frac{1}{N} \sum_{i=1}^{N} Q_{h,s}^{(i)}$$

where $\{Q_{h,s}^{(i)}\}_{i=1}^N$ are i.i.d. samples computed with Model(h)

- Cost per sample is $\mathcal{O}(M_h^\gamma)$ (optimal: $\gamma = 1$)

Standard Monte Carlo Quadrature

- Convergence of plain vanilla MC **(mean square error)**:

$$\underbrace{\mathbb{E}\big[\big(\hat{Q}^{\mathrm{MC}} - \mathbb{E}[Q]\big)^2\big]}_{=:\,\mathrm{MSE}} = \mathbb{V}[\hat{Q}^{\mathrm{MC}}] + \big(\mathbb{E}[\hat{Q}^{\mathrm{MC}}] - \mathbb{E}[Q]\big)^2$$

$$= \underbrace{\frac{\mathbb{V}[Q_{h,s}]}{N}}_{\text{sampling error}} + \underbrace{\Big(\mathbb{E}[Q_{h,s} - Q]\Big)^2}_{\text{model error ("bias")}}$$

- Typical: $\alpha = 1 \;\Rightarrow\; \mathrm{MSE} = \mathcal{O}(N^{-1}) + \mathcal{O}(h^2) \le \mathrm{TOL}^2$, and so $h \sim \mathrm{TOL}$ and $N \sim \mathrm{TOL}^{-2}$.
- Using optimal PDE solver: $\mathrm{Cost} = \mathcal{O}(Nh^{-d}) = \mathcal{O}(\mathrm{TOL}^{-(d+2)})$

  (e.g. for $\mathrm{TOL} = 10^{-3}$: $h \sim 10^{-3}$, $N \sim 10^6$ and $\mathrm{Cost} = \mathcal{O}(10^{12})$ in 2D!!)

Quickly becomes **prohibitively expensive !**

Numerical Experiment with standard Monte Carlo

$D = (0,1)^2$, unconditioned KL expansion, $Q = \| - k\frac{\partial p}{\partial x_1}\|_{L^1(D)}$ using mixed FEs and the AMG solver `amg1r5` [Ruge, Stüben, 1992]

- Numerically observed FE-error: $\approx \mathcal{O}(h^{3/4}) \implies \alpha \approx 3/4$.
- Numerically observed cost/sample: $\approx \mathcal{O}(h^{-2}) \implies \gamma \approx 1$.
- **Total cost** to get RMSE $\mathcal{O}(\mathrm{TOL})$: $\approx \mathcal{O}(\mathrm{TOL}^{-14/3})$

  to get error reduction by a factor 2 → cost grows by a factor 25!

Case 1: $\sigma^2 = 1$, $\lambda = 0.3$, $\nu = 0.5$

TOL	h^{-1}	N_h	Cost
0.01	129	1.4×10^4	21 min
0.002	1025	3.5×10^5	30 days

Case 2: $\sigma^2 = 3$, $\lambda = 0.1$, $\nu = 0.5$

TOL	h^{-1}	N_h	Cost
0.01	513	8.5×10^3	4 h
0.002	**Prohibitively large!!**		

(actual numbers & CPU times on a cluster of 2GHz Intel T7300 processors)

Alternatives – The Curse of Dimensionality

- **Polynomial quadrature: stochastic Galerkin/collocation methods**
  - Cost grows very fast with dimension s & polynomial order q
    $\rightarrow$ #stochastic DOFs $N_{\mathsf{SC}} = \mathcal{O}\left(\frac{(s+q)!}{s!q!}\right)$ **(faster than exponential!)**
  - Lower number with **sparse grids** (Smolyak), but **still exponential** growth with s!

The **"Curse of Dimensionality"**

  - Anisotropic sparse grids or adaptive best N-term approximation can be dimension independent with **sufficient smoothness**! More detail below!

- **Monte Carlo type methods**
  - Convergence of plain vanilla Monte Carlo is **always dimension independent** ! (No smoothness needed!) **BUT** (as shown) the order is too slow: $\mathcal{O}(N^{-1/2})$!
  - **Quasi-Monte Carlo** can also be dimension independent and (almost) $\mathcal{O}(N^{-1})$! **But** requires also **(some) smoothness** ! More detail below!
  - **Classical** quasi-MC methods (e.g. latin hypercube) have cost $\mathcal{O}(N^{-1}(\log N)^s)$! (**"Curse of Dimensionality"**)

6. Short Recap on Polynomial Approximation & Quadrature in one Dimension

Interpolation & Best Approximation

Given $y_0, \ldots, y_n$, the values of a function $f(x)$ at $x_0 < \ldots < x_n \in [a, b]$, as well as a class of approximating functions P, e.g. polynomials.

Interpolation. We say the function $g \in P$ *interpolates* the unknown function f if

$$g(x_i) = y_i\,, \quad \text{for all } \; i = 0, \ldots, n.$$

Best Approximation. We say the function $g \in P$ is the *best approximation* of the unknown function f in P with respect to the norm $\|\cdot\|$ if

$$\|f - g\| \;=\; \min_{h \in P} \|f - h\|.$$

The following is based on my lecture notes from **Numerik 0 – Einführung in die Numerik** from WS 2018/19.

Polynomials: Uniqueness and Interpolation Error

Interpolation.

Let $\|f\|_{\infty,[a,b]}$ be the uniform (maximum) norm $\|f\|_{\infty,[a,b]} := \max_{x\in[a,b]} |f(x)|$.

(This can be generalised for discontinuous functions to $\|f\|_{\infty,[a,b]} := \operatorname{ess\,sup}_{x\in[a,b]} |f(x)|$.)

Theorem 6.1 (Numerik 0, Satz 2.8 & Korollar 2.13)

Let $P = \mathbb{P}_n$. Then there exists a unique interpolant $p_n \in \mathbb{P}_n$ and

$$\|f - p_n\|_{\infty,[a,b]} \;\le\; \frac{|b-a|^{n+1}}{(n+1)!} \|f^{(n+1)}\|_{\infty,[a,b]}$$

Best Approximation.

Theorem 6.2 (Numerik 0, Satz 2.26)

Let $P = \mathbb{P}_n$ and $f \in C[a, b]$. Then there exists a unique best approximating polynomial $p_n \in \mathbb{P}_n$ (the norm can be arbitrary here) such that

$$\|f - p_n\| \;=\; \min_{q \in \mathbb{P}_n} \|f - q\|.$$

Interpolatory Quadrature Rules

Based on integrating the interpolating polynomial $p_n \in \mathbb{P}_n$, i.e.

$$\int_a^b f(x)\,\mathrm{d}x \;\approx\; Q_n(f) := \int_a^b p_n(x)\,\mathrm{d}x = \sum_{i=0}^{n} w_i f(x_i)$$

with $w_i := \int_a^b L_i^{(n)}(x)\,\mathrm{d}x$.

Newton-Cotes Rules.

Based on equidistant points $x_i = a + ih$, $i = 0, \ldots, n$, with $h = (b-a)/n$.

Theorem 6.3 (Proposition 3.6 & Korollar 3.11)

Newton-Cotes quadrature rules are exact for polynomials of degree n and $(n+1)$ for n odd and n even, respectively, and

$$\left| Q_n(f) - \int_a^b f(x)\,dx \right| \;\leq\; \begin{cases} C\frac{(b-a)^{n+2}}{(n+1)!}\|f^{(n+1)}\|_{\infty,[a,b]} & \text{for } n \text{ odd,} \\ C\frac{(b-a)^{n+3}}{(n+2)!}\|f^{(n+2)}\|_{\infty,[a,b]} & \text{for } n \text{ even.} \end{cases}$$

Interpolatory Quadrature Rules

Composite Newton-Cotes Rules.

Apply Newton-Cotes rule $Q_n(f)$ on subintervals $[y_{j-1}, y_j]$ of $[a,b]$ with $a = y_0 < y_1 < \ldots < y_N = b$ and **now** $h = \max_{j=1}^{N}(y_j - y_{j-1})$.

Theorem 6.4 (Korollar 3.14)

For the composite Newton-Cotes quadrature rule $Q_{n,N}(f)$, i.e. applying $Q_n(f)$ on N subintervals, we get

$$\left| Q_{n,N}(f) - \int_a^b f(x)\,dx \right| \;\leq\; C\|f^{(d+1)}\|_{\infty,[a,b]} h^{d+1}$$

where $d = n$ for n odd and $d = n+1$ for n even.

In the equidistant case $h = \frac{b-a}{N}$ and so the rate of convergence is $\mathcal{O}(N^{-(d+1)})$.

Using an error estimator, the subintervals $[y_{j-1}, y_j]$ can be chosen **adaptively**, which is particularly useful if the derivatives blow up near some points, but the function is smooth otherwise.

Interpolatory Quadrature Rules

Gauss Rules.

By choosing the quadrature points in an optimal way, rather than equidistant, the order of the quadrature rule can be significantly increased.

Theorem 6.5 (Satz 3.24 and Satz 3.27)

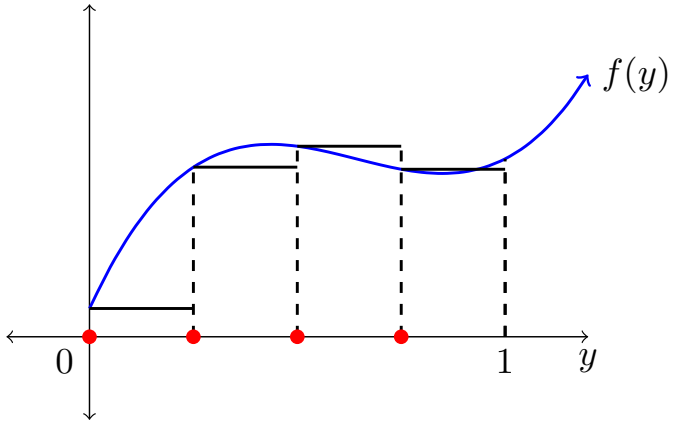
If the quadrature points $x_0, \ldots, x_n$ are the roots of the $(n+1)$th Legendre polynomial, then the interpolatory quadrature rule $Q_n(f)$ is exact for polynomials of degree $2n+1$ and

$$\left| Q_n(f) - \int_a^b f(x)\, dx \right| \leq C \left(\frac{b-a}{2} \right)^{2n+3} \frac{1}{(2n+2)!} \|f^{(2n+2)}\|_{\infty,[a,b]} .$$

Of course it is also possible to use Gauss quadrature rules in composite form. The rate of convergence is then $\mathcal{O}(N^{-(2n+2)})$.

7. The Curse of Dimensionality

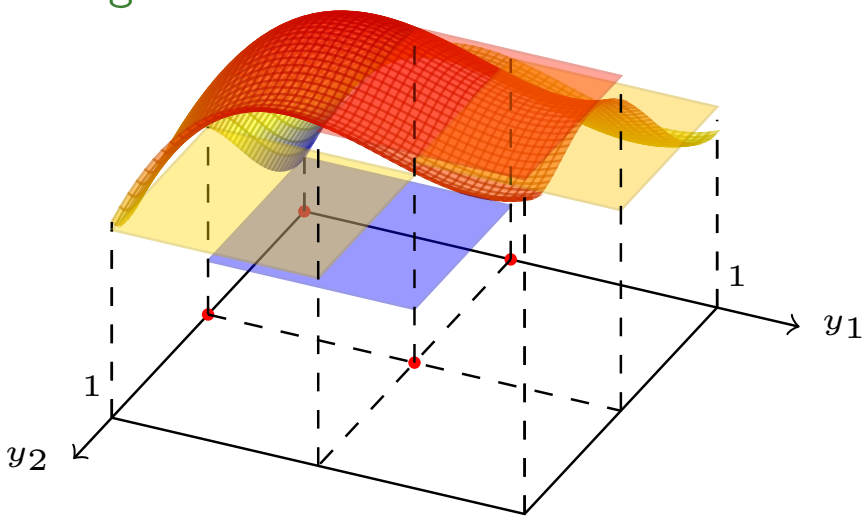
Numerical integration in one dimension



$$\int_0^1 f(y)\,\mathrm{d}y \;\approx\; \frac{1}{n}\sum_{k=0}^{n-1} f\left(\tfrac{k}{n}\right)$$

1. n function evaluations
2. if $\left|\frac{\mathrm{d}f}{\mathrm{d}y}\right| < \infty$ then error $\sim \mathcal{O}(h) = \mathcal{O}(n^{-1})$

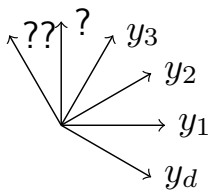
Numerical integration in two dimensions



$$\int_0^1\int_0^1 f(y_1,y_2)\,\mathrm{d}y_1\mathrm{d}y_2 \;\approx\; \frac{1}{N}\sum_{k=0}^{n-1}\sum_{\ell=0}^{n-1} f\left(\tfrac{k}{n},\tfrac{\ell}{n}\right)$$

1. $N = n^2$ function evaluations in total
2. if $\left|\frac{\partial f}{\partial y_1}\right|, \left|\frac{\partial f}{\partial y_2}\right| < \infty$ then error $\sim \mathcal{O}(h) = \mathcal{O}(N^{-1/2})$

... d dimensions — The curse of dimensionality!



$$\int_{[0,1]^d} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \approx \frac{1}{n^d} \sum_{k_1=0}^{n-1} \sum_{k_2=0}^{n-1} \cdots \sum_{k_d=0}^{n-1} f\left(\tfrac{k_1}{n}, \tfrac{k_2}{n}, \ldots, \tfrac{k_d}{n}\right)$$

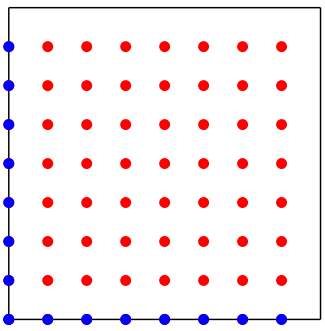
1. $N = n^d$ function evaluations in total
2. error $\sim \mathcal{O}(h) = \mathcal{O}(N^{-1/d})$

In 100 dimensions a rule using two points in each dimension needs
$2^{100} \approx 1,000,000,000,000,000,000,000,000,000,000$ function evaluations!

How to break the curse of dimensionality?

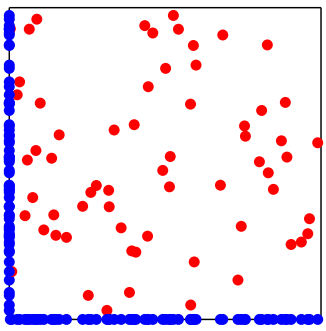
$$\int_{[0,1]^d} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \approx \frac{1}{N} \sum_{k=0}^{N-1} f(\boldsymbol{t}_k) =: Q_{N,d}(f) \quad \text{(only rules with weights } 1/N\text{)}$$

Product rules



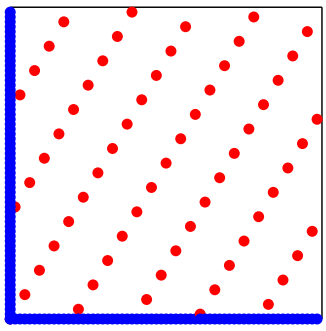
$\boldsymbol{t}_k$ on a product grid
error $\sim \mathcal{O}(N^{-1/d})$
$\|\nabla f\| < \infty$
Possible to improve with higher regularity and **"sparse grids"** (but still exponential in $1/d$).

Monte Carlo



$\boldsymbol{t}_k \sim U([0,1]^d)$ (random)
error $\sim \mathcal{O}(N^{-1/2})$
only $\|f\|_{L^2} < \infty$
independent of dimension!

Quasi–Monte Carlo



$\boldsymbol{t}_k$ chosen deterministically
error $\sim \mathcal{O}(N^{-1+\delta})$
$f \in H^1_{\text{mixed}}$ See below!
independent of dimension!

Summary of Chapter I

Monte Carlo methods do not suffer from the **curse of dimensionality**. They are **"non-intrusive"**, require **no regularity** and nonlinear parameter dependence is no problem.

But the plain vanilla version is **too slow!**

Polynomial quadrature rules (tensorised) in their basic form suffer from the **curse of dimensionality**. They may require a major software rewrite, typically require a lot of **regularity** and nonlinear parameter dependence may lead to further cost increase.

But they can potentially converge **much faster** than MC!

Alternatives?

- Accelerate Monte Carlo methods
- Sparsify polynomial rules

High-Dimensional Approximation and Applications in Uncertainty Quantification

II. (Multilevel) Monte Carlo Methods

Prof. Dr. Robert Scheichl
r.scheichl@uni-heidelberg.de

Dr. Alexander Gilbert
a.gilbert@uni-heidelberg.de

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Institut für Angewandte Mathematik, Universität Heidelberg

Summer Semester 2020

1. History – The Buffon Needle Problem

Monte Carlo

The Buffon Needle Problem

- In 1777, George Louis Leclerc, Comte de Buffon (1707–1788), French naturalist and mathematician, posed the following problem:

  Let a needle of length ℓ be thrown at random onto a horizontal plane ruled with parallel straight lines spaced by a distance $d > \ell$ from each other. What is the probability p that the needle will intersect one of these lines?

- Answer: $p = \frac{2\ell}{\pi d}$ (simple geometric arguments)
- Laplace later used similar randomised experiment to approximate π.
- The term "Monte Carlo method" was coined by Ulam, von Neumann, Metropolis in the Manhattan project (Los Alamos, 1946).

The Buffon Needle Problem

THE ROYAL SOCIETY

Ants estimate area using Buffon's needle

Eamonn B. Mallon* and Nigel R. Franks

Centre for Mathematical Biology, and Department of Biology and Biochemistry, University of Bath, Bath BA2 7AY, UK

We show for the first time, to our knowledge, that ants can measure the size of potential nest sites. Nest size assessment is by individual scouts. Such scouts always make more than one visit to a potential nest before initiating an emigration of their nest mates and they deploy individual-specific trails within the potential new nest on their first visit. We test three alternative hypotheses for the way in which scouts might measure nests. Experiments indicated that individual scouts use the intersection frequency between their own paths to assess nest areas. These results are consistent with ants using a 'Buffon's needle algorithm' to assess nest areas.

Keywords: ants; colony emigration; individual-specific pheromones; *Leptothorax*; nest sites; rules of thumb

Proceedings of the Royal Society of London, 2000

Monte Carlo Simulation for the Buffon Needle Problem

- Let $\{H_k\}_{k\in\mathbb{N}}$ denote a sequence of i.i.d. binomial random variables s.t.

$$H_k(\omega) = \begin{cases} 1 & \text{if } k\text{-th needle intersects a line,} \\ 0 & \text{otherwise.} \end{cases}$$

- Their common distribution is that of a Bernoulli trial with success probability $p = 2\ell/\pi d$. In particular: $\mathbb{E}[H_k] = p \quad \forall k$.
- $S_N = H_1 + \cdots + H_N$ is the total number of hits after N throws.
- *Strong Law of Large Numbers:*

$$\frac{S_N}{N} \to p \quad \text{almost surely (a.s.)}$$

- Compute realizations of H_k by sampling $X_k \sim \mathrm{U}[0, d/2]$ (distance of needle center to closest line) and $\Theta_k \sim \mathrm{U}[0, \pi/2]$ (acute angle of needle with lines) using a random number generator.

Monte Carlo Samples for the Buffon Needle Problem

Results of the Monte Carlo Simulation

- Setting $d = 2$, $\ell = 1$ gives $p = \frac{1}{\pi}$. We should get $N/S_N \overset{N\to\infty}{\longrightarrow} \pi$.
- A Matlab experiment yields

N	S_N	N/S_N	rel. Error
10	3	3.333	6.10e-2
100	32	3.125	5.28e-3
1000	330	3.030	3.54e-2
10000	3188	3.137	1.54e-3

- Mario Lazzarini (1901) built machine that carries out repetitions of this random experiment. His needle was 2.5cm long and the lines 3.0cm apart. He claims to have observed 1808 intersections for 3408 throws, i.e

$$\pi \approx 2 \cdot \frac{2.5}{3} \cdot \frac{3408}{1808} = 3.141592920353983\ldots$$

A relative error of $8.5 \cdot 10^{-8}$! Is this too good to be true?

2. Convergence Results for Basic Monte Carlo Simulation

Basic Monte Carlo Simulation – Convergence Results

- Given a sequence $\{X_k\}$ of i.i.d. copies of a given random variable X, basic MC simulation uses the estimator

$$\mathbb{E}[X] \approx \frac{S_N}{N}, \qquad S_N = X_1 + \cdots + X_N.$$

- By the *Strong Law of Large Numbers*, $\dfrac{S_N}{N} \to \mathbb{E}[X]$ a.s.

- Also, for any measurable function f, $\dfrac{1}{N}\sum_{k=1}^{N} f(X_k) \to \mathbb{E}[f(X)]$ a.s.

- If $\mathbb{E}[X] = \mu$ and $\mathbf{Var}[X] = \sigma^2$, then (via the *Central Limit Theorem*)

$$\mathbb{E}[S_N] = N\mu, \quad \mathbf{Var}[S_N] = N\sigma^2 \quad \text{and} \quad S_N^* = \frac{S_N - N\mu}{\sqrt{N}\sigma} \to \mathrm{N}(0,1),$$

  i.e. the estimate is unbiased, the standard error is $\sigma N^{-1/2}$ and the distribution of the normalised RV S_N^* becomes Gaussian as $N \to \infty$.

  (if $\mathbf{Var}[X] < \infty$ then the **normalised RV** $X^* := \frac{X - \mathbb{E}[X]}{\sqrt{\mathbf{Var}[X]}}$ has $\mathbb{E}[X^*] = 0$, $\mathbf{Var}[X^*] = 1$)

Various Convergence Statements

1. Since

$$\mathbb{E}\left[\left(\frac{S_N}{N} - \mu\right)^2\right] = \mathbf{Var}\,\frac{S_N}{N} = \frac{\sigma^2}{N} \to 0,$$

   we have mean square convergence of S_N/N to μ.

2. *Chebyshev's Inequality* implies, for any $\epsilon > 0$,

$$\mathbb{P}\left\{\left|\frac{S_N}{N} - \mu\right| > N^{-1/2+\epsilon}\right\} \le \frac{\sigma^2}{N^{2\epsilon}},$$

   i.e. the probability of the error being $> N^{-1/2+\epsilon}$ converges to zero as $N \to \infty$.

3. If $\rho := \mathbb{E}\left[|X-\mu|^3\right] < \infty$, then the *Berry-Esseen Inequality* gives

$$|\mathbb{P}\{S_N^* \le x\} - \Phi(x)| \le \frac{\rho}{2\sigma^3\sqrt{N}},$$

   where Φ denotes *cumulative density function (CDF)* of $\mathrm{N}(0,1)$.

Confidence Intervals

Exercise 2.1

(a) Using the Berry-Esseen bound derive a confidence interval for the estimate $\frac{S_N}{N}$ and (upper and lower) bounds on the probability that μ falls into the interval.

(b) In the Buffon needle problem, we have

$$\mathbb{E}[H_k] = p,\ \mathbf{Var}[H_k] = p(1-p),\ \mathbb{E}\left[|H_k - p|^3\right] = p(1-p)(1-2p+2p^2).$$

Calculate the confidence interval for this problem in the case $N = 3408$, $\ell = 2.5$, $d = 3$, and thus check how likely it is that Lazzarini's machine would produce 1808 intersections and a relative accuracy of π of $8.5 \cdot 10^{-8}$.

Please pause the video and attempt this exercise yourself before resuming!

Proposition 2.2 (Asymptotic 95% confidence interval for MC estimate)

$$0.95 - \frac{\rho}{\sigma^3\sqrt{N}} \leq \mathbb{P}\left\{\mu \in \left[\frac{S_N}{N} - \frac{1.96\sigma}{\sqrt{N}}, \frac{S_N}{N} + \frac{1.96\sigma}{\sqrt{N}}\right]\right\} \leq 0.95 + \frac{\rho}{\sigma^3\sqrt{N}}$$

3. Improving the Monte Carlo Method

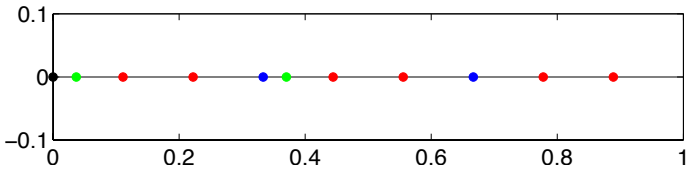
Quasi-Monte Carlo Methods

In quasi-Monte Carlo methods, the samples are not chosen randomly, but special (deterministic) number sequences, known as low-discrepancy sequences, are used instead. Discrepancy is a measure of equidistribution of a number sequence.

Example.

The van der Corput sequence is such a low-discrepancy sequence for the unit interval. For base 3, it is given by $x_n = \frac{k}{3^j}$, where j increases monotonically and, for each j, k runs through all nonnegative integers such that $k/3^j$ is an irreducible fraction < 1. The ordering in k is obtained by representing k in base 3 and reversing the digits. The first 11 numbers are

$$\{x_n\}_{n=1}^{11} = \{0, \frac{1}{3}, \frac{2}{3}, \frac{1}{9}, \frac{4}{9}, \frac{7}{9}, \frac{2}{9}, \frac{5}{9}, \frac{8}{9}, \frac{1}{27}, \frac{10}{27}\}.$$



Quasi-Monte Carlo Methods

- Replacing i.i.d. random numbers sampled from $\mathrm{U}[0,1]$ in a standard Monte Carlo approximation of $\mathbb{E}[f(X)]$ for some $f \in C^\infty(0,1)$ and $X \sim \mathrm{U}[0,1]$, by the van der Corput sequence of length N, yields a quasi-Monte Carlo method.
- The convergence rate is improved from $\mathcal{O}(N^{-1/2})$ to $\mathcal{O}(N^{-2})$.
- Although this improvement in one dimension is impressive, the method does not generalise easily and the rate of convergence depends on the problem.
- In particular, the rate of convergence for a quasi-Monte Carlo method generally does depend on the dimension.

Section 3: More details on QMC methods and their analysis

Variance Reduction

The constant in the MC convergence rates is the variance σ^2 of the RV from which MC samples are being drawn. By designing an equivalent MC approximation with lower variance, we can expect faster convergence.

- To approximate $\mathbb{E}[X]$ by standard MC, we draw independent samples $\{X_k\}_{k=1}^N$ of X and compute the sample average S_N/N.
- Now assume a second set of samples $\{\tilde{X}_k\}_{k=1}^N$ of X is given with sample average $\tilde{S}_N/N$.
- Since both sample averages converge to $\mathbb{E}[X]$, so does $\frac{1}{2}(S_N/N + \tilde{S}_N/N)$.
- When X_k and $\tilde{X}_k$ are negatively correlated they are called antithetic samples, and the approximation $\frac{1}{2N}(S_N + \tilde{S}_N)$ is a more reliable approximation of $\mathbb{E}[X]$ than $\frac{1}{2N}S_{2N}$.

Variance Reduction

Theorem 3.1

Let the two sequences of RVs $\{X_k\}$ and $\{\tilde{X}_k\}$ be identically distributed with

$$\textbf{Cov}(X_j, X_k) = \textbf{Cov}(\tilde{X}_j, \tilde{X}_k) = 0 \quad \text{for } j \neq k.$$

Then the sample averages S_N/N and $\tilde{S}_N/N$ satisfy

$$\textbf{Var}\left[\frac{S_N + \tilde{S}_N}{2N}\right] = \textbf{Var}\left[\frac{S_{2N}}{2N}\right] + \frac{1}{2}\,\textbf{Cov}\left(\frac{S_N}{N}, \frac{\tilde{S}_N}{N}\right) \leq \textbf{Var}\left[\frac{S_N}{N}\right].$$

- **Worst case:** Variance of average of N samples and N antithetic samples no better than variance of N independent samples.
- **Best case:** negatively correlated S_N/N and $\tilde{S}_N/N$; then the variance of N samples and N antithetic samples is less than the variance of $2N$ indepependent samples.

4. A Simple ODE Example

Predator-prey dynamical system

Now let us apply the Monte Carlo method in a UQ application. Consider the popular **Lotka-Volterra** (or **predator-prey**) model of the dynamics of two interacting populations

$$\dot{\mathbf{u}} = \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \end{bmatrix} = \begin{bmatrix} \theta_1 u_1 \ - \ \theta_{12} u_1 u_2 \\ \theta_{21} u_1 u_2 \ - \ \theta_2 u_2 \end{bmatrix} = \mathbf{f}(\mathbf{u}), \quad \mathbf{u}(0) = \mathbf{u}_0,$$

where u_1 is the number of *prey*, u_2 is the number of *predator* and $\theta_1, \theta_2, \theta_{12}, \theta_{21} \geq 0$ are *parameters* describing the interaction of the two species.

- For simplicity, assume that

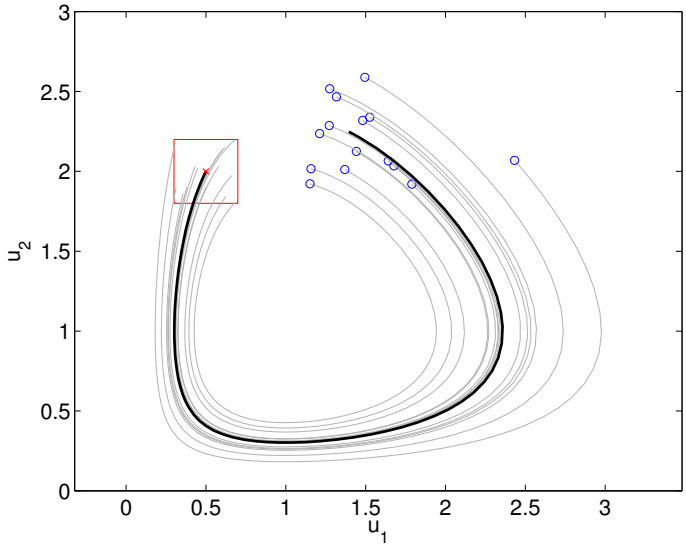
  $$\theta_1 = \theta_2 = \theta_{12} = \theta_{21} = 1$$

  and only the vector of initial conditions $\mathbf{u}_0$ is uncertain.
- It is modeled as a (uniform) random vector $\mathbf{u}_0 \sim \mathrm{U}(\Gamma)$, where Γ denotes the square

  $$\Gamma = \overline{\mathbf{u}}_0 + [-\delta, \delta]^2.$$

- **Goal:** estimate $\mathbb{E}[u_1(T)]$ at time $T > 0$ using the Monte Carlo method.

Predator-Prey Dynamical System – Sample Trajectories



Population dynamics problem (with $\theta_1 = \theta_2 = \theta_{12} = \theta_{21} = 1$) integrated over $[0, T]$ with $\overline{\mathbf{u}}_0 = [0.5, 2]^\mathsf{T}$, $\delta = 0.2$ and $T = 6$. Unperturbed trajectory (black) alongside 15 perturbed trajectories. For the unperturbed trajectory $u_1(T) = 1.3942$.

Modelling Epidemics like COVID-19

- Obviously there are arbitrarily many variations to this simple UQ problem (the distribution of $\mathbf{u}_0$ may be more complicated, the interaction parameters may also be uncertain, there may be more species, or the quantity of interest may be something more complicated) ... **in particular** ...
- A special case of the **Lotka-Volterra** model is the simplest and most widely used model for the spread of diseases (also at the moment with COVID-19), the **SIR model**:

$$\begin{aligned} \dot{S} &= -\tfrac{\beta}{N} SI \\ \dot{I} &= \tfrac{\beta}{N} SI - \gamma I \\ \dot{R} &= \gamma I \end{aligned}$$

  where a population of N individuals is divided into the categories *susceptible* (S), *infecteous* (I) and *recovered* (R).
- The total number $N = S + I + R$ of individuals is assumed to be constant, i.e. birth and death processes are assumed to be negligible.
- For constant N, this problem can be reduced to solving the first two ODEs, which is the Lotka-Volterra system with $\theta_1 = 0$, $\theta_{12} = \theta_{21} = \beta/N$, $\theta_2 = \gamma$.

Modelling Epidemics like COVID-19

To model the current situation, including the lock-down measures, a more accurate model is the **SEIR model**, which includes a category *exposed* (E):

$$\begin{aligned}\dot{S} &= \mu(N-S) - \tfrac{\beta}{N} SI \\ \dot{E} &= \tfrac{\beta}{N} SI - (\mu+\alpha)E \\ \dot{I} &= \alpha E - (\gamma+\mu)I \\ \dot{R} &= \gamma I - \mu R\end{aligned}$$

One of my postdocs, Tobias Siems, is currently collaborating with the *Heidelberg Institute of Global Health*, modelling the **Rhein-Neckar-Kreis** with **SEIR** to predict case numbers and the resulting need for hospital beds.

Explicit Euler Discretisation

- Denote by $\mathbf{u}_M = \mathbf{u}_M(\omega)$ the explicit Euler approximation after $M = M_h$ time steps of length $h = \frac{T}{M}$, starting with initial data $\mathbf{u}_0 = \mathbf{u}_0(\omega)$, i.e.

$$\mathbf{u}_j = \mathbf{u}_{j-1} + h\mathbf{f}(\mathbf{u}_{j-1}), \quad j = 1, \dots, M_h.$$

- Explicit Euler has consistency order 1 and thus there exists a constant $K > 0$ sucht that the discretisation error can be bounded by

$$\|\mathbf{u}(T) - \mathbf{u}_{M_h}\| \le K\, h\,.$$

e.g. [**Numerik 1** *(Numerical Analysis of ODEs)*, Example 2.2.4 & Theorem 2.2.8].

- Define the quantity of interest (QoI) $Q = \phi(\mathbf{u}(T)) = u_1(T)$ for $\mathbf{u} = [u_1, u_2]^\mathsf{T}$ and estimate $\mathbb{E}[Q_h]$ using the MC method just described with $Q_h = \phi(\mathbf{u}_{M_h})$.
- The QoI ϕ is Lipschitz-continuous with constant $L = 1$, such that also

$$|\mathbb{E}[Q] - \mathbb{E}[Q_h]| = |\mathbb{E}[Q - Q_h]| \le K\, h. \tag{4.1}$$

- Denote the Monte Carlo estimator for $\mathbb{E}[Q_h]$ by

$$\widehat{Q}_h := \widehat{Q}_{h,N} = \frac{1}{N}\sum_{k=1}^{N} Q_h^{(k)} \quad N \text{ samples } Q_h^{(1)}, \dots, Q_h^{(N)} \text{ of } Q_h.$$

Expect better approximations for N large and h small.

Bias-Variance Decomposition – Balancing Error Contributions

Lemma 4.1 (Bias-Variance Decomposition)

The mean square error (MSE) can be expanded

$$\mathbb{E}\left[\left(\mathbb{E}\left[Q\right]-\widehat{Q}_h\right)^2\right]=\left(\mathbb{E}\left[Q-Q_h\right]\right)^2+\frac{\mathbf{Var}[Q_h]}{N}$$

Please pause the video again to attempt to prove this lemma yourself!

Proof. Demonstrated on tablet.

Hint: Note that $\mathbb{E}[Q]$ is constant and only $\widehat{Q}_h$ is actually random. □

Thus, using the bias error bound above and the fact that, for h sufficiently small, $\mathbf{Var}[Q_h] \leq \sigma^2_{\text{bnd}} \leq 1.1\,\mathbf{Var}[Q]$ (independently of h), we get the following bound:

$$\text{MSE} := \mathbb{E}\left[\left(\mathbb{E}\left[Q\right]-\widehat{Q}_h\right)^2\right] \;\leq\; K^2h^2+\sigma^2_{\text{bnd}}N^{-1} \tag{4.2}$$

Balancing Discretisation and Sampling Error (in probability)

Using the above convergence results, the error can also be bounded and balanced in probability:

- Error with N samples and $M = T/h$ time steps:

$$e_{h,N} := |\mathbb{E}\left[Q\right]-\widehat{Q}_h| \leq \underbrace{|\mathbb{E}\left[Q\right]-\mathbb{E}\left[Q_h\right]|}_{\text{discretisation error}}+\underbrace{|\mathbb{E}\left[Q_h\right]-\widehat{Q}_h|}_{\text{Monte Carlo error}}$$

- For the MC error, from **Exercise 2.1** with $\mathbf{Var}[Q_h] = \sigma_h^2 \leq \sigma^2_{\text{bnd}}$ we get

$$\mathbb{P}\left\{\left|\mathbb{E}\left[Q_h\right]-\widehat{Q}_h\right| \leq \frac{1.96\sigma_h}{\sqrt{N}}\right\} > 0.95+\mathcal{O}(N^{-1/2})$$

- Combined with discretisation error in (4.1) (using triangle inequality):

$$\mathbb{P}\Big\{e_{h,N} \leq K\,h+1.96\sigma_h N^{-1/2}\Big\} > 0.95+\mathcal{O}(N^{-1/2}). \tag{4.3}$$

Monte Carlo Complexity for Predator-Prey Problem

Finally noting that the cost in each time step is 8 FLOPs, the total cost for the MC estimator is

$$\text{Cost}(\widehat{Q}_h) = 8M_hN = 8T\,h^{-1}N \quad \text{(FLOPs)} \tag{4.4}$$

and we have the following complexity result:

Proposition 4.2 (Monte Carlo Complexity)

The **total cost** *to compute a* **standard Monte Carlo** *estimator for* $\mathbb{E}[u_1(T)]$ *for the* **predator-prey** *model with* **explicit Euler** *time discretisation, such that* $\mathit{MSE} < \varepsilon^2$ *or* $\mathbb{P}\{e_{h,N} < \varepsilon\} > \theta$ *for any* $\theta \in (0,1)$*, satisfies*

$$\mathit{Cost}(\widehat{Q}_h) \;=\; \mathcal{O}(\varepsilon^{-3}).$$

Proof. (only the proof for in probability) A sufficient condition for $e_{h,N} < \varepsilon$ is

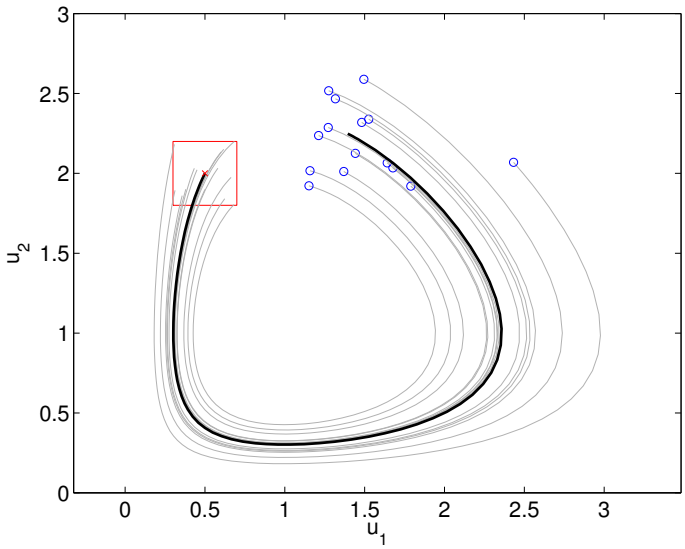
$$K\,h = \varepsilon/2 \quad \text{and} \quad 1.96\sigma_h N^{-1/2} = \varepsilon/2 \quad \text{(balancing the two terms)}.$$

This leads to

$$h = \tfrac{1}{2}K\varepsilon \quad \text{and} \quad N = 3.92^2\sigma_h^2\varepsilon^{-2} \quad \Rightarrow \quad \text{Cost}(\widehat{Q}_h) \le \frac{256T\sigma_{\text{bnd}}^2}{K}\,\varepsilon^{-3}\,.$$

□

Sample Trajectories



Population dynamics problem (with $\theta_1 = \theta_2 = \theta_{12} = \theta_{21} = 1$) integrated over $[0,T]$ with $\overline{\mathbf{u}}_0 = [0.5, 2]^{\mathsf{T}}$, $\delta = 0.2$ and $T = 6$. Unperturbed trajectory (black) alongside 15 perturbed trajectories. For the unperturbed trajectory $u_1(T) = 1.3942$.

Antithetic Sampling for the Predator-Prey System

We may introduce antithetic sampling to this problem by noting that, if $\mathbf{u}_0 \sim \mathrm{U}(\Gamma)$ with $\Gamma = \overline{\mathbf{u}}_0 + [-\delta, \delta]^2$, then the same holds for the random vector

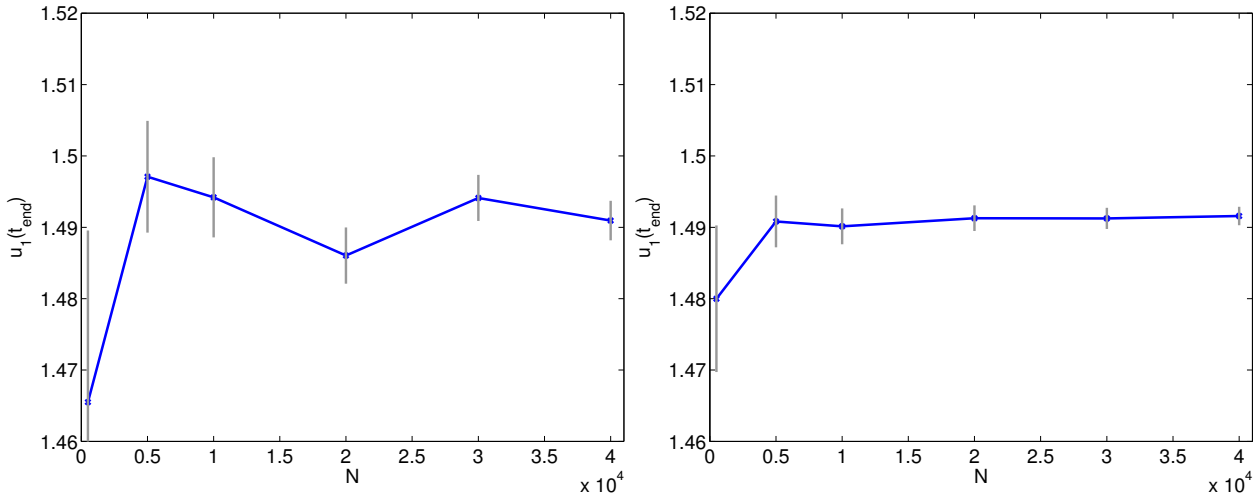
$$\tilde{\mathbf{u}}_0 := 2\overline{\mathbf{u}}_0 - \mathbf{u}_0.$$

Thus, the trajectories generated by the random initial data $\tilde{\mathbf{u}}_0$ have the same distribution as those generated by $\mathbf{u}_0$.

- Let $Q_h = \phi(\mathbf{u}_{M_h})$ be the basic samples and $\tilde{Q}_h = \phi(\tilde{\mathbf{u}}_{M_h})$ the antithetic counterparts. Note that all pairs of samples are independent except each sample and its antithetic counterpart.
- Then use $\frac{1}{2}(\widehat{Q}_{h,N} + \widehat{\tilde{Q}}_{h,N})$ instead of $\widehat{Q}_{h,2N}$ (same cost).
- For the actual implementation, to estimate $\mathbf{Var}[Q_h]$ and $\mathbf{Cov}(Q_h, \tilde{Q}_h)$ we can use sample variance and covariance (resp.), i.e.

$$\frac{1}{N-1}\sum_{k=1}^{N}(Q_h^{(k)} - \widehat{Q}_{h,N})^2 \quad \text{and} \quad \frac{1}{N-1}\sum_{k=1}^{N}(Q_h^{(k)} - \widehat{Q}_{h,N})(\tilde{Q}_h^{(k)} - \widehat{\tilde{Q}}_{h,N})$$

Numerical Experiment – Comparing Standard and Antithetic Sampling



MC estimation of $\mathbb{E}[u_1(T)]$ using standard MC with N samples (left) vs. MC with antithetic sampling using $N/2$ samples of the initial data (right), showing the estimate along with 95% confidence intervals.

Programming Exercise

Exercise 4.3

(a) *Find an estimate for* **Var** $\left[\frac{1}{2}(\widehat{Q}_{h,N} + \widehat{\widetilde{Q}}_{h,N})\right]$ *based on the sample variances and sample covariances of* $\{Q_h^{(k)}\}$ *and* $\{\tilde{Q}_h^{(k)}\}$ *defined above.*

(b) *Implement the Monte Carlo method for the predator-prey system with* $\overline{\mathbf{u}}_0 = [0.5, 2]^\mathsf{T}$, $\delta = 0.2$, $T = 6$, *using explicit Euler discretisation, i.e.*

$$\dot{\mathbf{u}} = \mathbf{f}(\mathbf{u}) \quad \textit{and} \quad \mathbf{u}(0) = \mathbf{u}_0 \quad \longrightarrow \quad \mathbf{u}_j = \mathbf{u}_{j-1} + h\,\mathbf{f}(\mathbf{u}_{j-1}).$$

Study the rates of convergence of the discretisation and MC errors and compute confidence intervals.

(c) *Implement also the antithetic estimator and compare the variance of the two estimators. How much is the variance reduced? Does this reduction depend on the selected tolerance* ε.

5. The Multilevel Monte Carlo Method

History

- The **multilevel Monte Carlo** method is a powerful new variance reduction technique (especially for UQ applications).
- First ideas for high-dimensional quadrature by [Heinrich, 2000].
- Independently discovered and popularised by [Giles, 2007] in the context of **stochastic DEs in mathematical finance**.
- First papers in the context of UQ:
  - [Cliffe, Giles, **RS**, Teckentrup, 2011]
  - [Barth, Schwab, Zollinger, 2011]
- Stochastic simulation of discrete state systems (biology, chemistry) by [Anderson, Higham, 2012]
- ...

Goal: Estimate $\mathbb{E}[Q]$ for an inacccessible random variable Q (e.g. derived from solution of a DE model). However, we have access to a sequence of approximations $Q_h \approx Q$, parametrised by h (#time steps, #grid points, ...) s.t. $\lim_{h\to 0} Q_h = Q$.

Idea: Reduce variance by a clever use of a hierarchy of approximations.

Abstract Complexity result for Standard MC

Recall from Lemma 4.1 that the mean square error (MSE) for the standard MC estimator $\widehat{Q}_{h,N}$ (using samples from the approximation Q_h instead of Q) expands as

$$\mathbb{E}\left[\left(\widehat{Q}_{h,N} - \mathbb{E}[Q]\right)^2\right] = \left(\mathbb{E}[Q_h - Q]\right)^2 + \frac{\mathbf{Var}[Q_h]}{N}.$$

Thus, we can derive an abstract version of Proposition 4.2 (with identical proof):

Theorem 5.1 (Abstract Complexity Theorem for standard MC)

Assume that there exist constants $\alpha, \gamma > 0$, such that

$$|\mathbb{E}[Q_h - Q]| = \mathcal{O}(h^\alpha), \quad \text{as } h \to 0, \tag{5.1}$$

$$\mathit{Cost}(Q_h^{(k)}) = \mathcal{O}(h^{-\gamma}), \quad \text{as } h \to 0, \tag{5.2}$$

where $\mathit{Cost}(Q_h^{(k)})$ denotes the cost per sample from approximation Q_h.

Then, for any $\varepsilon > 0$ and $\theta \in (0,1)$, the **total cost** *to compute a standard Monte Carlo estimator for $\mathbb{E}[Q]$, such that $\mathit{MSE} < \varepsilon^2$ or $\mathbb{P}\{e_{h,N} < \varepsilon\} > \theta$, satisfies*

$$\mathit{Cost}(\widehat{Q}_{h,N}) = \mathcal{O}(\varepsilon^{-2-\gamma/\alpha}).$$

Multilevel Estimator

- **Key idea:** use samples of Q_h on a hierarchy of different **levels**, i.e., for different values $h_0, \dots, h_L$ of the discretization parameter, and decompose

$$\mathbb{E}\left[Q_{h_L}\right] = \mathbb{E}\left[Q_{h_0}\right] + \sum_{\ell=1}^{L} \mathbb{E}\left[Q_{h_\ell} - Q_{h_{\ell-1}}\right] =: \sum_{\ell=0}^{L} \mathbb{E}\left[Y_\ell\right],$$

- For simplicity, we will often choose $h_{\ell-1} = m h_\ell$, $\ell = 1, \dots, L$, for some $m \in \mathbb{N} \setminus \{1\}$ and $h_0 > 0$) (uniform grid refinement).
- Given estimators $\{\widehat{Y}_\ell\}_{\ell=0}^{L}$ for $\mathbb{E}\left[Y_\ell\right]$, we refer to

$$\widehat{Q}_L^{\mathsf{ML}} := \sum_{\ell=0}^{L} \widehat{Y}_\ell$$

as a **multilevel estimator** for Q.

- Different variants of this multilevel estimator now arise from different choices of the level estimators, e.g. standard Monte Carlo, quasi-Monte Carlo, etc ...

Multilevel Monte Carlo (MLMC) Estimator

- If each $\widehat{Y}_\ell$ is itself a standard Monte Carlo estimator, i.e.,

$$\widehat{Y}_0 = \widehat{Y}_{0,N_0} := \frac{1}{N_0} \sum_{k=1}^{N_0} Q_{h_0}^{(k)}$$

and

$$\widehat{Y}_\ell = \widehat{Y}_{\ell,N_\ell} := \frac{1}{N_\ell} \sum_{k=1}^{N_\ell} \left(Q_{h_\ell}^{(k)} - Q_{h_{\ell-1}}^{(k)} \right), \qquad \ell = 1, \dots, L,$$

one obtains the multilevel Monte Carlo estimator and $\widehat{Q}_L^{\mathsf{ML}}$ is **unbiased**.

- If all expectations $\mathbb{E}\left[Y_\ell\right]$ are sampled independently (not neccessary), then

$$\mathbf{Var}\, \widehat{Q}_L^{\mathsf{ML}} = \sum_{\ell=0}^{L} \mathbf{Var}\, \widehat{Y}_\ell.$$

and the associated MSE has the standard decomposition

$$\mathbb{E}\left[\left(\widehat{Q}_{L,\{N_\ell\}}^{\mathsf{ML}} - \mathbb{E}\left[Q\right]\right)^2\right] = \mathbb{E}\left[Q_{h_L} - Q\right]^2 + \sum_{\ell=0}^{L} \frac{\mathbf{Var}\, Y_\ell}{N_\ell}$$

into bias and sample variance (shown as for standard MC in Lemma 4.1).

MLMC variance reduction

- Choose the discretisation parameter h_L on the highest level and the numbers of samples $(N_\ell)_{\ell=0}^L$ again to balance the terms in the MSE.
- The bias term is the same as for the standard MC estimator if $h_L = h$, so that under Assumption (5.1), this leads again to a choice of $h_L = \mathcal{O}(\varepsilon^{1/\alpha})$.
- **But why** do we get variance reduction or lower cost for the same variance?
- **Two reasons:**

1. As we coarsen the problem, the cost per sample **decays** rapidly from level to level under Assumption (5.2); by a factor m^γ if $h_{\ell-1}/h_\ell = m$.
2. Since $Q_h \to Q$, then $\mathbf{Var}[Y_\ell] = \mathbf{Var}[Q_{h_\ell} - Q_{h_{\ell-1}}] \to 0$ as $\ell \to \infty$, allowing for smaller and smaller sample sizes N_ℓ on higher and higher levels.

Optimal Sample Sizes

- The cost of the MLMC estimator is

$$\text{Cost}(\widehat{Q}^{\text{ML}}_{L,\{N_\ell\}}) = \sum_{\ell=0}^{L} N_\ell \mathcal{C}_\ell, \qquad \mathcal{C}_\ell := \text{Cost}(Y_\ell^{(k)}).$$

- Treating the N_ℓ as continuous variables, the cost of the MLMC estimator can be minimised for a fixed variance

$$\sum_{\ell=0}^{L} \frac{\mathbf{Var}\, Y_\ell}{N_\ell} = \frac{\varepsilon^2}{2}$$

- The solution to this constrained minimisation problem is (see below):

$$N_\ell \simeq \sqrt{\mathbf{Var}[Y_\ell]/\mathcal{C}_\ell} \tag{5.3}$$

with implied constant chosen such that the total variance is $\frac{\varepsilon^2}{2}$, leading to the constant $\frac{2}{\varepsilon^2}\sum_\ell \sqrt{\mathcal{C}_\ell \mathbf{Var}[Y_\ell]}$.

Cost Comparison MLMC vs. Standard MC

- Thus the total cost on level ℓ is proportional to $\sqrt{\mathcal{C}_\ell \, \mathbf{Var}[Y_\ell]}$ and therefore

$$\mathrm{Cost}(\widehat{Q}^{\mathrm{ML}}_{L,\{N_\ell\}}) \leq \frac{2}{\varepsilon^2} \left(\sum_{\ell=0}^{L} \sqrt{\mathcal{C}_\ell \, \mathbf{Var}[Y_\ell]} \right)^2$$

- For comparison, standard MC has $\mathrm{Cost}(\widehat{Q}_{h_L,N}) = \frac{2}{\varepsilon^2} \mathcal{C}_L \, \mathbf{Var}[Q_{M_L}]$.
- If $\mathbf{Var}[Y_\ell]$ decays faster than $\mathcal{C}_\ell$ increases, the cost on level $\ell = 0$ dominates. Since $\mathbf{Var}[Q_{h_0}] \approx \mathbf{Var}[Q_{h_L}]$, the cost ratio of MLMC to MC estimation is then approximately

$$\mathcal{C}_0 / \mathcal{C}_L \;\eqsim\; \left(m^{-\gamma}\right)^L$$

- If $\mathcal{C}_\ell$ increases faster than $\mathbf{Var}[Y_\ell]$ decays, then the cost on level $\ell = L$ dominates, and then the cost ratio is approximately

$$\mathbf{Var}[Y_L] / \mathbf{Var}[Q_{h_L}] \;\eqsim\; \varepsilon^2$$

(provided $\mathbb{E}\left[(Q - Q_L)^2\right] \eqsim \left(\mathbb{E}\left[Q - Q_L\right]\right)^2$, which is problem dependent).

General Multilevel Monte Carlo Complexity Theorem

Theorem 5.2

Let $\varepsilon < \exp(-1)$ and assume that there are constants $\alpha, \beta, \gamma > 0$ such that $\alpha \geq \frac{1}{2} \min\{\beta, \gamma\}$ and, for all $\ell = 0, \ldots, L$,

(M1) $\left|\mathbb{E}\left[Q_{h_\ell}\right] - \mathbb{E}\left[Q\right]\right| = \mathcal{O}(h_\ell^\alpha)$,

(M2) $\mathbf{Var}[Y_\ell] = \mathcal{O}(h_\ell^\beta)$,

(M3) $\mathcal{C}_\ell = \mathcal{O}(h_\ell^{-\gamma})$.

Then there are L and $\{N_\ell\}_{\ell=0}^L$ such that $\mathbb{E}\left[\left(\widehat{Q}^{ML}_{L,\{N_\ell\}} - \mathbb{E}\left[Q\right]\right)^2\right] \leq \varepsilon^2$ and

$$\mathrm{Cost}\left(\widehat{Q}^{ML}_{L,\{N_\ell\}}\right) = \begin{cases} \mathcal{O}\left(\varepsilon^{-2}\right), & \text{if } \beta > \gamma, \\ \mathcal{O}\left(\varepsilon^{-2} \left|\log \varepsilon\right|^2\right), & \text{if } \beta = \gamma, \\ \mathcal{O}\left(\varepsilon^{-2-(\gamma-\beta)/\alpha}\right), & \text{if } \beta < \gamma. \end{cases}$$

Proof. Demonstrated on tablet.

[Giles, 2007] for special case of SDEs with $\alpha = \gamma = 1$.

[Cliffe, Giles, **RS**, Teckentrup, 2011] for the general case. □

Application to the Predator-Prey Problem

In the case of the predator-prey model problem we have already seen in (4.1) and (4.4) that (M1) and (M3) hold with $\alpha = 1$ and $\gamma = 1$, respectively.

Finally, it can be proved similarly to (M1) that (M2) holds with $\beta = 2$, i.e.

$$\begin{aligned}\mathbf{Var}[Y_\ell] = \mathbf{Var}[Q_{h_\ell} - Q_{h_{\ell-1}}] &\leq \mathbb{E}\left[\left(Q_{h_\ell} - Q_{h_{\ell-1}}\right)^2\right] \\ &\leq 2\left(\mathbb{E}\left[\left(Q - Q_{h_{\ell-1}}\right)^2\right] + \mathbb{E}\left[\left(Q - Q_{h_\ell}\right)^2\right]\right) \\ &\leq 2\left(K^2 h_{\ell-1}^2 + K^2 h_\ell^2\right) \\ &\leq \underbrace{2K^2(1+m^2)}_{\text{constant}}\, h_\ell^2 \,.\end{aligned}$$

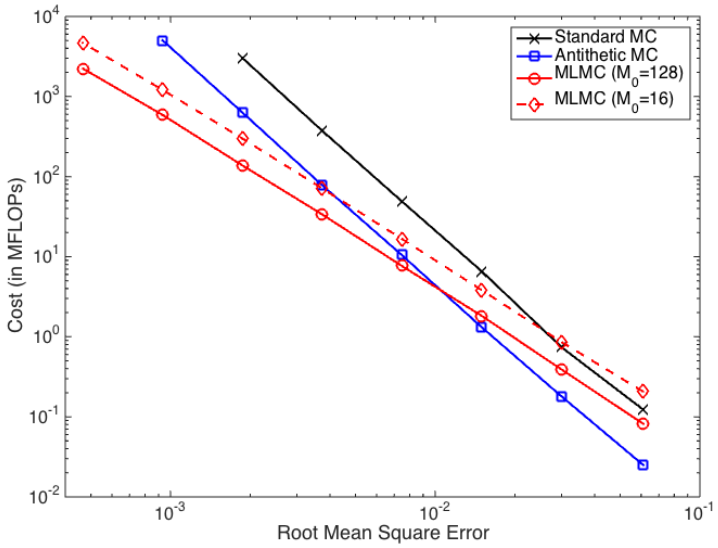
Thus, $\beta > \gamma$ and it follows from **Theorem 5.2** that

$$\mathrm{Cost}\big(\widehat{Q}^{\mathrm{ML}}_{L,\{N_\ell\}}\big) = \mathcal{O}\big(\varepsilon^{-2}\big).$$

Recall that for standard MC we had $\mathrm{Cost}(\widehat{Q}_{h,N}) = \mathcal{O}\big(\varepsilon^{-3}\big)$, so we gained a whole **order of magnitude**.

Numerical Results – CPU time vs. Root Mean Square Error

Comparing the three estimators described – standard MC, anithetic MC & MLMC:



We can observe the variance reduction through antithetic sampling, but the cost for both one level MC methods grows like $\mathcal{O}(\varepsilon^{-3})$, as predicted, while the cost for MLMC grows like $\mathcal{O}(\varepsilon^{-2})$. The actual cost depends on the number of levels.

Adaptive MLMC Algorithm

- The following MLMC algorithm computes the optimal values of L and N_ℓ adaptively using the sample averages $\widehat{Y}_{\ell,N_\ell}$ and sample variances

$$\mathrm{s}_\ell^2 := \frac{1}{N-1}\sum_{k=1}^{N_\ell}\left(Y_\ell^{(k)} - \widehat{Y}_{\ell,N_\ell}\right)^2 \quad \text{of} \quad Y_\ell\,.$$

- Sample variances can be used directly to estimate the MC error on each level.
- To bound the bias error, we assume there exists an $h^\star > 0$ such that the error decay in $|\mathbb{E}[Q_h - Q]|$ is monotonic for $h \le h^\star$ and satisfies

$$ch^\alpha \le |\mathbb{E}[Q_h - Q]| \le Ch^\alpha.$$

- This ensures that in the case $\frac{h_{\ell-1}}{h_\ell} = m$ (via inverse triangle inequality) DIY

$$|\mathbb{E}[Q_{h_\ell} - Q]| \le \tfrac{1}{rm^\alpha - 1}\widehat{Y}_\ell \qquad \text{for} \quad r = c/C.$$

- For the predator-prey problem for example $r = c/C \approx 1$ (you can safely choose $c = 0.9$) and this gives a computable error estimator on level L to determine whether h_L is sufficiently small or whether L needs to be increased.

Adaptive MLMC Algorithm

Adaptive MLMC Algorithm

1. Set h_0, m, ε, $L = 1$ and $N_0 = N_1 = N_{\text{Init}}$.
2. For all levels $\ell = 0, \dots, L$ do
   a. Compute new samples $Y_\ell^{(k)}$ on level ℓ until there are N_ℓ.
   b. Compute $\widehat{Y}_\ell$ and s_ℓ^2, and estimate $\mathcal{C}_\ell$.
3. Update estimates for N_ℓ using the formula in (5.3) and if $\widehat{Y}_L > \frac{rm^\alpha - 1}{\sqrt{2}}\varepsilon$, increase $L \to L+1$ and set $N_L = N_{\text{Init}}$.
4. If $\widehat{Y}_L \le \frac{rm^\alpha - 1}{\sqrt{2}}\varepsilon$ and $\sum_{\ell=0}^{L} s_\ell^2/N_\ell \le \varepsilon^2/2$
   Go to 5.
   Else
   Return to 2.
5. Set $\widehat{Q}^{\text{ML}}_{L,\{N_\ell\}} = \sum_{\ell=0}^{L}\widehat{Y}_\ell$.

Numerical Experiments

Exercise 5.3

(a) *Implement the multilevel MC method for the predator-prey problem. Choose h_0 sufficiently small to avoid stability problems with the explicit Euler method. Compare the cost to achieve a certain tolerance ε for the mean square error (in terms of FLOPs) against your other two implementations (standard and antithetic MC). How big is the computational gain?*

(b) *Recall that $\alpha = \gamma = 1$ and $\beta = 2$ in that case. Verify this with your code. What are the numerically observed rates?*

See [Giles, 2007], [Cliffe, Giles, RS, Teckentrup, 2011] for good ways to visualise your results.

6. Random Fields

Model Elliptic PDE & Random Fields

We return to our model elliptic boundary value problem (radwaste case study)

$$-\nabla\cdot(a\nabla u) = f, \quad \text{on } D \subset \mathbb{R}^d, \qquad u_{|\partial D} = 0, \tag{6.1}$$

where a and f are random fields defined on D.

Definition 6.1

Let $D \subset \mathbb{R}^d$, $d \in \mathbb{N}$, and let $(\Omega, \mathfrak{A}, \mathbb{P})$ be a probability space (see Appendix A). A (real-valued) random field is a mapping

$$a : D \times \Omega \to \mathbb{R}$$

such that each function $a(\mathbf{x}, \cdot) : \Omega \to \mathbb{R}$, $\mathbf{x} \in D$, is a random variable.

Definition 6.2

For each fixed $\omega \in \Omega$ the associated function $a(\cdot, \omega) : D \to \mathbb{R}$ is called a realization of the random field.

Let $\mathbb{R}^D$ denote the set of all real-valued functions $f : D \to \mathbb{R}$. The mapping $\omega \mapsto a(\cdot, \omega)$ from $(\Omega, \mathfrak{A})$ to $(\mathbb{R}^D, \mathfrak{A}(\mathbb{R}^D))$ is measurable and hence a random variable with values in $\mathbb{R}^D$.

Model Elliptic PDE & Random Fields

- Similar to a random vector or a stochastic process, a random field is a family of random variables indexed by a parameter. The former concepts are often tied to an ordered parameter set (e.g. $\mathbb{N}$ or $\mathbb{R}_0^+$), whereas for random fields the parameter is a spatial coordinate, typically from subsets of $\mathbb{R}^2$ or $\mathbb{R}^3$.
- Random fields first arose in the field of geostatistics to model phenomena in Earth Sciences such as hydrology, agriculture or geology.
- Since the data for PDE models often consists of one or more functions of space, it is natural to specify the uncertain or random data for PDEs as random fields.
- Naturally, there are extensions to spatio-temporal random fields featuring an additional (ordered) parameter used to model, e.g., turbulence or meteorological phenomena.
- Before we now apply the Monte Carlo method to (7.1), let us study some properties of random fields.

Second-order and Gaussian Random Fields

Definition 6.3

A random field a on $D \subset \mathbb{R}^d$ is said to be of second order if for all $\mathbf{x} \in D$ there holds $a(\mathbf{x}, \cdot) \in L^2(\Omega; \mathbb{R})$ (see Appendix A). We say a second-order random field a has mean function $\overline{a}(\mathbf{x}) := \mathbb{E}[a(\mathbf{x}, \cdot)]$ and covariance function

$$c(\mathbf{x}, \mathbf{y}) := \mathbf{Cov}(a(\mathbf{x}, \cdot), a(\mathbf{y}, \cdot)), \qquad \mathbf{x}, \mathbf{y} \in D.$$

A sufficient and necessary condition for a being second-order is that $c(\mathbf{x}, \mathbf{y})$ is **symmetric and positive semidefinite**.

Definition 6.4

A random field on $D \subset \mathbb{R}^d$ is called Gaussian if, for any $n \in \mathbb{N}$ and for any $\mathbf{x}_1, \ldots, \mathbf{x}_n \in D$, the random vector $[a(\mathbf{x}_1, \cdot), \ldots, a(\mathbf{x}_n, \cdot)]$ follows an n-variate normal distribution. The field is then uniquely determined by its mean and covariance function.

Random Fields in $L^2(D)$ – Karhunen-Loève Expansion

Let a be a 2nd-order random field on $D \subset \mathbb{R}^d$ with mean $\overline{a}$. Then the centred field $a - \overline{a}$ can be expanded in any complete orthonormal system $\{\psi_m\}_{m \in \mathbb{N}}$ of $L^2(D)$.

The Karhunen-Loève expansion of a results from choosing as a particular CONS the eigenfunctions of the covariance operator $C : L^2(D) \to L^2(D)$ of a, given by

$$(Cu)(\mathbf{x}) = \int_D u(\mathbf{y}) c(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{y}, \quad \mathbf{x} \in D. \tag{6.2}$$

Theorem 6.5 (Karhunen-Loève (KL) Expansion)

Let $a \in L^2(\Omega; L^2(D))$ (see Appendix A) with mean function $\overline{a}(\mathbf{x})$ and denote by $(\lambda_m, a_m)_{m \in \mathbb{N}}$, $\|a_m\|_{L^2(D)} = 1$, the sequence of eigenpairs of the covariance operator C in descending order. Then

$$a(\mathbf{x}, \omega) = \overline{a}(\mathbf{x}) + \sum_{m=1}^{\infty} \sqrt{\lambda_m} \, a_m(\mathbf{x}) \, \xi_m(\omega), \tag{6.3}$$

where the random variables $\xi_m(\omega) = \frac{1}{\sqrt{\lambda_m}} (a(\cdot, \omega) - \overline{a}, a_m)_{L^2(D)}$ have mean zero, unit variance and are pairwise uncorrelated. The series converges in $L^2(\Omega; L^2(D))$. If the random field is, in addition, Gaussian, then $\xi_m \sim \mathrm{N}(0, 1)$ are i.i.d.

One-Dimensional Example [Ghanem & Spanos, 1991]

Example 6.6

For $d = 1$ and $D = [-1, 1]$, consider the exponential covariance function

$$c(x, y) = e^{\frac{-|x-y|}{\ell}}, \qquad \ell > 0.$$

The eigenvalues of the associated covariance operator are given by

$$\lambda_m = \frac{2\ell}{\ell^2\omega_m^2 + 1}, \ (m \text{ even}), \qquad \lambda_m = \frac{2\ell}{\ell^2\tilde{\omega}_m^2 + 1}, \ (m \text{ odd})$$

where ω_m and $\tilde{\omega}_m$ denote the solutions of the transcendental equations

$$1 - \omega\ell\tan(\omega) = 0 \quad \text{and} \quad \tilde{\omega}\ell + \tan(\tilde{\omega}) = 0, \quad \text{respectively.}$$

The associated eigenfunctions are given by

$$f_m(x) = \sqrt{\tfrac{2\omega_m}{1+\sin(2\omega_m)}}\cos(\omega_m x), \qquad \tilde{f}_m(x) = \sqrt{\tfrac{2\tilde{\omega}_m}{1+\sin(2\tilde{\omega}_m)}}\sin(\tilde{\omega}_m x).$$

However, in general it is not possible to compute the KL-expansion analytically.

Practical Application – Truncated KL Expansion

- The KL expansion suggests a convenient approach for approximating a random field to a specified accuracy by truncation:

$$a(\mathbf{x}, \omega) \ \approx \ a_s(\mathbf{x}, \omega) := \overline{a}(\mathbf{x}) + \sum_{m=1}^{s} \sqrt{\lambda_m}\, a_m(\mathbf{x})\, \xi_m(\omega). \tag{6.4}$$

- The truncated RF a_s has the same mean as a and the covariance function

$$c_s(\mathbf{x}, \mathbf{y}) = \sum_{m=1}^{s} \lambda_m a_m(\mathbf{x}) a_m(\mathbf{y}), \qquad \mathbf{x}, \mathbf{y} \in D\,, \tag{6.5}$$

  converges uniformly to c as $S \to \infty$.

- For the variance of the truncated KL expansion, we have DIY

$$\mathbf{Var}(a(\mathbf{x}, \cdot)) - \mathbf{Var}(a_s(\mathbf{x}, \cdot)) = \sum_{m=s+1}^{\infty} \lambda_m a_m(\mathbf{x})^2 \geq 0.$$

  Hence, a_s always underestimates the variance of a. Moreover, this implies

$$\|a - a_s\|^2_{L^2(\Omega; L^2(D))} = \sum_{m=s+1}^{\infty} \lambda_m = \int_D \mathbf{Var}\, a(\mathbf{x})\, \mathrm{d}\mathbf{x} - \sum_{m=1}^{s} \lambda_m\,,$$

  i.e. the truncation error in $L^2(\Omega; L^2(D))$ is explicitly computable.

Stationary and Isotropic Random Fields

Definition 6.7

(a) A random field a is stationary or homogeneous if it is invariant under translation, i.e. if the multivariate distributions of $(a(\mathbf{x}_1,\cdot),\ldots,a(\mathbf{x}_n,\cdot))$ and $(a(\mathbf{x}_1+\mathbf{h},\cdot),\ldots,a(\mathbf{x}_n+\mathbf{h},\cdot))$ are the same, for any $\mathbf{x}_1,\ldots,\mathbf{x}_n$ and $\mathbf{h}$.

(b) A stationary random field a is isotropic if its covariance function is invariant under rotations, i.e.,

$$c(\mathbf{x},\mathbf{y}) = c(r), \qquad r = \|\mathbf{x}-\mathbf{y}\|_2.$$

Example 6.8 (Isotropic Gaussian covariance)

A simple and widely used example of an isotropic covariance function is the *Gaussian covariance* $c(r) = \sigma^2 e^{-r^2/\rho^2}$, where σ^2 and ρ are two constants defining the *variance* and the *correlation length* of the field.

The Matérn Class

A family of isotropic covariance functions that is very popular in spatial statistics, climatology, or machine learning, is the Matérn class with covariance function given by

$$c(r) = \frac{\sigma^2}{2^{\nu-1}\,\Gamma(\nu)} \left(\frac{2\sqrt{\nu}\,r}{\rho}\right)^{\nu} K_\nu\left(\frac{2\sqrt{\nu}\,r}{\rho}\right), \qquad r = \|\mathbf{x}-\mathbf{y}\|_2, \tag{6.6}$$

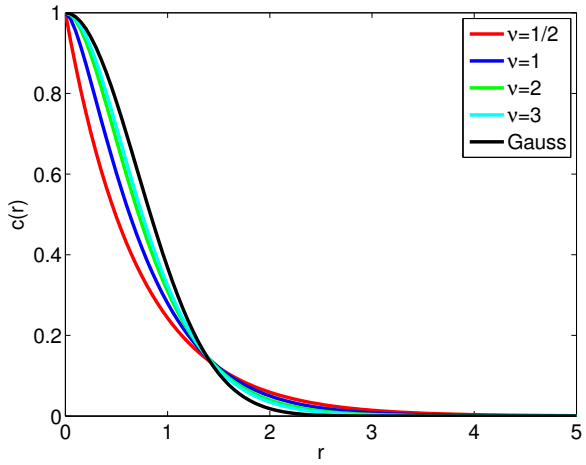
where

K_ν	is the modified (second-kind) Bessel function of order ν,
Γ	denotes the Gamma-function,
ν	is known as the smoothness parameter,
σ^2	is the variance parameter,
ρ	is the correlation length parameter.

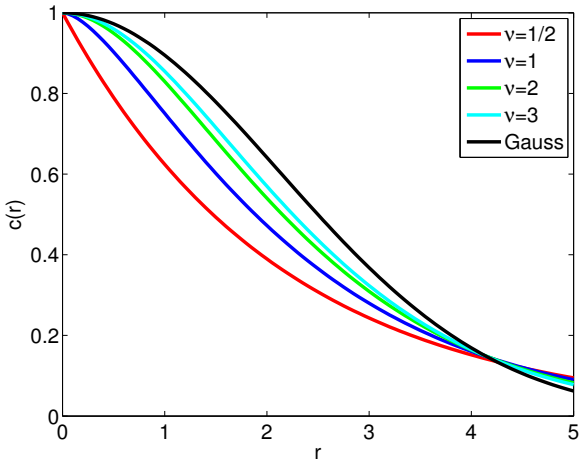
It contains exponential, Gaussian, as well as Bessel covariance functions as special cases:

$\nu = \frac{1}{2}$:	$c(r) = \sigma^2 \exp(-\sqrt{2}r/\rho)$	exponential covariance
$\nu = 1$:	$c(r) = \sigma^2 \left(\frac{2r}{\rho}\right) K_1\left(\frac{2r}{\rho}\right)$	Bessel covariance
$\nu \to \infty$:	$c(r) = \sigma^2 \exp(-r^2/\rho^2)$	Gaussian covariance

The Matérn Class



$\rho = 1$



$\rho = 3$

- By reducing the correlation length ρ the Matérn covariance function can be concentrated more strongly near $r = 0$.
- By increasing the smoothness parameter ν the Matérn covariance function becomes smoother at $r = 0$. (It is analytic everywhere else.)
- The flexibility of the parametrisation allows its application to many statistical situations. (Parameters may be estimated from observed data using statistical techniques.)

Eigenvalue Decay for the Matérn Class

A result by H. Widom from 1963[1] allows us to analyse the decay rate of the eigenvalues of the covariance operator of isotropic random fields:
(His result is more general, but we only consider the the Matérn class.)

Theorem 6.9 (Widom, 1963)

Let $c = c(r)$ be the (isotropic) Matérn covariance function with parameters ν, σ^2 and ρ. Let D be a bounded domain in $\mathbb{R}^d$ and let $\{\lambda_m\}_{m\in\mathbb{N}}$ denote the (nonincreasing) eigenvalues of the covariance operator C given by (6.2).

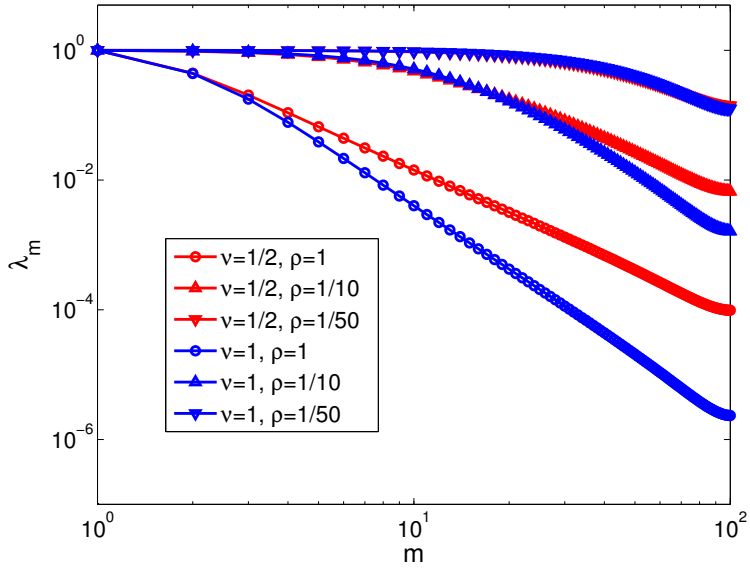
$$\lambda_m \asymp m^{-(1+2\nu/d)}, \qquad \text{for } m \to \infty.$$

- Allows to estimate truncation error and thus dimensionality of the problem.
- Rate of convergence of the eigenvalues is crucial to obtain dimension-independent QMC and sparse grid quadrature and approximation results.
- The (spatial) smoothness of realizations is also linked directly to the parameter ν: in particular, a random field with Matérn covariance function is k-times mean-square differentiable if and only if $\nu > k$.

[1] Widom, H., Asymptotic behavior of the eigenvalues of certain integral equations. *Trans. Amer. Math. Soc.* 109, 278–295 (1963).

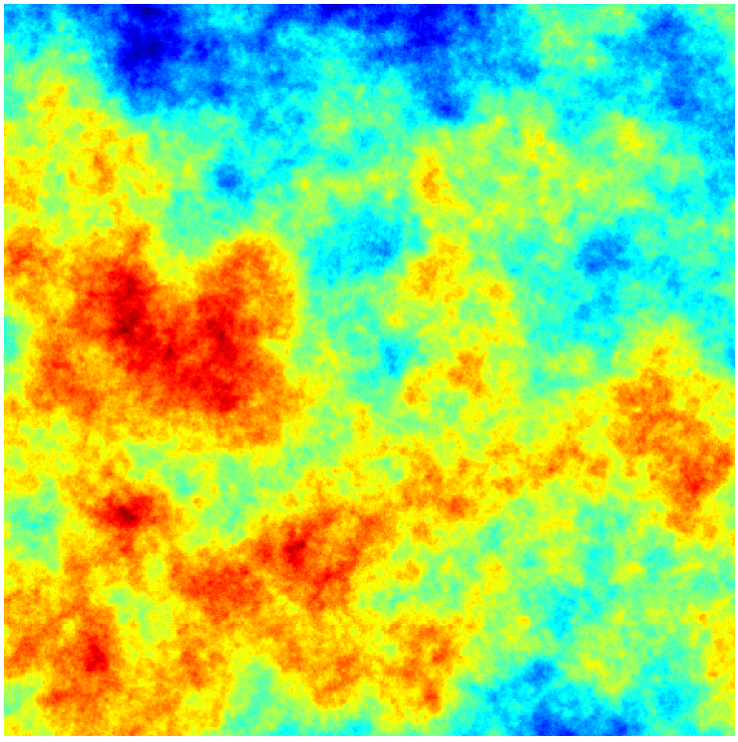
Asymptotic Eigenvalue Decay & Plateau (Matérn)

Before asymptotic decay sets in (determined by the smoothness of the covariance function), there is a preasymptotic plateau whose length is determined by the correlation length parameter ρ.



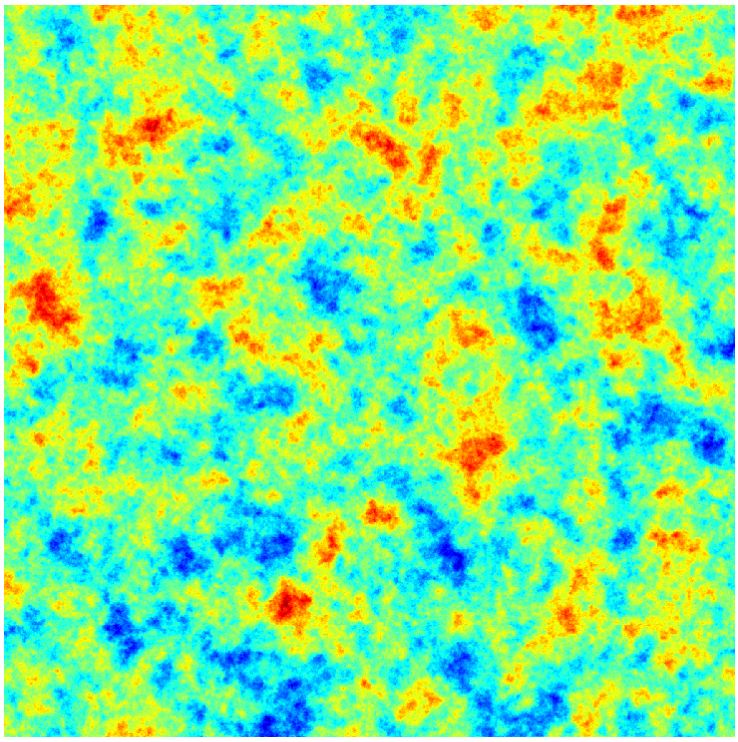
Eigenvalue decay, Matérn covariance kernel, $D = [-1, 1]$.

Realizations of Gaussian Random Fields



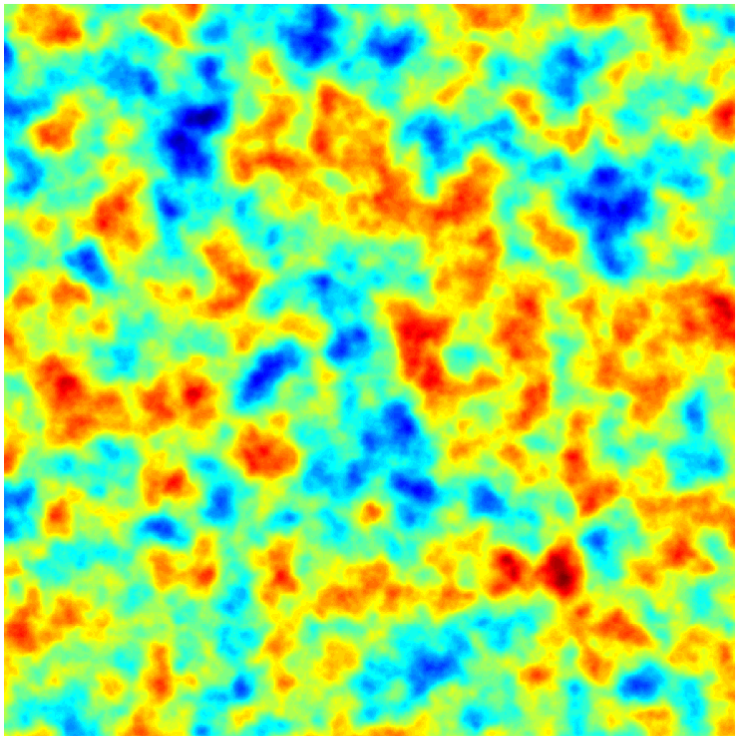
Matérn covariance: $\nu = 1/2$, $\sigma = 1$, $\ell = 0.5$

Realizations of Gaussian Random Fields



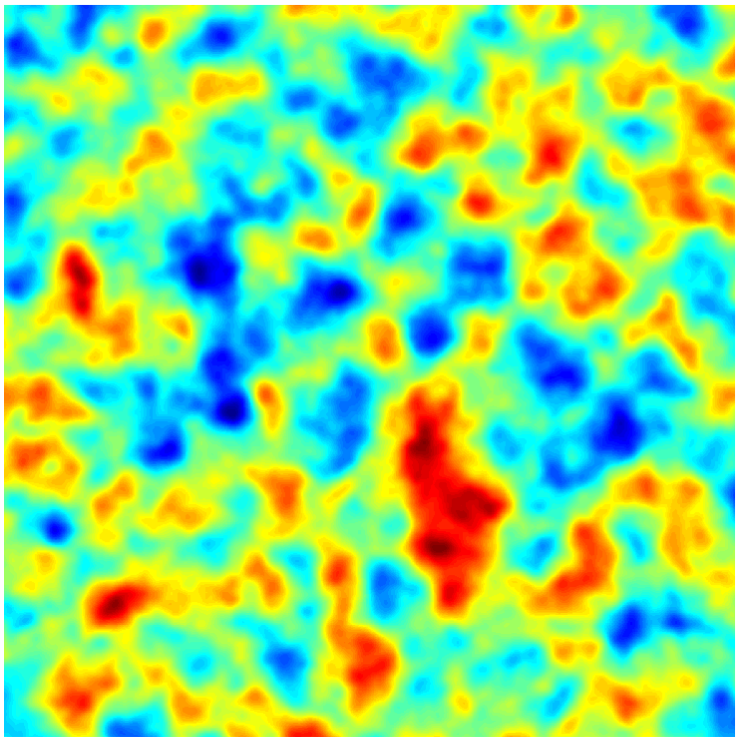
Matérn covariance: $\nu = 1/2$, $\sigma = 1$, $\ell = 0.05$

Realizations of Gaussian Random Fields



Matérn covariance: $\nu = 3/2$, $\sigma = 1$, $\ell = 0.05$

Realizations of Gaussian Random Fields



Matérn covariance: $\nu = 5/2$, $\sigma = 1$, $\ell = 0.05$

Further Reading & Research in Our Group

- KL expansion widely used, especially in theoretical NA literature, because of the very clear convergence theory and parameter dependence.
- **But**, especially for rough fields (e.g. $\nu < 1$ for Matérn), the cost can grow very quickly – both to compute the eigenbasis and to compute realizations.
- More efficient methods for isotropic random fields: circulant embedding and other FFT-based methods:
  - Dietrich & Newsam, Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix, *SIAM J Sci Comput* **18**, 1997
  - Graham, Kuo, Nuyens, **RS** & Sloan, Analysis of circulant embedding methods for sampling stationary random fields, *SIAM J Num Anal* **56**, 2018
  - Bachmayr, Graham, Nguyen & **RS**, Unified analysis of periodization-based sampling methods for Matérn covariances, Preprint arXiv:1905.13522, 2019
- Ole Klein (Bastian's & my AG) wrote (one of?) the fastest parallel circulant embedding code(s): `https://gitlab.dune-project.org/oklein/dune-randomfield`
- Ongoing research on both approaches in our group!

Further Reading & Research in Our Group

- Another very interesting approach to sample random fields is exploiting a link between the inverse C^{-1} of the covariance operator and stochastic PDEs, e.g. Matérn fields can be sampled by solving the sPDE

$$(\kappa^2 - \Delta)^{\beta} a(\mathbf{x}, \omega) =^d \mathcal{W}(\mathbf{x}, \omega) \quad \text{in } \mathbb{R}^d, \tag{6.7}$$

  where Δ is the Laplacian and $\mathcal{W}$ is Gaussian white noise on $\mathbb{R}^d$.

- The resulting RF a is Gaussian with Matérn covariance with parameters $\nu = 2\beta - \frac{d}{2}$, $\rho = 2\frac{\sqrt{\nu}}{\kappa}$ and $\sigma^2 = \sigma^2(\kappa, \beta)$ (e.g. for $d = 2, \nu = 1$, $\sigma^2 = (4\pi\kappa^2)^{-1}$).
  - Lindgren, Rue & Lindström, An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic PDE approach, *J Roy Statist Soc B* **73**, 2011
  - Bolin, Kirchner, Kovács, Numerical solution of fractional elliptic stochastic PDEs with spatial white noise, *IMA J Num Anal* **40**, 2020
- Numerically interesting: can apply fast parallel multigrid solvers to (6.7).
  - Drzisga, Gmeiner, Rüde, **RS** & Wohlmuth, Scheduling massively parallel multigrid for multilevel Monte Carlo methods, *SIAM J Sci Comput* **39**, 2017
- Statistically interesting: can extend easily to non-stationary RFs – ongoing research in our group!

7. Monte Carlo Finite Element Methods

Elliptic Boundary Value Problems with Random Data

We return again to our model elliptic boundary value problem with random data

$$-\nabla\cdot(a\nabla u) = f, \quad \text{on } D \subset \mathbb{R}^d, \qquad u_{|\partial D} = 0, \tag{7.1}$$

where a and f are random fields on D with respect to a probability space $(\Omega, \mathfrak{A}, \mathbb{P})$.

- If f is random, we assume $f(\cdot,\omega) \in L^2(D)$ for (almost) all $\omega \in \Omega$.
- To ensure a unique solution $u(\cdot,\omega) \in H^1_0(D)$ (with norm $\|\cdot\|_{H^1_0(D)} = |\cdot|_{H^1(D)}$) for each realization, **could** require the coefficient a to satisfy Assumption 1 in Appendix B **uniformly**. However, for many applications this is too restrictive and it suffices to require just realization-wise bounds:

Assumption 1

For almost all $\omega \in \Omega$ ($\mathbb{P}$-a.s.), realizations $a(\cdot,\omega)$ of the coefficient function a are strictly positive and lie in $L^\infty(D)$ and satisfy

$$0 < a_{\min}(\omega) \le a(\mathbf{x},\omega) \le a_{\max}(\omega) < \infty \quad \textit{almost everywhere (a.e.)} \quad \textit{in } D, \tag{7.2}$$

where

$$a_{\min}(\omega) := \operatorname*{ess\,inf}_{\mathbf{x}\in D} a(\mathbf{x},\omega), \qquad a_{\max}(\omega) := \operatorname*{ess\,sup}_{\mathbf{x}\in D} a(\mathbf{x},\omega). \tag{7.3}$$

Realization-Wise Solvability

For any realization ω for which Assumption 1 holds and $f(\cdot,\omega) \in L^2(D)$, we may apply the Lax-Milgram Lemma (Lemma B.5) and obtain a unique solution of (7.1).

Theorem 7.1

Let Assumption 1 hold and $f(\cdot,\omega) \in L^2(D)$ $\mathbb{P}$-a.s. Then (7.1) has a unique solution $u(\cdot,\omega) \in H^1_0(D)$ and $|u(\cdot,\omega)|_{H^1(D)} \le C a_{\min}^{-1}(\omega)\|f(\cdot,\omega)\|_{L^2(D)}$ $\mathbb{P}$-a.s.

Recall Definition A.21, of Banach space-valued L^p-spaces over a probability space $(\Omega, \mathfrak{A}, \mathbb{P})$ – so-called *Bochner spaces*. These spaces provide a generalisation of standard Lebesgues spaces. A result that we will use throughout is:

Lemma 7.2 (Hölder's Inequality)

Let $p, q, r \in [1,\infty]$ be such that $\frac{1}{p} = \frac{1}{q} + \frac{1}{r}$. Then

$$\|XY\|_{L^p(\Omega,W)} \le \|X\|_{L^q(\Omega,W)}\|Y\|_{L^r(\Omega,W)}, \quad \textit{for all } X \in L^q(\Omega,W), Y \in L^r(\Omega,W).$$

Note that the case of $q = \infty$ is explicitly included; in that case $p = r$.

For $p = 1$ & $q = r = 2$, Hölder's Inequality reduces to the Cauchy-Schwarz inequality.

The inequality holds over any measure space Ω; in particular, also in standard Lebesgues spaces.

Summability

The following theorem provides sufficient conditions for u to have finite p-th moments, i.e., to lie in $L^p(\Omega; H_0^1(D))$. It follows directly from Theorem 7.1 using Hölder's Inequality (for Part (b)).

Theorem 7.3

Let Assumption 1 hold. Assume further that the mappings $a : \Omega \to L^\infty(D)$ and $f : \Omega \to L^2(D)$ are measurable and that $a_{\min}^{-1} \in L^q(\Omega; \mathbb{R})$ for some $q \in [1, \infty]$.

(a) *If $f \in L^2(D)$ deterministic (i.e. a degenerate constant RF), then*

$$\|u\|_{L^p(\Omega; H_0^1(D))} \leq C \|a_{\min}^{-1}\|_{L^p(\Omega;\mathbb{R})} \|f\|_{L^2(D)}\,, \quad \text{for all } \; p \leq q.$$

(b) *If $f \in L^r(\Omega; L^2(D))$ with $r \in [1, \infty]$ and $\frac{1}{p} = \frac{1}{q} + \frac{1}{r} \leq 1$, then*

$$\|u\|_{L^p(\Omega; H_0^1(D))} \leq C \|a_{\min}^{-1}\|_{L^q(\Omega;\mathbb{R})} \|f\|_{L^r(\Omega; L^2(D))}.$$

(c) *If, in addition, a is independent of f in (b), the bound holds for $p \leq \min(q, r)$.*

Finite Element Discretization

- Let $V_h \subset H_0^1(D)$ denote a closed subspace, e.g., the finite element (FE) space of piecewise polynomial functions with respect to a triangulation $\mathscr{T}_h$ of D with mesh width $h > 0$ (see Appendix B).
- Suppose $u_h : \Omega \to V_h$ satisfies $\mathbb{P}$-a.s.

$$\int_D a(\mathbf{x}, \omega) \nabla u_h(\mathbf{x}, \omega) \cdot \nabla v_h(\mathbf{x})\, \mathrm{d}\mathbf{x} = \int_D f(\mathbf{x}, \omega) v_h(\mathbf{x})\, \mathrm{d}\mathbf{x} \quad \forall v_h \in V_h\,. \quad (7.4)$$

- Since V_h is a closed subspace of $H_0^1(D)$ with norm $|\cdot|_{H^1(D)}$ all the above results hold in an identical form also for u_h:

Theorem 7.4

The results about solvability and summability, as well as the norm bounds in Theorems 7.1 and 7.3 hold under the same assumptions on a and f also for (7.4) and its solution u_h.

H^2 Regularity Assumption & Error Analysis

The regularity assumption, which is necessary to bound the finite element error (cf. Assumption 2 in Appendix B), is again made only realization-wise.

Assumption 2

For almost all $\omega \in \Omega$, there exists a constant $C_2(\omega) > 0$ such that, for every $f(\cdot,\omega) \in L^2(D)$, we have $u(\cdot,\omega) \in H^2(D)$ and

$$|u(\cdot,\omega)|_{H^2(D)} \leq C_2(\omega)\|f(\cdot,\omega)\|_{L^2(D)}.$$

- As discussed in [Bastian, Numerik 2, Sect. 8.3], for Assumption 2 to hold, it suffices that D is convex, $a(\cdot,\omega)$ is Lipschitz cts. and Assumption 1 holds.
- A careful derivation how $C_2(\omega)$ depends on $\|a(\cdot,\omega)\|_{C^{0,1}(D)}, a_{\min}(\omega), a_{\max}(\omega)$ can be found in [Charrier, **RS**, Teckentrup, *SIAM J Num Anal*, 2013].
- In particular, it is shown there that for lognormal a with Matérn covariance, $C_2 \in L^p(\Omega;\mathbb{R})$ for all $p < \infty$.
- The constant C in the interpolation result on Slide 84 of Appendix B is independent of ω.

Finite Element Convergence Results

Let $V^h \subset H_0^1(D)$ be the space of piecewise linear finite elements with respect to a shape-regular triangulation $\mathcal{T}_h$ (see Appendix B).

Theorem 7.5 (Deterministic, L^∞ or Statistically Independent RHS)

Let Assumptions 1 and 2 hold, and let $f : \Omega \to L^2(D)$ be either

(a) constant (i.e. deterministic), (b) in $L^\infty(\Omega;L^2(D))$, or (c) independent of a.

If $a_{\min}^{-1/2} a_{\max}^{1/2} \in L^q(\Omega;\mathbb{R})$ and $C_2 \in L^r(\Omega;\mathbb{R})$ with $q, r \in [1,\infty]$ s.t. $\frac{1}{p} = \frac{1}{q} + \frac{1}{r} \leq 1$, then

$$\|u - u_h\|_{L^p(\Omega;H_0^1(D))} \leq ch \begin{cases} \|f\|_{L^2(D)} & \text{Case (a),} \\ \|f\|_{L^\infty(\Omega;L^2(D))} & \text{Case (b),} \\ \|f\|_{L^p(\Omega;L^2(D))} & \text{Case (c).} \end{cases}$$

- The general case in (b) can be proved in a very similar way.
- Via duality arguments (recall [Bastian, Thm. 8.18]), it is possible to show faster convergence in the (spatial) $L^2(D)$-norm and for functionals $G(u)$ on $H_0^1(D)$, i.e. under Assumption 2 (and further assumptions on G, f and a):

$$\|u - u_h\|_{L^p(\Omega;L^2(D))} = \mathcal{O}(h^2) \quad \text{and} \quad \|G(u) - G(u_h)\|_{L^p(\Omega;\mathbb{R})} = \mathcal{O}(h^2). \quad (7.5)$$

Monte Carlo Finite Element Method

- Our goal now is to use the MC method to estimate a quantity of interest that depends on the (random) solution u. This could be the mean $\mathbb{E}[u(\mathbf{x},\cdot)]$, the variance $\mathbf{Var}[u(\mathbf{x},\cdot)]$ or the expected value of a functional $G(u)$.
- With each of N i.i.d. realizations $a^{(j)} = a(\cdot,\omega_j)$ and $f^{(j)} = f(\cdot,\omega_j)$ we associate the unique solution $u^{(j)} = u(\cdot,\omega_j) \in H_0^1(D)$ as well as the FE approximation $u_h^{(j)} = u_h(\cdot,\omega_j) \in V_h$ and compute the ($H_0^1(D)$-valued) MC estimates

$$\overline{u}_{h,N} := \frac{1}{N}\sum_{j=1}^{N} u_h^{(j)}, \quad s_{h,N}^2 := \frac{1}{N-1}\sum_{j=1}^{N}\left(u_h^{(j)} - \overline{u}_{h,N}\right)^2,$$

  as well as the (scalar-valued) estimate

$$\widehat{Q}_{h,N} := \frac{1}{N}\sum_{j=1}^{N} G(u_h^{(j)}),$$

  for $Q := G(u)$ with $G : H_0^1(D) \to \mathbb{R}$ bounded or Fréchet differentiable.
- To estimate the complexity of these estimators we can use the abstract Theorem 5.1. We simply have to verify Assumptions (5.1) and (5.2).

Let us first consider **Assumption** (5.1):

- For a scalar functional $Q = G(u)$ with $G : H_0^1(D) \to \mathbb{R}$ suff. smooth, using Jensen's inequality (Thm. A.20), it follows from (7.5) that

$$|\mathbb{E}[Q - Q_h]| \le \mathbb{E}[|G(u) - G(u_h)|] = \mathcal{O}(h^2).$$

  Thus, Assumption (5.1) holds with $\alpha = 2$.
- For $Q = u \in H_0^1(D)$, measuring the bias error in $|\cdot|_{H^1(D)}$, we get again using Jensen's inequality (noting that norms are convex functions) and Theorem 7.5 that

$$|\mathbb{E}[u - u_h]|_{H^1(D)} \le \mathbb{E}\big[|u - u_h|_{H^1(D)}\big] = \mathcal{O}(h).$$

  Thus in that case, Assumption (5.1) holds with $\alpha = 1$.

Next consider **Assumption** (5.2):

- If in addition to shape-regularity we also assume that the meshes $\mathscr{T}_h$ are (quasi-)uniform (cf. [Bastian, Numerik 2, Defn. 7.12]) then the number of unknowns M_h in the resulting FE system (B.8) satisfies $M_h = \mathcal{O}(h^{-d})$.
- As proved in [Bastian, Numerik 2, Chap. 10], using a multigrid iterative method it is possible to solve the FE system (B.8) in linear complexity, i.e.

$$\text{Cost}(Q_h^j) \;=\; \mathcal{O}(M_h) \;=\; \mathcal{O}(h^{-d}).$$

  Thus, Assumption (5.2) holds with $\gamma = d$.

Monte Carlo Finite Element Complexity Result

Corollary 7.6

Consider the **Monte Carlo FE method** *with p.w. linear FEs applied to the* **elliptic BVP** (7.1) *in* $\mathbb{R}^d$ *to estimate* $\mathbb{E}[u]$ *or* $\mathbb{E}[G(u)]$, *with* $G : H_0^1(D) \to \mathbb{R}$ *suff. smooth. For any* $\varepsilon > 0$ *and* $\theta \in (0,1)$ *there exist* $h > 0$, $N \in \mathbb{N}$, *such that*

Case $Q = G(u)$: $\|\mathbb{E}[Q] - \widehat{Q}_{h,N}\|_{L^2(\Omega;\mathbb{R})} < \varepsilon$ *or* $\mathbb{P}\{|\mathbb{E}[Q] - \widehat{Q}_{h,N}| < \varepsilon\} > \theta$ *and*

$$\mathit{Cost}(\widehat{Q}_{h,N}) \;=\; \mathcal{O}(\varepsilon^{-2-d/2}).$$

Case $Q = u$: $\|\mathbb{E}[u] - \overline{u}_{h,N}\|_{L^2(\Omega;H_0^1(D))} < \varepsilon$ *or* $\mathbb{P}\{|\mathbb{E}[u] - \overline{u}_{h,N}|_{H^1(D)} < \varepsilon\} > \theta$ *and*

$$\mathit{Cost}(\overline{u}_{h,N}) \;=\; \mathcal{O}(\varepsilon^{-2-d}).$$

Proof. For $Q = G(u)$, we can simply apply Theorem 5.1 with $\alpha = 2$ and $\gamma = d$.

For $Q = u$, the bias-variance decomposition also works in the $|\cdot|_{H^1(D)}$-norm (both in mean squared and in probability). To bound the sampling error, we only require square-summability of $u_h : \Omega \to H_0^1(D)$, which is guaranteed by Theorem 7.4 (under suitable conditions on a and f). □

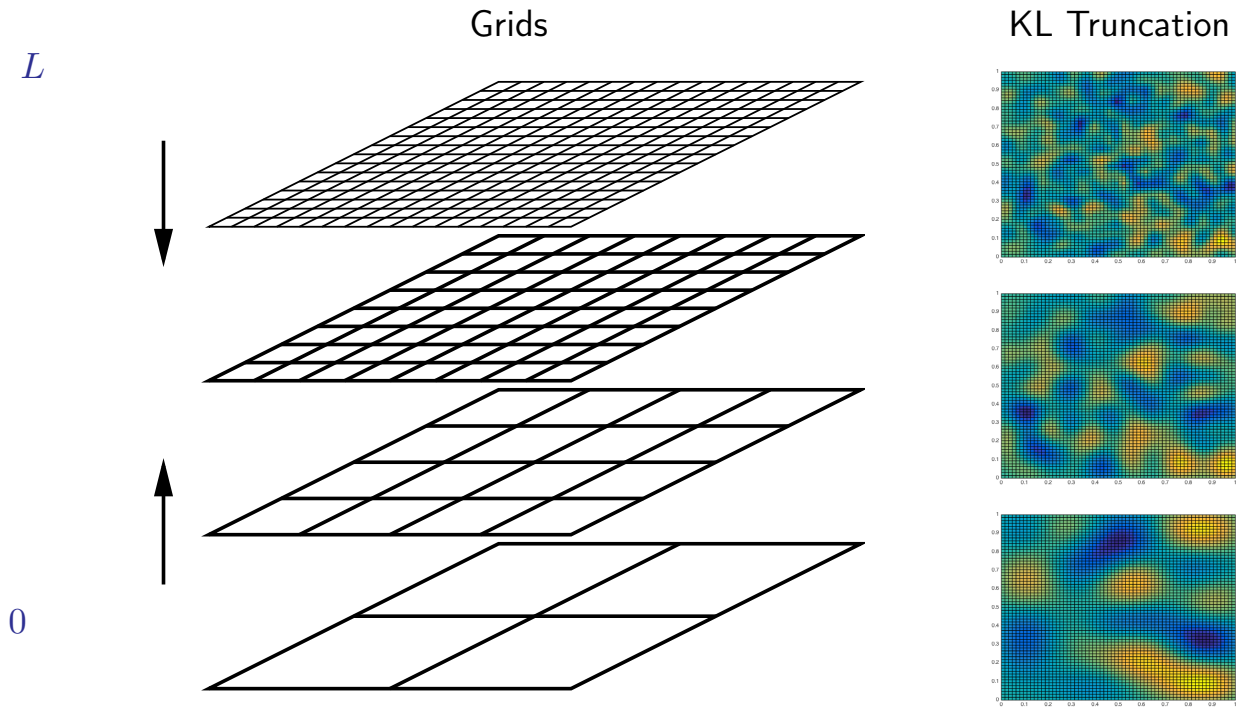
Multilevel Acceleration

- Especially in 2D and 3D this is a very high complexity.
- However, it is straightforward again to accelerate the Monte Carlo Finite Element method via a **multilevel approach.**
- Consider a hierarchy of FE meshes $\mathcal{T}_0, \ldots, \mathcal{T}_L$, for simplicity using uniform grid refinement of an (arbitrary) coarsest grid $\mathcal{T}_0$, i.e. $h_\ell = h_{\ell-1}/2$ $(m = 2)$
- Note that of course these grids are also needed in the multigrid solver we assumed above, so there is no extra overhead.
- The complexity of a multilevel MC-FE estimator for (7.1) and the gains over the standard MC-FE estimator can again easily be estimated using the abstract complexity theorem, Theorem 5.2.
- Assumptions (M1) and (M3) in Theorem 5.2 have already been verified above. So it only remains to consider Assumption (M2).
- For simplicity, we will only consider scalar (smooth) $Q := G(u)$. Using (7.5)

$$\begin{aligned}\mathbf{Var}[Y_\ell] &\le \mathbb{E}\left[(Q_\ell - Q_{\ell-1})^2\right] \\ &\le 2\mathbb{E}\left[(G(u) - G(u_{h_\ell}))^2\right] + 2\mathbb{E}\left[(G(u) - G(u_{h_{\ell-1}}))^2\right] = \mathcal{O}(h_\ell^4)\end{aligned}$$

Thus, Assumption (M2) in Theorem 5.2 holds with $\beta = 4$.

Grid & Model Hierarchy for Elliptic BVP



Have not really discussed how to sample the field or how to also change the truncation dimension across the levels.

Multilevel Complexity Theorem for the Elliptic BVP

Corollary 7.7

Consider the **Multilevel Monte Carlo FE method** *with p.w. linear FEs (uniform refinement) applied to the* **elliptic BVP** (7.1) *in* $\mathbb{R}^d$ *to estimate* $\mathbb{E}[G(u)]$, *with* $G : H_0^1(D) \to \mathbb{R}$ *suff. smooth. For any* $0 < \varepsilon < \exp(-1)$ *and* $\theta \in (0,1)$ *there exist* $L, N_\ell \in \mathbb{N}$, *such that* $\|\mathbb{E}[Q] - \widehat{Q}_L^{ML}\|_{L^2(\Omega;\mathbb{R})} < \varepsilon$ *or* $\mathbb{P}\{|\mathbb{E}[Q] - \widehat{Q}_L^{ML}| < \varepsilon\} > \theta$ *and*

$$\mathrm{Cost}(\widehat{Q}_L^{ML}) \;=\; \mathcal{O}(\varepsilon^{-2}).$$

- For $Q = u$ (see above), for less smooth functionals, or for less smooth data, we often obtain only $\alpha = 1$ and $\beta = 2$, so that for $d = 2, 3$ the other regimes in the MLMC complexity theorem become important.
- Also, for rough coefficients often only $\gamma > d$ is possible (even with a MG solver).
- Thus, we can make the following very important observation (for $d = 2, 3$):

Optimality of MLMC (for $\gamma > \beta = 2\alpha$)

In that case, the MLMC cost is asymptotically the same as **one deterministic solve** to accuracy ε, i.e. $\mathrm{Cost}(\widehat{Q}_L^{ML}) = \mathcal{O}(\varepsilon^{-2-(\gamma-\beta)/\alpha}) = \mathcal{O}(\varepsilon^{-\gamma/\alpha})$ **!!**

Comparison of Complexities

We compare MLMC-FE and MC-FE for (7.1) in the two regimes discussed above:

Case $\alpha = 2$, $\beta = 4$, $\gamma = d$:

d	**MC**	**MLMC**	**Gain**	One Sample Q_L^j
1	$\mathcal{O}(\varepsilon^{-5/2})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-1/2})$	$\mathcal{O}(\varepsilon^{-1/2})$
2	$\mathcal{O}(\varepsilon^{-3})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-1})$	$\mathcal{O}(\varepsilon^{-1})$
3	$\mathcal{O}(\varepsilon^{-7/2})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-3/2})$	$\mathcal{O}(\varepsilon^{-3/2})$

Case $\alpha = 1$, $\beta = 2$, $\gamma = d$:

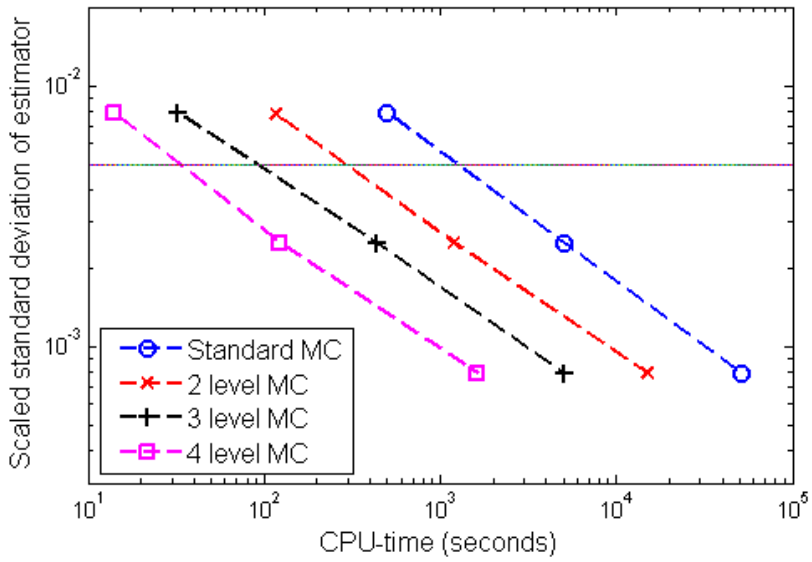
d	**MC**	**MLMC**	**Gain**	One Sample Q_L^j
1	$\mathcal{O}(\varepsilon^{-3})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-1})$	$\mathcal{O}(\varepsilon^{-1})$
2	$\mathcal{O}(\varepsilon^{-4})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2})$
3	$\mathcal{O}(\varepsilon^{-5})$	$\mathcal{O}(\varepsilon^{-3})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-3})$

(ignoring log-factors)

Can we achieve such huge gains in practice?

Multilevel MC-FE Method for Radioactive Waste Disposal Problem

$D = (0,1)^2$; lognormal a w. exponential covariance; $Q = \|u\|_{L_2(D)}$; p.w. linear FE

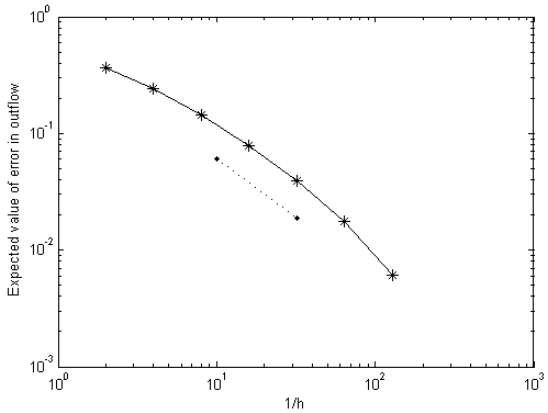


$h_L = 1/256$ (solid line is FE-error)

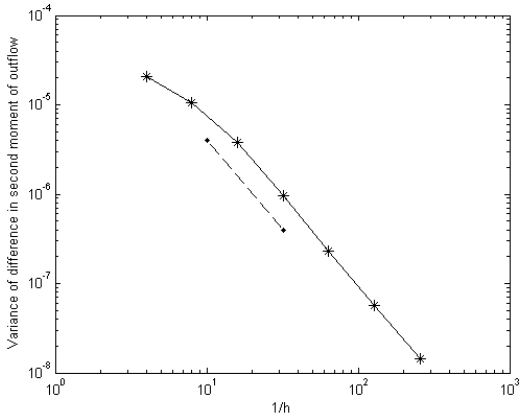
`Matlab` implementation on 3GHz Intel Core 2 Duo E8400 processor, 3.2GByte RAM, with **sparse direct solver**, i.e. $\gamma \approx 2.4$

Verifying Assumptions in Complexity Theorem Numerically

Lognormal a with exponential covariance (i.e. $\nu = 1/2$), $\sigma^2 = 1$ and $\lambda = 0.3$.



$$\left|\mathbb{E}[G_1(u) - G_1(u_h)]\right|$$



$$\mathbb{V}\left[G_2(u_h) - G_2(u_{2h})\right]$$

where, given $\Psi(x) = x$,
$G_1(u) := (f, \Psi)_{L^2(D)} - (a\nabla u, \nabla\Psi)_{L^2(D)}$
(average flow through D).

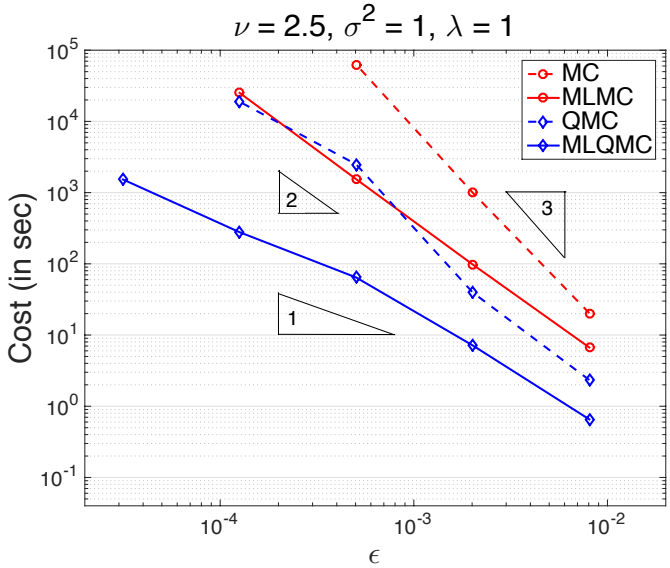
where
$G_2(u) := \left(\frac{1}{|D^*|}\int_{D^*} u(x)\,\mathrm{d}x\right)^2$
(i.e. 2nd moment of u over patch D^*)

$$\Longrightarrow \quad \alpha = 1 \text{ and } \beta = 2$$

Can be proved rigorously! [Teckentrup, **RS** Giles, Ullmann, *Numer Math* **125**, 2013]

Smoother Coefficients & Outlook to Multilevel QMC

$Q = \frac{1}{|D^*|}\int_{D^*} u\,\mathrm{d}x$ & lognormal a with Matérn covariance and



For QMC using a randomised lattice rule with product weights $\gamma_j = 1/j^2$.

[Kuo, **RS**, Schwab, Sloan, Ullmann, *Math Comput* **86**, 2017]

Further Reading & Research in Our Group

- Analysis simplifies considerably for uniformly bounded, affine coefficients, i.e.,

$$0 < a_{\min} = \text{const} < a(\mathbf{x}, \omega) < a_{\max} = \text{const} < \infty \qquad \mathbb{P} - \text{a.s.}$$

  - Barth, Schwab & Zollinger, Multi-level Monte Carlo Finite Element method for elliptic PDEs with stochastic coefficients, *Numer Math* **119**, 2011

- The MLMC-FE method has been applied to many other PDEs. For a comprehensive list see Mike Giles' MLMC Community Webpage
  - `http://people.maths.ox.ac.uk/~gilesm/mlmc_community.html`

- Of particular current interest is the use of adaptive FEs and sample-adaptive model hierarchies in MLMC:
  - Kornhuber & Youett, Adaptive Multilevel Monte Carlo Methods for Stochastic Variational Inequalities, *SIAM J Numer Anal* **56**, 2018
  - Detommaso, Dodwell & **RS**, Continuous Level Monte Carlo and Sample-Adaptive Model Hierarchies, *SIAM/ASA J Uncertain Q* **7**, 2019

- In the latter, we have also extended the concept of MLMC to allow for a continuous level parameter ℓ.

- Ongoing research on all those topics in our group!

Further Reading & Research in Our Group

- Of particular interest is the extension to multilevel Markov chain Monte Carlo
  - Hoang, Schwab & Stuart, Complexity Analysis of Accelerated MCMC Methods for Bayesian Inversion, *Inverse Prob* **29**, 2013
  - Dodwell, Ketelsen, **RS** & Teckentrup, A Hierarchical Multilevel Markov Chain Monte Carlo Algorithm with Applications..., *SIAM/ASA J Uncertain Q* **3**, 2015

- MCMC methods, in particular the Metropolis-Hastings algorithm, allow to sample from unnormalised distributions and from distributions that are known only implicitly (especially in physics the more common situation!)

- In UQ, MCMC can be used for inference on model parameters, where distributions are typically not known a priori – so-called Bayesian Inference.

- **This is the most active research topic in our group!** In particular, not only looking at UQ applications, but also applications in theoretical physics.

- Linus Seelinger (PhD in Bastian's & my AG) is currently implementing an efficient parallel MLMCMC code in MUQ (`http://muq.mit.edu`)

- He is also extending MLMCMC to exploit variance reduction w.r.t. more than one discretization parameter, following the concept (for i.i.d. samples) in
  - Haji-Ali, Nobile & Tempone, Multi-index Monte Carlo: when sparsity meets sampling, *Numer Math* **132**, 2016

High-Dimensional Approximation and Applications in Uncertainty Quantification

III. Quasi-Monte Carlo Methods

Prof. Dr. Robert Scheichl
r.scheichl@uni-heidelberg.de

Dr. Alexander Gilbert
a.gilbert@uni-heidelberg.de

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Institut für Angewandte Mathematik, Universität Heidelberg

Summer Semester 2020

1. Quasi-Monte Carlo Methods

Notation

- $\mathbb{N} := \{1, 2, \ldots\}$
- $\mathbb{N}_0 := \{0\} \cup \mathbb{N}$
- $\mathbb{Z} := \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
- $\mathbb{Z}_N := \{0, 1, 2, \ldots, N-1\}$
- $\mathbb{U}_N := \{z \in \mathbb{Z}_N : \gcd(z, N) = 1\}$
- $\mathfrak{u},\ \mathfrak{v},\ \mathfrak{w} \subseteq \mathbb{N}_0$
- $\boldsymbol{y}_{\mathfrak{u}} := (y_j : j \in \mathfrak{u})$
- for $\mathfrak{u} \subseteq \{1:s\}$ $\boldsymbol{y}_{-\mathfrak{u}} := (y_j : j \in \{1:s\} \setminus \mathfrak{u})$
- $(\boldsymbol{y}_{\mathfrak{u}}, \boldsymbol{a}) := \begin{cases} y_j & \text{if } j \in \mathfrak{u}, \\ a & \text{otherwise.} \end{cases}$
- $\dfrac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} := \displaystyle\prod_{j \in \mathfrak{u}} \frac{\partial}{\partial y_j}$
- $\mathbb{E}_y$ — expectation (w.r.t. y)
- $\mathbb{V}_y$ — variance (w.r.t. y)
- s — dimension of problem
- $\{1:s\} := \{1, 2, \ldots, s\}$
- $[0,1]^s = \underbrace{[0,1] \times [0,1] \times \cdots \times [0,1]}_{s \text{ times}}$
- $y \in \mathbb{R}$, $\boldsymbol{y} = (y_1, y_2, \ldots, y_s) \in \mathbb{R}^s$ $\boldsymbol{t}_k := (t_{k,1}, t_{k,2}, \ldots, t_{k,s}) \in \mathbb{R}^s$
- $\{y\} := y \mod 1$, $\{\boldsymbol{y}\} := (\{y_1\}, \ldots, \{y_s\})$
- pointset: $\mathcal{P}_N := \{\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{N-1}\}$
- sequence: $\mathcal{P} := \{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2, \ldots\}$
- $\sim$ — distributed as
- Uni — uniform distribution
- $\mathrm{N}(\mu, \sigma^2)$ — Normal/Gaussian
- $\lambda_s : \mathbb{R}^s \to \mathbb{R}^+$ — s-dimensional Lebesgue measure
- $\mathbb{1}_A$ — indicator function for a set A

What are *quasi*-Monte Carlo methods?

Let $f : [0,1]^s \to \mathbb{R}$ and suppose we wish to compute

$$\int_{[0,1]^s} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} = \int_0^1 \int_0^1 \cdots \int_0^1 f(y_1, y_2, \ldots, y_s)\,\mathrm{d}y_1 \mathrm{d}y_2 \cdots \mathrm{d}y_s. \tag{1.1}$$

Definition 1.1

A quasi-Monte Carlo (QMC) rule is quadrature approximation of (1.1) with equal weights ($w_k = 1/N$) and deterministic points $\{\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{N-1}\} \subset [0,1]^s$:

$$Q_{s,N} f \coloneqq \frac{1}{N} \sum_{k=0}^{N-1} f(\boldsymbol{t}_k). \tag{1.2}$$

The whole point set is denoted $\mathcal{P}_N \coloneqq \{\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{N-1}\}$ (not necessarily extensible in N). A sequence is denoted by $\mathcal{P} \coloneqq \{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2 \ldots\}$.

Goal: Design point sets/sequences such that:

1. the error of (1.2) converges faster than Monte Carlo (i.e., $<< 1/\sqrt{N}$), and
2. the error is independent of the dimension s.

Comparing different quadrature points

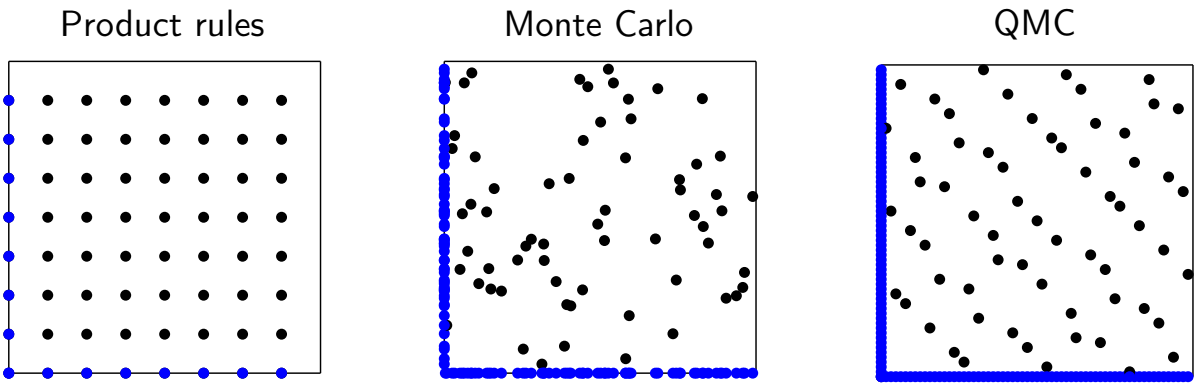


Figure: Examples of different quadrature points in 2D ($N = 64$ points).

- QMC points are more uniformly distributed than Monte Carlo.
- 1D projections of QMC points result in N unique points, compared to $N^{1/s}$ unique points for product rules.

Types of QMC points I: Lattice rules

Definition 1.2

A *rank 1 lattice rule* has points given by

$$\boldsymbol{t}_k = \left\{ \frac{k\boldsymbol{z}}{N} \right\}, \qquad (1.3)$$

where

- N is the number of points,
- $\boldsymbol{z} \in \mathbb{Z}_N^s$ is the *generating vector*, and
- $\{\cdot\}$ denotes that we take the fractional part of each component.

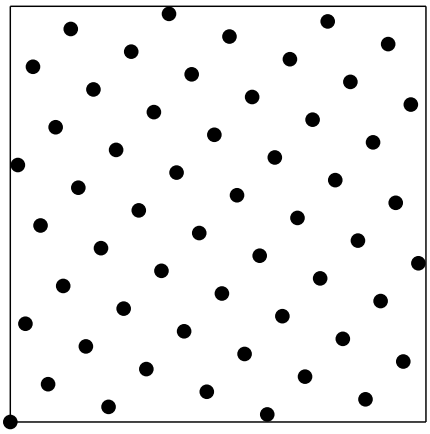


Figure: 2D lattice rule with $N = 55$, $\boldsymbol{z} = (1, 34)$.

Properties:

- quality of lattice rule relies on choosing a "good" $\boldsymbol{z}$,
- simple implementation & low storage cost, and
- simple structure allows for rigorous error analysis (see Section 4).

Types of QMC points II: Digital nets

Definition 1.3

A (t, m, s)-*net in base* b is a set of $N = b^m$ points in $[0, 1)^s$ such that every *elementary interval* of the form

$$\prod_{j=1}^{s} \left[\frac{a_j}{b^{k_j}}, \frac{a_j + 1}{b^{k_j}} \right), \quad k_j, a_j \in \mathbb{Z}_N \text{ s.t. } k_1 + k_2 + \cdots k_s = m - t,\ 0 \le a_j < b^{k_j},$$

with volume $b^{-(m-t)}$ contains exactly b^t points.
A (t, s)-*sequence in base* b is a sequence $\mathcal{P} = \{\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots\}$, such that for all $m > t$ any block of b^m points $\{\boldsymbol{t}_{kb^m}, \ldots, \boldsymbol{t}_{(k+1)b^m}\}$ forms a (t, m, s)-net.

- t is called the *quality parameter* — smaller t implies a better point set.
- Two important examples are polynomial lattice rules and Sobol' sequences. See also the van der Corput, Kronecker, Faure, Niederreiter and Niederreiter–Xing sequences.

Digital net examples

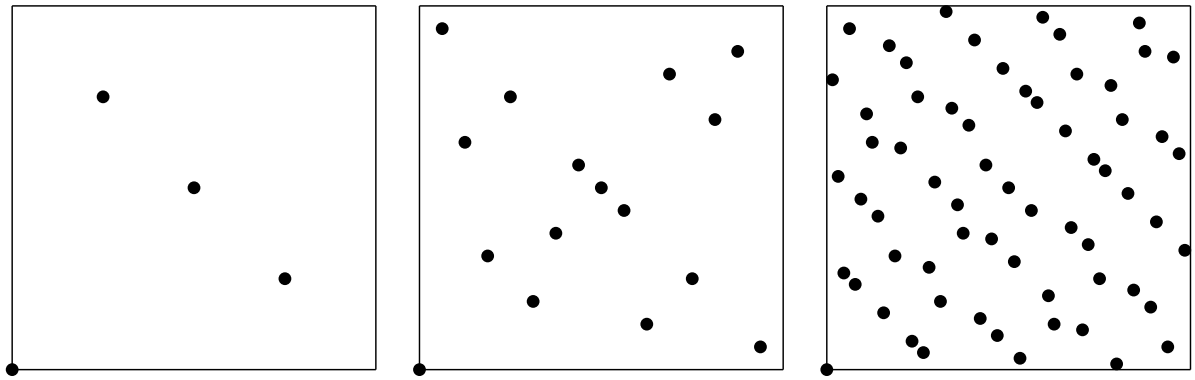


Figure: 2D Sobol' points for $m = 2, 4, 8$ ($N = 4, 16, 64$).

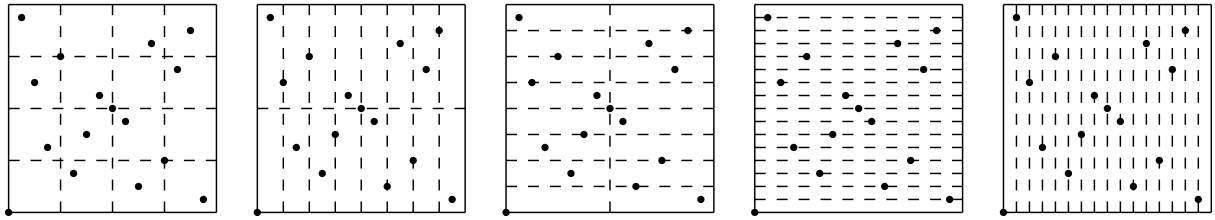


Figure: Sobol' points in elementary intervals for $m = 4$ (16 points).

Digital net construction

Algorithm 1 Digital net construction

Given $m \in \mathbb{N}$, b prime and $G_1, G_2, \ldots, G_s \in \mathbb{Z}_b^{m \times m}$.
For $k = 0, 1, \ldots, N-1$ and $j = 1, 2, \ldots, s$ construct $t_{k,j}$ as follows:

1: Expand k in base b:

$$k \;=\; (k_m \cdots k_2 k_1)_b \;=\; k_1 + k_2 b + \cdots + k_m b^{m-1}.$$

2: Compute $(z_1, z_2, \ldots, z_m)^\top = G_j (k_1, k_2, \ldots, k_m)^\top$.

3: Set

$$t_{k,j} \;=\; (0.z_1 z_2 \cdots z_m)_b \;=\; \frac{z_1}{b} + \frac{z_2}{b^2} + \cdots + \frac{z_m}{b^m}.$$

- Quality of a digital net depends on the *generator matrices* G_j.
- t is not specified in Algorithm 1, and will depend on G_j. Every set of b^m points is an (m, m, s)-net, so clearly $t \le m$.
- Popular choices: the van der Corput sequence uses the $m \times m$ identity, and polynomial lattice rules where entries of G_j are given by the roots of an irreducible polynomial (see [Dick & Pillichshammer 2010]).

Randomised QMC

In practice it is good to use randomised point sets:

$$\mathcal{P}_N \;\mapsto\; \mathcal{P}_N(\omega).$$

The benefits of using *randomised* QMC point sets are:

- Randomised approximations give an unbiased estimate of the integral (1.1).
- Sample variance of multiple realisations of a randomised QMC approximation gives a practical estimate of the mean-square error.
- Randomisation can often aid the theoretical analysis.

Three main methods of randomisation are:

- Shifting — A single random vector from $[0,1)^s$ is added to all points. E.g., randomly shifted lattice rules.
- Scrambling — The points within elementary intervals are randomly and recursively permuted, so that digital net structure is preserved.
- Digital shifting — Similar to shifting except instead the bits in the base b representation of the points are shifted by the bits of a single random shift. It is a type of scrambling.

Randomly-shifted QMC

Let $\mathcal{P}_N$ be a point set. For a *random shift* $\boldsymbol{\Delta} \sim \mathrm{Uni}[0,1)^s$, the *randomly-shifted pointset* $\mathcal{P}_N + \boldsymbol{\Delta}$ consists of points given by

$$\widehat{\boldsymbol{t}}_k \;=\; \{\boldsymbol{t}_k + \boldsymbol{\Delta}\},$$

and the *randomly shifted* QMC approximation is

$$Q_{s,N}(\boldsymbol{\Delta})f \;=\; \frac{1}{N}\sum_{k=0}^{N-1} f(\widehat{\boldsymbol{t}}_k).$$

For $R \in \mathbb{N}$ i.i.d. random shifts $\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2, \ldots, \boldsymbol{\Delta}_R \sim \mathrm{Uni}[0,1)^s$, the *shift-averaged* QMC approximation is

$$\widehat{Q}_{s,N,R}f \;=\; \frac{1}{R}\sum_{r=1}^{R} Q_{s,N}(\boldsymbol{\Delta}_r)f.$$

The mean-square error can be estimated by the sample variance

$$\mathbb{E}_{\boldsymbol{\Delta}}\left[\left|\int_{[0,1]^s} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} - Q_{s,N}(\boldsymbol{\Delta})f\right|^2\right] \;\approx\; \underbrace{\frac{1}{R(R-1)}\sum_{r=1}^{R}\left|Q_{s,N}(\boldsymbol{\Delta}_r)f - \widehat{Q}_{s,N,R}f\right|^2}_{\widehat{\mathbb{V}}[\widehat{Q}_{s,N,R}]}.$$

Randomised QMC examples

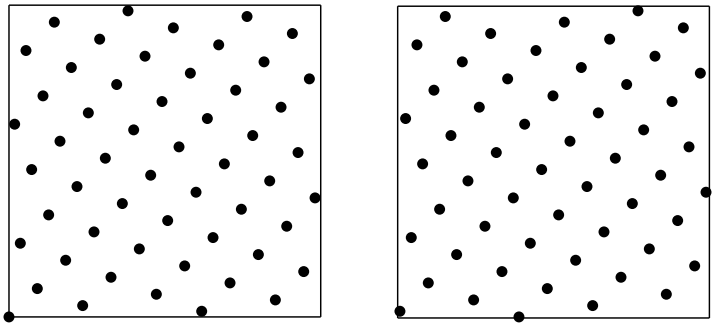


Figure: 2D lattice rules with $N = 55$, $\boldsymbol{z} = (1, 34)$: original (L) and randomly shifted (R).

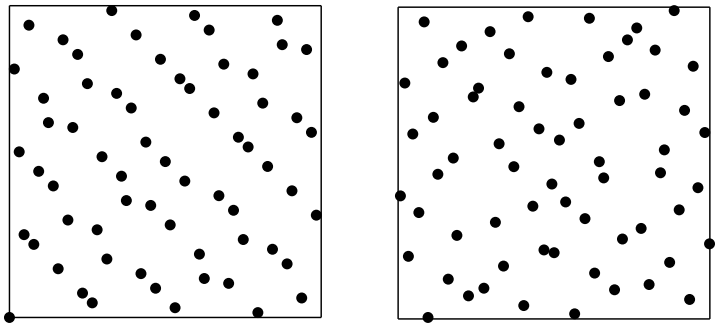


Figure: 64 Sobol' points in 2D: original (L) and randomly scrambled (R).

Summary

- QMC approximation:

$$\int_{[0,1]^s} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \approx Q_{s,N} f = \frac{1}{N} \sum_{k=0}^{N-1} f(\boldsymbol{t}_k).$$

- $\mathcal{P}_N = \{\boldsymbol{t}_0, \ldots, \boldsymbol{t}_{N-1}\}$ chosen deterministically to be well distributed in $[0,1]^s$.
- Two main goals:
  - better than $\mathcal{O}(1/\sqrt{N})$ convergence, and
  - error independent of dimension.
- Two key families: lattice rules and digital nets.
- In practice, it is beneficial to use a randomised QMC approximation. Typically one uses random shifting for lattice rules and random scrambling/digital shifting for digital nets.

2. Classical discrepancy theory

A brief history of quasi-Monte Carlo

- 1950's: [Koksma, Halton, Hlawka, Sobol′, Weyl ...] Number theory was used to construct "low-discrepancy" point sets, and to study their geometric properties.
  Analysis was very theoretical, using geometric and number theoretic tools.
  The application of using such point sets for quadrature not the main focus.
  Dimension was considered fixed, and so the effect of the dimension was largely ignored.
  Result was the error bounds depended poorly on dimension.
- 1990's: QMC applied to problems in finance with great success.
  [Paskov, Traub 1995] used QMC to efficiently approximate a 360-dimensional integral from options pricing. Why did QMC work so well?
- Late 90's – 2000's: Modern QMC theory
  - Weighted Sobolev spaces explain *why* QMC could work well in high-dimensions [Sloan & Woźniakowski 1998].
  - Constructive proof that lattice rules achieve dimension-independent errors that converge as $\mathcal{O}(N^{-1+\delta})$ for $\delta > 0$ [Kuo 2003].
  - Invention of higher-order QMC rules that achieve a dimension-independent errors that converge as $\mathcal{O}(N^{-\alpha})$ for $\alpha \geq 1$ [Dick 2008].
- 2010's: Application of QMC to UQ problems [Graham, Kuo, Nuyens, **RS**, Sloan 2011; Kuo, Schwab, Sloan 2013 & many more]

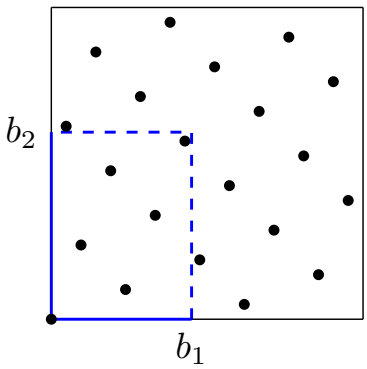
Discrepancy

Definition 2.1

The *discrepancy function* of a point set (sequence) $\mathcal{P}$ at $\boldsymbol{b} \in [0,1)^s$ is defined by

$$\Delta_{s,N}(\mathcal{P}, \boldsymbol{b}) \coloneqq \frac{1}{N} \sum_{k=0}^{N-1} \mathbb{1}_{[\mathbf{0},\boldsymbol{b})}(\boldsymbol{t}_k) - \lambda_s([\mathbf{0}, \boldsymbol{b})).$$

where $[\mathbf{0}, \boldsymbol{b}) = [0, b_1) \times [0, b_2) \times \cdots \times [0, b_s)$ and λ_s is the Lebesgue measure on $[0,1]^s$.



Definition 2.2

The *star discrepancy* of a point set (sequence) $\mathcal{P}$ is defined by

$$D^*_{s,N}(\mathcal{P}) \coloneqq \sup_{\boldsymbol{b} \in [0,1)^s} |\Delta_{s,N}(\mathcal{P}_N, \boldsymbol{b})| = \sup_{\boldsymbol{b} \in [0,1)^s} \left| \frac{1}{N} \sum_{k=0}^{N-1} \mathbb{1}_{[\mathbf{0},\boldsymbol{b})}(\boldsymbol{t}_k) - \prod_{j=1}^{s} b_j \right|.$$

Note: Different notions of discrepancy can be defined by taking a different norm instead of the sup above, e.g., L^p-discrepancy.

Hardy–Krause variation

For $\mathfrak{u} \subseteq \{1:s\} \coloneqq \{1, 2, \ldots, s\}$ and $a \in \mathbb{R}$, define

$$\frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_\mathfrak{u}} \coloneqq \prod_{j \in \mathfrak{u}} \frac{\partial}{\partial y_j} \quad \text{and} \quad (\boldsymbol{y}_\mathfrak{u}, \boldsymbol{a}) \coloneqq \begin{cases} y_j & \text{if } j \in \mathfrak{u}, \\ a & \text{otherwise.} \end{cases}$$

Definition 2.3

Let $f \in C([0,1]^s)$ be such that $\partial^{|\mathfrak{u}|} f / \partial \boldsymbol{y}_\mathfrak{u} \in C([0,1]^s)$ for all $\mathfrak{u} \subseteq \{1:s\}$, then the *variation in the sense of Hardy and Krause* is given by

$$\mathbf{Var}_{\mathrm{HK}}(f) = \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_\mathfrak{u}} f(\boldsymbol{y}_\mathfrak{u}, \mathbf{1}) \right| \mathrm{d}\boldsymbol{y}_\mathfrak{u}.$$

We also define $\|f\|_{\mathrm{HK}} \coloneqq |f(\mathbf{1})| + \mathbf{Var}_{\mathrm{HK}}(f)$, and denote by $\mathrm{HK}([0,1]^s)$ the (Banach) space of all such f as above that also satisfy $\|f\|_{\mathrm{HK}} < \infty$.

Note: The general definition of the HK variation is different from above and still holds when the mixed partial derivatives are not continuous (the two definitions coincide when they are continuous).

Zaremba's identity

Proposition 2.4 (Zaremba's Identity)

Suppose $f : [0,1]^s \to \mathbb{R}$ *has bounded HK variation, then the error of the QMC approximation* (1.2) *using a point set* $\mathcal{P}$ *is*

$$Q_{s,N} f - \int_{[0,1]^s} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} = \sum_{\mathfrak{u} \subseteq \{1:s\}} (-1)^{|\mathfrak{u}|} \int_{[0,1]^{|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\boldsymbol{y}_{\mathfrak{u}}, \mathbf{1}) \Delta_{s,N}(\mathcal{P}, (\boldsymbol{y}_{\mathfrak{u}}, \mathbf{1})) \, \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}.$$

Koksma–Hlawka inequality

Theorem 2.5 (Koksma–Hlawka Inequality)

Suppose $f : [0,1]^s \to \mathbb{R}$ *has bounded Hardy–Krause variation, then the QMC estimate* (1.2) *using the point set (sequence)* $\mathcal{P}$ *satisfies*

$$\left| \int_{[0,1]^s} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} - Q_{s,N} f \right| \leq \|f\|_{\mathrm{HK}} \times D^*_{s,N}(\mathcal{P}).$$

Proof. Apply the Hölder inequality to Zaremba's identity. □

An important property of this inequality is that the error bound splits into one factor that depends only on f, and another that depends only on $\mathcal{P}$.

Note: Applying Hölder's inequality with different exponents will lead to versions of the Koksma–Hlawka inequality with a different norm on f and a different type of discrepancy.

Star discrepancy bounds

Theorem 2.6 (Low-discrepancy points)

There exist families of point sets $\{\mathcal{P}_N\}_{N\in\mathbb{N}}$ such that for all $N \in \mathbb{N}$ their star discrepancy satisfies

$$D^*_{s,N}(\mathcal{P}_N) \leq C_s \frac{(\log N)^s}{N}, \tag{2.1}$$

where C_s may depend on the dimension but is independent of N.

Several well-known point sets are known to achieve the bound (2.1), e.g., Hammersley point sets, (t, m, s)-nets and lattice rules.

Theorem 2.7 (Roth's lower bound)

For any point set (sequence) $\mathcal{P} \subset [0,1]^s$, the star discrepancy is bounded from below by

$$D^*_{s,N}(\mathcal{P}) \geq c_s \frac{(\log N)^{(s-1)/2}}{N},$$

where $c_s = \frac{1}{2^{2s+4}(\log 2)^{(s-1)/2}\sqrt{(s-1)!}}$.

Classical QMC error bound

Corollary 2.8

Let $f \in \mathrm{HK}([0,1]^s)$ and let $\mathcal{P}$ be a point set (sequence) satisfying (2.1), then the error of the QMC approximation (1.2) using $\mathcal{P}$ is bounded by

$$\left| \int_{[0,1]^s} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} - Q_{s,N} f \right| \leq C_s \frac{(\log N)^s}{N} \|f\|_{\mathrm{HK}},$$

where the constant C_s depends on the dimension.

GOOD

- Asymptotically better than MC.
- Mild smoothness assumptions.
- Bound is generic, and there are several low-discrepancy point sets to choose from.
- Splits into f and $\mathcal{P}$ dependence.

BAD

- Bound still depends on the dimension.
- Bound is asymptotic. In particular, $\log(N)^s/N$ is increasing in N until $N = e^s$.
- Roth's lower bound implies that this cannot be improved.
- QMC rule cannot be tailored to a specific problem.

Summary

- The classical study of QMC rules has roots in number theory and is based on the geometric notion of discrepancy.
- Koksma–Hlawka inequality gives a bound on the integration error in terms of the discrepancy.
- There exist several points sets which are known to have low-discrepancy, and lead to quadrature convergence of the order $\mathcal{O}((\log N)^s/N)$.
- Error bounds always depend on the dimension (due to discrepancy lower bound).
- Focus was on the construction of well-distributed point sets, rather than the integration problem.
  To break the curse of dimensionality we must also consider the problem setting, i.e., the properties of f.

3. Functional analysis and Sobolev spaces

Banach & Hilbert spaces

Definition 3.1 (Banach space)

A normed space $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$ that is complete is called a *Banach* space.

Examples

$\mathbb{R}$ equipped with the the absolute value, $\mathbb{R}^s$ equipped with the Euclidean distance, $C[0,1]$ equipped with the $\max$ norm, and the Lebesgue spaces $L^p(\mathbb{R})$.

Definition 3.2 (Hilbert space)

An inner product space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ that is complete is called a *Hilbert* space.

Examples

$\mathbb{R}$ equipped with multiplication, $\mathbb{R}^s$ equipped with the Euclidean dot product, and the Lebesgue space $L^2(\mathbb{R})$ equipped with

$$\langle f, g\rangle_{0,\mathbb{R}} = \int_{\mathbb{R}} f(y)g(y)\,\mathrm{d}y.$$

Riesz representation theorem

Definition 3.3 (Dual space)

Let $\mathcal{X}$ be a Banach space. The dual space of $\mathcal{X}$ is the space of all continuous linear functionals $\ell : \mathcal{X} \to \mathbb{R}$, it is denoted by

$$\mathcal{X}^* = \{\ell : \mathcal{X} \to \mathbb{R} : \ell \text{ is linear and continuous}\}.$$

The dual space is equipped with the *dual norm*

$$\|\ell\|_{\mathcal{X}^*} = \sup_{g\in\mathcal{X}, \|g\|_{\mathcal{X}}\le 1} |\ell(g)|.$$

Theorem 3.4 (Riesz Representation Theorem)

Let $\mathcal{H}$ be a Hilbert space equipped with an inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. Then for every $\ell \in \mathcal{H}^*$ there exists a unique $f \in \mathcal{H}$ such that

$$\langle f, g\rangle_{\mathcal{H}} = \ell(g), \quad \text{for all } g \in \mathcal{H},$$

and

$$\|f\|_{\mathcal{H}} = \|\ell\|_{\mathcal{H}^*}.$$

Classical derivative spaces

Let $\Omega \subset \mathbb{R}^s$ be convex (in this course typically, $\Omega \in \{[0,1]^s, \mathbb{R}^s\}$). Let $C(\Omega) := \{f : \Omega \to \mathbb{R} : f \text{ is continuous}\}$ denote the space of continuous functions.

For higher order mixed derivatives[1] we use *multiindex* notation. Let $\boldsymbol{\alpha} \in \mathbb{N}_0^s$ be given by $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_s)$. For $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{N}_0^s$, we define the following notation:

- $\dfrac{\partial^{|\boldsymbol{\alpha}|}}{\partial \boldsymbol{y}_{\boldsymbol{\alpha}}} := \displaystyle\prod_{j=1}^{s} \frac{\partial^{\alpha_j}}{\partial y_j^{\alpha_j}}$ (when the variable is clear we will simply write $\partial^{\boldsymbol{\alpha}}$)
- $|\boldsymbol{\alpha}| = \displaystyle\sum_{j=1}^{s} \alpha_j$
- $\boldsymbol{\alpha} + \boldsymbol{\beta} = (\alpha_1 + \beta_1, \alpha_2 + \beta_2, \ldots, \alpha_s + \beta_s)$
- $\boldsymbol{\alpha} \le \boldsymbol{\beta} \iff \alpha_j \le \beta_j$ for all $j = 1, 2, \ldots, s$

For $k \in \mathbb{N}$ let $C^k(\Omega) = \{f \in C(\Omega) : \partial^{\boldsymbol{\alpha}} f \in C(\Omega) \text{ for all } |\boldsymbol{\alpha}| \le k\}$ denote the space of k-times continuously differentiable functions, let $C^\infty(\Omega) := \cap_{k=1}^{\infty} C^k(\Omega)$ denote the smooth functions and let $C_0^\infty(\Omega)$ denote the smooth functions with compact support.

[1] For first order mixed derivatives we will continue to use the set notation, because it is more convenient and consistent with the QMC literature.

Weak derivatives

Definition 3.5 (Weak derivative)

For $f \in L^2(\Omega)$, the *weak partial derivative* with respect to y_i is the function $g \in L^2(\Omega)$ such that

$$\int_\Omega g(\boldsymbol{y})\phi(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} = -\int_\Omega f(\boldsymbol{y})\frac{\partial}{\partial y_i}\phi(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}, \quad \text{for all } \phi \in C_0^\infty(\Omega).$$

When such a weak derivative exists we will denote it by $\dfrac{\partial f}{\partial y_i} = g$.

Properties of the weak derivative:

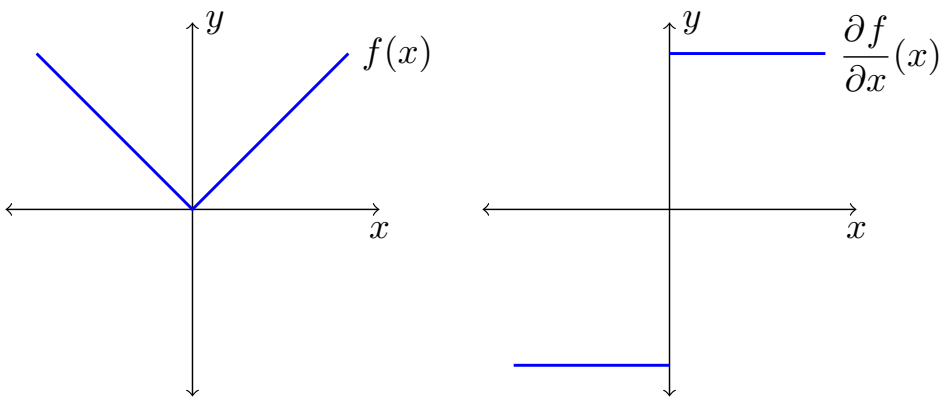
- The definition of the weak derivative easily generalises to higher order and mixed weak derivatives, in which case we use the same notation as the classical derivative.
- Weak derivatives commute.
- If the classical derivative exists, then the weak derivative coincides with it.

Weak Derivatives

Example 1

Let $\Omega = (-1, 1)$ and consider $f(x) = |x|$. The weak derivative of f is given by the Heaviside function

$$H(y) = \begin{cases} -1 & \text{if } x \leq 0, \\ 1 & \text{if } x > 0. \end{cases}$$



Example 2

The weak derivative of the Heaviside function H does not exist.

Sobolev spaces

Definition 3.6 (Sobolev space)

Let $k \in \mathbb{N}$ and $1 \leq p \leq \infty$, we define the *Sobolev space* $W^{k,p}(\Omega)$ by

$$W^{k,p}(\Omega) := \{f \in L^p(\Omega) : \partial^{\boldsymbol{\alpha}} f \in L^p(\Omega) \text{ for all } |\boldsymbol{\alpha}| \leq k\}.$$

For $1 \leq p < \infty$ we equip $W^{k,p}(\Omega)$ with the norm

$$\|f\|_{W^{k,p}(\Omega)} = \left(\sum_{|\boldsymbol{\alpha}| \leq k} \|\partial^{\boldsymbol{\alpha}} f\|^p_{L^p(\Omega)} \right)^{1/p} = \left(\sum_{|\boldsymbol{\alpha}| \leq k} \int_\Omega |\partial^{\boldsymbol{\alpha}} f(\boldsymbol{y})|^p \, \mathrm{d}\boldsymbol{y} \right)^{1/p},$$

and for $p = \infty$

$$\|f\|_{W^{k,\infty}(\Omega)} = \max_{|\alpha| \leq k} \|\partial^\alpha f\|_{L^\infty(\Omega)}.$$

Properties of Sobolev spaces

Proposition 3.7 (Alternate construction of Sobolev spaces)

The space $W^{k,p}(\Omega)$ is the completion of $C^k(\overline{\Omega})$ with respect to the norm $\|\cdot\|_{W^{k,p}(\Omega)}$.

Corollary 3.8

For all $k \in \mathbb{N}$ and $1 \le p \le \infty$ the Sobolev space $W^{k,p}(\Omega)$ is a Banach space. For the case $p = 2$, $W^{k,2}(\Omega)$ is a Hilbert space corresponding to the inner product

$$\langle f, g \rangle_{k,\Omega} = \sum_{|\boldsymbol{\alpha}| \le k} \int_\Omega \partial^{\boldsymbol{\alpha}} f(\boldsymbol{y}) \partial^{\boldsymbol{\alpha}} g(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y}.$$

We use the notation $H^k(\Omega) := W^{k,2}(\Omega)$.

Discussion of isotropic Sobolev spaces

The spaces $W^{k,p}$ are sometimes referred to as "isotropic" Sobolev spaces, because they contain weakly differentiable functions of total order k and so the smoothness is the same in all directions.

Consider the case where we are only interested in the functions that have mixed *first* order weak derivatives, i.e.,

$$\frac{\partial^{|\boldsymbol{\alpha}|} f}{\partial \boldsymbol{y}_{\boldsymbol{\alpha}}} \in L^2(\Omega) \quad \text{for all } \boldsymbol{\alpha} \in \{0,1\}^s,$$

so that we only differentiate once in each direction (cf. the conditions for the Koksma–Hlawka inequality, Theorem 2.5). The smallest isotropic space that guarantees this condition is satisfied is $H^s(\Omega)$, which also requires weak derivatives of order s in every dimension!

For high dimensions we need spaces that allow more flexible characterisations of smoothness.

Sobolev spaces of dominating mixed smoothness

Definition 3.9

Let $r \in \mathbb{N}$ and $1 \le p \le \infty$, then the Sobolev space of *dominating mixed smoothness* or order r is defined by

$$W^{\boldsymbol{r},p}_{\mathrm{mix}} := \{f \in L^p(\Omega) : \partial^{\boldsymbol{\alpha}} f \in L^p(\Omega) \text{ for all } \boldsymbol{\alpha} \in \mathbb{N}_0^s, \alpha_j \le r\}.$$

We equip $W^{\boldsymbol{r},p}_{\mathrm{mix}}(\Omega)$ with the norm

$$\|f\|_{W^{\boldsymbol{r},p}_{\mathrm{mix}}(\Omega)} = \left(\sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^s,\alpha_j\le r} \|\partial^\alpha f\|^p_{L^p(\Omega)}\right)^{1/p}, \qquad \text{for } 1 \le p < \infty,$$

$$\|f\|_{W^{\boldsymbol{r},\infty}_{\mathrm{mix}}} = \max_{\boldsymbol{\alpha}\in\mathbb{N}_0^s,\alpha_j\le r} \|\partial^\alpha f\|_{L^\infty(\Omega)}, \qquad \text{for } p = \infty.$$

For the case $p = 2$ we write $H^{\boldsymbol{r}}_{\mathrm{mix}}(\Omega) = W^{\boldsymbol{r},2}_{\mathrm{mix}}(\Omega)$, and equip $H^{\boldsymbol{r}}_{\mathrm{mix}}(\Omega)$ with the inner product

$$\langle f, g\rangle_{\boldsymbol{r},\Omega} = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^s,\alpha_j\le r} \int_\Omega \partial^{\boldsymbol{\alpha}} f(\boldsymbol{y})\partial^{\boldsymbol{\alpha}} g(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}.$$

Tensor product Hilbert spaces

Let $\Omega \subset \mathbb{R}$ and let $\mathcal{H}_1$ be a Hilbert space of functions $f : \Omega \to \mathbb{R}$ with inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}_1}$. We can construct the s-fold *tensor product Hilbert space* based on $\mathcal{H}_1$ as follows.
For $f_1, f_2, \ldots, f_s \in \mathcal{H}_1$ let $f = \prod_{j=1}^s f_j : \Omega^s \to \mathbb{R}$ be the function given by

$$f(\boldsymbol{y}) = \prod_{j=1}^s f_j(y_j), \quad \text{for all } \boldsymbol{y} \in \Omega^s.$$

For $f = \prod_{j=1}^s f_j, g = \prod_{j=1}^s g_j$ with $f_j, g_j \in \mathcal{H}_1$, define the inner product

$$\langle f, g\rangle_{\mathcal{H}_s} = \prod_{j=1}^s \langle f_j, g_j\rangle_{\mathcal{H}_1},$$

which by linearity can be extended to linear combinations of products also.
We define the s-fold tensor product Hilbert space

$$\mathcal{H}_s := \bigotimes_{j=1}^s \mathcal{H}_1$$

to be the completion of $\mathrm{span}\{f = \prod_{j=1}^s f_j : f_j \in \mathcal{H}_1\}$ with respect to the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}_s}$.

Tensor product Hilbert spaces

Example

Let $\mathcal{H}_1 = H^1[0,1]$ and consider the tensor product space $\mathcal{H}_s = \bigotimes_{j=1}^s H^1[0,1]$. For $f = \prod_{j=1}^s f_j, g = \prod_{j=1}^s g_j$ with $f_j, g_j \in H^1[0,1]$, the inner product for $\mathcal{H}_s$ is given by

$$\begin{aligned}
\langle f, g\rangle_{\mathcal{H}_s} &= \prod_{j=1}^s \left(\int_0^1 f_j(y_j) g_j(y_j)\,\mathrm{d}y_j + \int_0^1 f_j'(y_j) g_j'(y_j)\,\mathrm{d}y_j \right) \\
&= \int_{[0,1]^s} \prod_{j=1}^s \left(f_j(y_j) g_j(y_j) + f_j'(y_j) g_j'(y_j) \right) \mathrm{d}\boldsymbol{y} \\
&= \int_{[0,1]^s} \sum_{\mathfrak{u} \subseteq \{1:s\}} \prod_{j \in \mathfrak{u}} f_j'(y_j) g_j'(y_j) \prod_{j \notin \mathfrak{u}} f_j(y_j) g_j(y_j)\,\mathrm{d}\boldsymbol{y} \\
&= \sum_{\mathfrak{u} \subseteq \{1:s\}} \int_{[0,1]^s} \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\boldsymbol{y}) \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} g(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \;=\; \langle f, g\rangle_{H^{\mathbf{1}}_{\mathrm{mix}}([0,1]^s)}.
\end{aligned}$$

And so $H^{\mathbf{1}}_{\mathrm{mix}}([0,1]^s) = \bigotimes_{j=1}^s H^1[0,1]$.
We can similarly construct $H^{\boldsymbol{r}}_{\mathrm{mix}}([0,1]^s) = \bigotimes_{j=1}^s H^r[0,1]$.

Reproducing kernel Hilbert spaces

Definition 3.10

Let $\mathcal{H}$ be a Hilbert space of functions $f : \Omega \to \mathbb{R}$ equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. $\mathcal{H}$ is a *reproducing kernel Hilbert space* (RKHS) if there exists a *kernel* function $K : \Omega \times \Omega \to \mathbb{R}$ that satisfies:

K1. $K(\cdot, \boldsymbol{y}) \in \mathcal{H}$ for each $\boldsymbol{y} \in \Omega$,

K2. Reproducing property:

$$f(\boldsymbol{y}) = \langle f, K(\cdot, \boldsymbol{y})\rangle_{\mathcal{H}} \quad \text{for all } f \in \mathcal{H}, \boldsymbol{y} \in \Omega.$$

Proposition 3.11 (Properties of the kernel)

Let $\mathcal{H}$ be a RKHS, then the kernel $K : \Omega \times \Omega \to \mathbb{R}$ must also satisfy

K3. Symmetry*: $K(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{y}, \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$,*

K4. Uniqueness*: K is unique,*

K5. Positive semi-definiteness*: for all $\boldsymbol{v} \in \mathbb{R}^M$ and $\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{M-1} \in \Omega$*

$$\sum_{m,n=0}^{M-1} v_m K(\boldsymbol{y}_m, \boldsymbol{y}_n) v_n \;\geq\; 0. \tag{3.1}$$

Properties of reproducing kernel Hilbert spaces

Proposition 3.12

A Hilbert space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ is a RKHS if and only if point evaluation is a bounded linear functional, i.e., letting

$$\ell_{\boldsymbol{y}}(f) = f(\boldsymbol{y}) \quad \text{for } f \in \mathcal{H},$$

then $\ell_{\boldsymbol{y}} \in \mathcal{H}^*$ for all $\boldsymbol{y} \in \Omega$.

Proof. See accompanying notes.

Theorem 3.13 (Moore–Aronszajn Theorem [Aronszajn 1950])

Let $K : \Omega \times \Omega \to \mathbb{R}$ be a symmetric and positive semi-definite kernel, then there exists a unique RKHS $\mathcal{H}$ and inner product for which K is the kernel.

Reproducing kernel Hilbert spaces

Example

Let $\Omega = [0,1]$, then define the "anchored" inner product

$$\langle f, g\rangle_{\mathrm{anc},1} \;=\; f(1)g(1) + \int_0^1 f'(y)g'(y)\,\mathrm{d}y,$$

and the induced norm $\|f\|_{\mathrm{anc},1} = \sqrt{\langle f, f\rangle_{\mathrm{anc},1}}$.
The space

$$\mathcal{H}_1^{\mathrm{anc}} \;=\; \{f \in C[0,1] : f \text{ is absolutely continuous and } \|f\|_{\mathrm{anc},1} < \infty\} \qquad (3.2)$$

equipped with $\langle\cdot,\cdot\rangle_{\mathrm{anc},1}$ is a RKHS with kernel

$$K_1^{\mathrm{anc}}(x,y) \;=\; 1 + \min(1-x, 1-y). \qquad (3.3)$$

$\mathcal{H}_1^{\mathrm{anc}}$ is called the *first order anchored* space with anchor 1.

Tensor product reproducing kernel Hilbert spaces

Theorem 3.14

Let $\Omega \subset \mathbb{R}$, and let $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_s$ be a collection of reproducing kernel Hilbert spaces on Ω with inner products $\langle \cdot, \cdot \rangle_{\mathcal{H}_j}$, and kernel functions $K_j : \Omega \times \Omega \to \mathbb{R}$. The tensor product Hilbert space

$$\mathcal{H} := \bigotimes_{j=1}^{s} \mathcal{H}_j,$$

is also a reproducing kernel Hilbert space on Ω^s, with kernel $K : \Omega^s \times \Omega^s \to \mathbb{R}$ given by

$$K(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{s} K_j(x_j, y_j), \quad \text{for } \boldsymbol{x}, \boldsymbol{y} \in \Omega^s.$$

Proof. The case of two RKHS' is given in Theorem 8.I [Aronszajn 1950], and the s-dimensional case easily generalises. □

An application of RKHS

Proposition 3.15 (Zaremba's identity in one dimension)

Let $f \in \mathcal{H}_1^{\text{anc}}$ as defined in (3.2)*, then the error of the QMC approximation* (1.2) *using a point set $\mathcal{P}$ is*

$$\int_0^1 f(y)\,\mathrm{d}y - Q_{1,N} f = \int_0^1 f'(y) \Delta_{1,N}(\mathcal{P}, y)\,\mathrm{d}\boldsymbol{y}.$$

Proof. See accompanying notes.

Summary

- Weak derivatives generalise the concept of differentiation to functions which are integrable but not necessarily continuous.
- Sobolev spaces are spaces of functions whose weak derivatives belong to some Lebesgue space.
  They are very useful in characterising weak differentiability.
- A reproducing kernel Hilbert space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ is a Hilbert space of functions on Ω, for which there exists a kernel $K : \Omega \times \Omega \to \mathbb{R}$ that satisfies the reproducing property

$$f(\boldsymbol{y}) = \langle f, K(\cdot, \boldsymbol{y})\rangle_{\mathcal{H}}, \quad \text{for all } f \in \mathcal{H}, \boldsymbol{y} \in \Omega.$$

4. Modern QMC theory for weighted spaces

Goal: Analyse the error of a QMC approximation for all functions in some class $\mathcal{X}$, and for specific point sets obtain an error bound

$$\left| \int_{[0,1]^s} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} - Q_{s,N} f \right| \leq \mathcal{E}, \quad \text{for all } f \in \mathcal{X}.$$

Wishlist:

- Break the curse of dimensionality — $\mathcal{E}$ independent of dimension.
- Faster than MC convergence — $\mathcal{E} = \mathcal{O}(1/N)$.
- Practical — a QMC rule achieving the error bound can be constructed in practice.
- Analysis is constructive — the error analysis also informs us *how* to construct good QMC points.

For lattice rules we can achieve all of the items on our wishlist.
But first what function space $\mathcal{X}$ do we choose?

Worst-case error

Definition 4.1

Let $\mathcal{X}$ be a Banach space of functions on $[0,1]^s$. The *worst-case error* (WCE) for a QMC rule (1.2) using the point set $\mathcal{P}_N$ is defined to be

$$e(\mathcal{X}, \mathcal{P}_N) := \sup_{f \in \mathcal{X}, \|f\|_{\mathcal{X}} \leq 1} \left| \int_{[0,1]^s} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} - Q_{s,N} f \right|. \tag{4.1}$$

Note: By linearity we have the following bound on the error

$$\left| \int_{[0,1]^s} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} - Q_{s,N} f \right| \leq e(\mathcal{X}, \mathcal{P}_N) \|f\|_{\mathcal{X}}.$$

Similar to the Koksma–Hlawka inequality (Theorem 2.5), the WCE bound splits into one factor that depends on $\mathcal{P}_N$ and one factor that depends on f.

However, *both* factors depend on the function class $\mathcal{X}$, which we are free to choose. How to choose the function class?

Worst-case error in a RKHS

Proposition 4.2 (Formula for the worst-case error)

Let $\mathcal{H}$ be a RKHS of functions on $[0,1]^s$, with inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ and kernel K. Then the square worst-case error of a QMC approximation using a point set $\mathcal{P}_N = \{\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{N-1}\}$ is given by

$$e^2(K, \mathcal{P}_N) = \int_{[0,1]^{2s}} K(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{y} - \frac{2}{N} \sum_{k=0}^{N-1} \int_{[0,1]^s} K(\boldsymbol{t}_k, \boldsymbol{y}) \mathrm{d}\boldsymbol{y} + \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} K(\boldsymbol{t}_k, \boldsymbol{t}_\ell). \tag{4.2}$$

Proof. See accompanying notes. □

Error for randomised QMC

For a randomised QMC approximation $Q_{s,N}(\boldsymbol{\Delta})$, it is more appropriate to study the root-mean-square (RMS) error

$$\sqrt{\mathbb{E}_{\boldsymbol{\Delta}}\left[\left| \int_{[0,1]^s} f(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y} - Q_{s,N}(\boldsymbol{\Delta}) f \right|^2\right]}.$$

Hence, for randomly shifted QMC approximations, the error analysis should be performed in the *shift-averaged* setting.

Definition 4.3 (Shift-averaged worst-case error)

Let $\mathcal{X}$ be a Banach space of functions on $[0,1]^s$. The *shift-averaged worst case error* of a randomly shifted QMC rule using the shifted point set $\mathcal{P}_N + \boldsymbol{\Delta}$ is defined to be

$$\widehat{e}(\mathcal{X}, \mathcal{P}_N) = \sqrt{|\mathbb{E}_{\boldsymbol{\Delta}}[e^2(\mathcal{X}, \mathcal{P}_N + \boldsymbol{\Delta})]}. \tag{4.3}$$

Again, linearity yields an upper bound on the RMS error

$$\sqrt{\mathbb{E}_{\boldsymbol{\Delta}}\left[\left| \int_{[0,1]^s} f(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y} - Q_{s,N}(\boldsymbol{\Delta}) f \right|^2\right]} \le \widehat{e}(\mathcal{X}, \mathcal{P}_N) \|f\|_{\mathcal{X}}. \tag{4.4}$$

Shift-invariant kernels

Definition 4.4 (Shift invariance)

A kernel $K : [0,1]^s \times [0,1]^s \to \mathbb{R}$ is called *shift-invariant* is

$$K(\{\boldsymbol{x}+\boldsymbol{\Delta}\},\{\boldsymbol{y}+\boldsymbol{\Delta}\}) = K(\boldsymbol{x},\boldsymbol{y}) \quad \text{for all } \boldsymbol{x},\boldsymbol{y},\boldsymbol{\Delta} \in [0,1]^s.$$

For any kernel K we can define the associated *shift-invariant kernel*

$$\widehat{K}(\boldsymbol{x},\boldsymbol{y}) := \int_{[0,1]^s} K(\{\boldsymbol{x}+\boldsymbol{\Delta}\},\{\boldsymbol{y}+\boldsymbol{\Delta}\})\,\mathrm{d}\boldsymbol{\Delta}. \tag{4.5}$$

Worst-case error for a shift-invariant kernel

Proposition 4.5

Let $\mathcal{H}$ be a RKHS of functions on $[0,1]^s$, corresponding to a shift-invariant kernel K. Then the square worst-case error of a QMC approximation using a point set $\mathcal{P}_N = \{\boldsymbol{t}_0,\boldsymbol{t}_1,\ldots,\boldsymbol{t}_{N-1}\}$ is given by

$$e^2(K,\mathcal{P}_N) = -\int_{[0,1]^s} K(\boldsymbol{x},\boldsymbol{0})\,\mathrm{d}\boldsymbol{x} + \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{\ell=0}^{N-1} K(\{\boldsymbol{t}_k-\boldsymbol{t}_\ell\},\boldsymbol{0}). \tag{4.6}$$

Proof. Let $\boldsymbol{\Delta} = \boldsymbol{1}-\boldsymbol{y}$, then by shift-invariance $K(\boldsymbol{x},\boldsymbol{y}) = K(\{\boldsymbol{x}-\boldsymbol{y}\},\boldsymbol{0})$. The change of variables $\boldsymbol{u} = \{\boldsymbol{x}-\boldsymbol{y}\} \in [0,1]^s$ gives

$$\int_{[0,1]^{2s}} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} = \int_{[0,1]^{2s}} K(\{\boldsymbol{x}-\boldsymbol{y}\},\boldsymbol{0})\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} = \int_{[0,1]^s} K(\boldsymbol{u},\boldsymbol{0})\,\mathrm{d}\boldsymbol{u}.$$

Similarly, the change of variables $\boldsymbol{u} = \{\boldsymbol{t}_k-\boldsymbol{y}\} \in [0,1]^s$ gives

$$\int_{[0,1]^s} K(\boldsymbol{t}_k,\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} = \int_{[0,1]^s} K(\boldsymbol{u},\boldsymbol{0})\,\mathrm{d}\boldsymbol{u}.$$

Substituting the above three formulas into (4.2) gives the result. □

Formula for the shift-averaged worst-case error

Theorem 4.6 (Shift-averaged worst-case error in a RKHS)

Let $\mathcal{H}$ be a RKHS of functions on $[0,1]^s$ with kernel K. Then the shift-averaged worst case error is given by

$$\widehat{e}(K, \mathcal{P}_N) = e(\widehat{K}, \mathcal{P}_N), \tag{4.7}$$

where $\widehat{K}$ is the associated shift-invariant kernel as in (4.5).

Proof. See accompanying notes. □

Corollary 4.7

For a RKHS with kernel K, the shift-averaged worst-case error for $\mathcal{P}_N$ is given by

$$\widehat{e}(K, \mathcal{P}_N) = -\int_{[0,1]^s} \widehat{K}(\boldsymbol{x}, \mathbf{0}) \, \mathrm{d}\boldsymbol{x} + \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} \widehat{K}(\{\boldsymbol{t}_k - \boldsymbol{t}_\ell\}, \mathbf{0}), \tag{4.8}$$

where $\widehat{K}$ is the associated shift-invariant kernel.

Proof. Combine Proposition 4.5 and Theorem 4.6. □

Weighted Sobolev spaces

Definition 4.8 ((Anchored) weighted Sobolev space of order 1)

Let $a \in [0,1]$ and let

$$\boldsymbol{\gamma} := \{\gamma_{\mathfrak{u}} \in \mathbb{R}^+ : \mathfrak{u} \subseteq \{1:s\}\} \tag{4.9}$$

be a collection of *weight parameters*. Define $\mathcal{W}_{s,\boldsymbol{\gamma}}^{\mathrm{anc}}$ to be the *weighted Sobolev space of order 1 anchored at a* to be the space of continuous functions on $[0,1]^s$ with square-integrable mixed first derivatives, equipped with the inner product

$$\langle f, g \rangle_{\mathrm{anc},s,\boldsymbol{\gamma}} = \sum_{\mathfrak{u} \subseteq \{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left(\frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\boldsymbol{y}_{\mathfrak{u}}, \boldsymbol{a}) \right) \left(\frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} g(\boldsymbol{y}_{\mathfrak{u}}, \boldsymbol{a}) \right) \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}. \tag{4.10}$$

The (squared) norm in $\mathcal{W}_{s,\boldsymbol{\gamma}}^{\mathrm{anc}}$ is given by

$$\|f\|_{\mathrm{anc},s,\boldsymbol{\gamma}}^2 = \langle f, f \rangle_{\mathrm{anc},s,\boldsymbol{\gamma}} = \sum_{\mathfrak{u} \subseteq \{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\boldsymbol{y}_{\mathfrak{u}}, \boldsymbol{a}) \right|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}. \tag{4.11}$$

Notes:

- Spaces of this form were first introduced by [Sloan & Woźniakowski, 1998].
- Each *weight* $\gamma_{\mathfrak{u}}$ represents the relative importance of the variables $\boldsymbol{y}_{\mathfrak{u}}$.

Weighted Sobolev spaces

Definition 4.9 ((Unanchored) weighted Sobolev space of order 1)

Let $\boldsymbol{\gamma} := \{\gamma_{\mathfrak{u}} \in \mathbb{R}^+ : \mathfrak{u} \subseteq \{1:s\}\}$ be a collection of weights. Define $\mathcal{W}_{s,\boldsymbol{\gamma}}$ to be the *(unanchored) weighted Sobolev space of order 1* to be the space of continuous functions on $[0,1]^s$ with square-integrable mixed first derivatives, equipped with the inner product

$$\langle f, g\rangle_{s,\boldsymbol{\gamma}} = \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left(\int_{[0,1]^{s-|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \right) \left(\int_{[0,1]^{s-|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} g(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \right) \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}. \tag{4.12}$$

The (squared) norm in $\mathcal{W}_{s,\boldsymbol{\gamma}}^{\mathrm{anc}}$ is given by

$$\|f\|_{s,\gamma}^2 = \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \int_{[0,1]^{s-|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} f(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}_{-\mathfrak{u}} \right|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}}. \tag{4.13}$$

Weighted RKHS

The weighted anchored and unanchored spaces are reproducing kernel Hilbert spaces, with kernel given by

$$K_{s,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \eta_j(x_j, y_j), \tag{4.14}$$

where $\eta_j : [0,1]\times[0,1] \to \mathbb{R}$ is given by:

- anchored case:

$$\eta_j(x,y) = \begin{cases} a - \min(x,y), & \text{if } x,y < a, \\ \max(x,y) - a, & \text{if } x,y > a, \\ 0, & \text{otherwise.} \end{cases}$$

- unanchored case:

$$\eta_j(x,y) = \tfrac{1}{2} B_2(|x-y|) + \left(x - \tfrac{1}{2}\right)\left(y - \tfrac{1}{2}\right), \tag{4.15}$$

  where $B_2(\xi) = \xi^2 - \xi + 1/6$ is the 2nd Bernoulli polynomial.

Key point: $K_{s,\boldsymbol{\gamma}}$ depends on weights, so by (4.2) the worst-case error $e_{s,\boldsymbol{\gamma}} = e(K_{s,\boldsymbol{\gamma}})$ must also depend on the weights.

Common types of weights

Product weights: Let $\gamma_1, \gamma_2, \ldots, \gamma_s \in \mathbb{R}^+$ and let

$$\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j. \tag{4.16}$$

- e.g., $\gamma_j = j^{-\alpha}, 2^{-j}, \ldots$
- each variable is weighted independently,
- s weights in total and easy to compute with,
- ignores interactions of variables within sets.

Order-dependent: Let $\Gamma_0, \Gamma_1, \ldots, \Gamma_s \in \mathbb{R}^+$ and let

$$\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|}. \tag{4.17}$$

- e.g., $\Gamma_k = k^{-\alpha}, k!, \alpha^k, \ldots,$
- only depends on the number of variables,
- $s+1$ parameters in total and easy to compute with,
- ignores individual variables and interactions of variables.

Common types of weights

Product and order-dependent (POD): Let $\{\gamma_j\} \subset \mathbb{R}^+$ and $\{\Gamma_k\} \subset \mathbb{R}^+$

$$\gamma_{\mathfrak{u}} = \Gamma_{|\mathfrak{u}|} \prod_{j \in \mathfrak{u}} \gamma_j. \tag{4.18}$$

- each variable is weighted independently,
- $2s+1$ parameters in total and easy to compute with,
- "optimal" form of weights for certain applications.
- Motivated by UQ PDE problem [Kuo, Schwab & Sloan 2013].

General weights: $\gamma_{\mathfrak{u}} \in \mathbb{R}^+$.

- allows complete flexibility,
- 2^s weights in total,
- in practice impossible to deal with.

Shift-invariant kernel

Lemma 4.10

The shift-invariant kernel corresponding to the kernel K defined in (4.14) *is given by*

$$\widehat{K}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \widehat{\eta}_j(x_j,y_j), \tag{4.19}$$

where

$$\widehat{\eta}_j(x,y) = \begin{cases} B_2(|x-y|)+B_2(a)+\frac{1}{6}, & \text{anchored at } a, \\ B_2(|x-y|) & \text{unanchored case.} \end{cases} \tag{4.20}$$

Existence of QMC rules with dimension-independent errors

Theorem 4.11 (Dimension-independent QMC (product weights))

Let $\boldsymbol{\gamma} = \{\gamma_1, \gamma_2, \ldots\}$ be a sequence of product weights such that

$$\sum_{j=1}^{\infty} \gamma_j < \infty.$$

Then for all $s \in \mathbb{N}$ and $N \in \mathbb{N}$, there exists a point set $\mathcal{P}_N$ such that the worst-case error in $\mathcal{W}_{s,\boldsymbol{\gamma}}$ satisfies

$$e_{s,\boldsymbol{\gamma}}(\mathcal{P}_N) = e(K_{s,\boldsymbol{\gamma}},\mathcal{P}_N) \le \frac{C}{\sqrt{N}},$$

for $C < \infty$ indepedent of s.

Notes:

- Dimension-independent convergence!
- But the same rate as Monte Carlo, so we must study specific QMC rules.
- Result also holds for general weights and in the anchored space.
- Proved by averaging over *all* possible QMC points sets.

Summary

- The worst-case error provides a measure of the quality of a QMC rules in a given function space, and can be used to give simple Koksma–Hlawka-type error bound.
- In a RKHS the worst-case error has a known formula in terms of the kernel.
- Weighted RKHS spaces provide the correct setting to analyse QMC rules in. Key properties:
  - Mixed first derivatives corresponds to smoothness expected for $\mathcal{O}(1/N)$ error.
  - Weights allow characterisation of importance of different variables.
  - RKHS with known kernels.
- Worst-case error depends on the weights, which also allows to measure quality of point sets differently according to the weights.
- If the weights are summable, then we can bound the error independently of dimension.

5. Theory and construction of lattice rules

Rank-1 lattice rules

A *rank 1 lattice rule* has points given by

$$\boldsymbol{t}_k = \left\{ \frac{k\boldsymbol{z}}{N} \right\}, \qquad (5.1)$$

where

- N is the number of points,
- $\boldsymbol{z} \in \mathbb{Z}_N^s$ is the *generating vector*,
- $\{\cdot\}$ denotes the fractional part.

A *randomly shifted* lattice rule has points

$$\widehat{\boldsymbol{t}}_k = \left\{ \frac{k\boldsymbol{z}}{N} + \boldsymbol{\Delta} \right\}, \qquad (5.2)$$

where $\boldsymbol{\Delta} \sim \mathrm{Uni}[0,1)^s$.

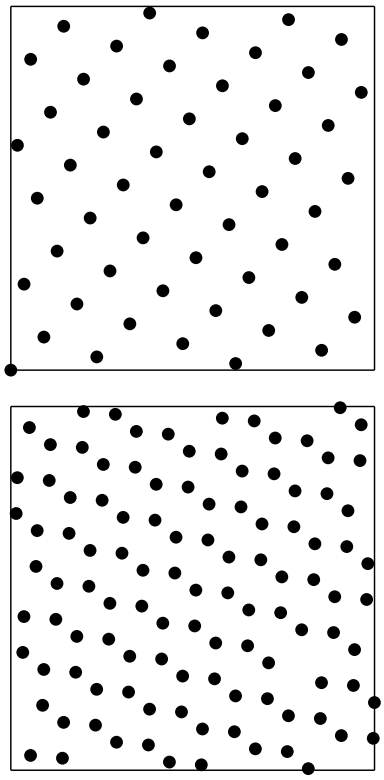


Figure: 2D lattice rule with $N = 55$, $\boldsymbol{z} = (1, 34)$ & randomly shifted lattice.

How to construct good generating vectors?

The worst-case error in $\mathcal{W}_{s,\boldsymbol{\gamma}}$ of a lattice rule with generating vector $\boldsymbol{z}$ is given by

$$e^2_{s,\boldsymbol{\gamma},N}(\boldsymbol{z}) = -1 + \frac{1}{N^2} \sum_{k,\ell=0}^{N-1} \sum_{\mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \eta_j \left(\left\{ \frac{kz_j}{N} \right\}, \left\{ \frac{\ell z_j}{N} \right\} \right), \qquad (5.3)$$

with η_j given by (4.15) or (4.20) for the shift-averaged case.
Goal: Construct a good $\boldsymbol{z}$ with small worst-case error, i.e., $\mathcal{O}(1/N)$.

Due to the structure of lattice rules,

$$t_{k,j} = \left\{ \frac{kz_j}{N} \right\} = \frac{kz_j \mod N}{N},$$

it suffices to consider only z_j in the *multiplicative group of integers modulo* N

$$\mathbb{U}_N := \{\xi \in \mathbb{N}_0 : \xi < N \text{ and } \gcd(\xi, N) = 1\}.$$

We can compute $e_{s,\boldsymbol{\gamma},N}(\boldsymbol{z})$, so why not search $\mathbb{U}_N^s$ for the best generating vector?
However, $|\mathbb{U}_N| = \varphi(N) = \mathcal{O}(N)$ (the Euler Totient function) and so the total number of possible generating vectors is $\varphi(N)^s = \mathcal{O}(N^s)$. Hence, a brute force search is infeasible.

Component-by-component construction

Algorithm 2 Component-by-component (CBC) construction

Given $s \in \mathbb{N}$, $N \in \mathbb{N}$ and weights $\boldsymbol{\gamma}$.

1: Set $z_1 \leftarrow 1$.
2: **for** $j = 2, 3, \ldots, s$ **do**
3: Choose $z_j \in \mathbb{U}_N$ to minimise the wce in dimension j *while keeping all previous components fixed*: $z_j \leftarrow \operatorname{argmin}_{\xi \in \mathbb{U}_N} e_{j,\boldsymbol{\gamma},N}(z_1, \ldots, z_{j-1}, \xi)$.
4: **end for**

Notes:

- CBC construction is extensible in s, but not in N.
- For general weights the cost of computing the worst-case error at each step $\mathcal{O}(2^s N)$. Hence, the total cost is $\mathcal{O}(s 2^s N^2)$, which is prohibitive in practice.
- The CBC algorithm can be used for any RKHS with kernel K, by simply using the formula (4.2).
- Greedy structure means that the CBC works best when the variables are ordered in decreasing importance.
- For randomly-shifted lattice rules, one simply replaces $e_{s,\boldsymbol{\gamma},N}$ with the corresponding shift-averaged worst-case error $\widehat{e}_{s,\boldsymbol{\gamma},N}$.

Shift-averaged worst-case error for product weights

Let $\boldsymbol{\gamma}$ be a collection of product weights (4.16). Then, using Corollary 4.7 and Lemma 4.10, the squared shift-averaged worst-case error in $\mathcal{W}_{s,\boldsymbol{\gamma}}$ simplifies to

$$\widehat{e}^2_{s,\boldsymbol{\gamma},N}(\boldsymbol{z}) = -1 + \frac{1}{N} \sum_{k=0}^{N-1} \prod_{j=1}^{s} \left(1 + \gamma_j B_2\left(\left\{\frac{k z_j}{N}\right\}\right)\right) \tag{5.4}$$

$$= \widehat{e}^2_{s-1,\boldsymbol{\gamma},N}(\boldsymbol{z}) + \underbrace{\frac{\gamma_s}{N} \sum_{k=0}^{N-1} B_2\left(\left\{\frac{k z_s}{N}\right\}\right) \prod_{j=1}^{s-1} \left(1 + \gamma_j B_2\left(\left\{\frac{k z_j}{N}\right\}\right)\right)}_{\theta_s(z_s)}. \tag{5.5}$$

Let $\boldsymbol{\theta}_s := [\theta_s(\xi)]_{\xi \in \mathbb{U}_N}$ and define

$$G_N := \left[B_2\left(\left\{\frac{k\xi}{N}\right\}\right)\right]_{\substack{\xi \in \mathbb{U}_N \\ k \in \mathbb{Z}_N}}, \quad \boldsymbol{p}_{s-1} := \left[\prod_{j=1}^{s-1} \left(1 + \gamma_j B_2\left(\left\{\frac{k z_j}{N}\right\}\right)\right)\right]_{k \in \mathbb{Z}_N},$$

then we can write $\boldsymbol{\theta}_s = \frac{\gamma_s}{N} G_N \boldsymbol{p}_{s-1}$.

Notes:

- Storing $\boldsymbol{p}_{s-1} \in \mathbb{R}^N$, the cost of computing $\boldsymbol{\theta}_j$ is $\mathcal{O}(N^2)$.
- Similar formulas hold for other kernels and POD weights (4.18).

CBC for randomly shifted lattice rules with product weights

Algorithm 3 CBC for product weights (matrix version)

Given $s \in \mathbb{N}$, $N \in \mathbb{N}$ and product weights $\boldsymbol{\gamma}$.

1: Compute G_N.
2: Set $z_1 \leftarrow 1$, $\boldsymbol{p}_1 \leftarrow \mathbf{1}$, and $\widehat{e}^2_{1,\boldsymbol{\gamma},N} \leftarrow \frac{\gamma_1}{N} G_N \boldsymbol{p}_0$.
3: **for** $j = 2, 3, \ldots, s$ **do**
4: Compute $\boldsymbol{\theta}_j \leftarrow \frac{\gamma_s}{N} G_N \boldsymbol{p}_{j-1}$.
5: Set $z_j \leftarrow \operatorname{argmin}_{\xi \in \mathbb{U}_N} \theta_j(\xi)$.
6: Update $\boldsymbol{p}_j \leftarrow \big(\mathbf{1} + \gamma_j G_N(z_j, :)\big) .* \boldsymbol{p}_{j-1}$.
7: Update $\widehat{e}^2_{j,\boldsymbol{\gamma},N}(\boldsymbol{z}) \leftarrow \widehat{e}^2_{j-1,\boldsymbol{\gamma},N}(\boldsymbol{z}) + \theta_j(z_j)$.
8: **end for**

Notes:

- We have used the MATLAB notation : for all entries, and $.*$ for component-wise multiplication.
- The total cost is now $\mathcal{O}(sN^2)$, which is feasible in practice.
- Can be extended to POD weights and/or other kernels.

Randomly-shifted lattice rules achieve optimal error

Theorem 5.1 (Optimal CBC error for the unanchored space)

Let $\boldsymbol{z} \in \mathbb{U}_N^s$ be a generating vector given by the CBC algorithm that minimises the shift-averaged worst case error in $\mathcal{W}_{s,\boldsymbol{\gamma}}$. Then

$$\widehat{e}^2_{s,\boldsymbol{\gamma},N}(\boldsymbol{z}) \le \frac{1}{\varphi(N)^{1/\lambda}} \left(\sum_{\emptyset \ne \mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}}^{\lambda} \left(\frac{2\zeta(2\lambda)}{(2\pi^2)^{\lambda}} \right)^{|\mathfrak{u}|} \right)^{1/\lambda} \quad \text{for } \lambda \in (\tfrac{1}{2}, 1]. \qquad (5.6)$$

Notes:

- $\zeta(x) = \sum_{k=1}^{\infty} k^{-x}$ is the Riemann zeta function, and $\zeta(x) \to \infty$ as $x \to 1^+$. Hence, the constant diverges as $\lambda \to 1/2$.
- The first proof was in [Kuo, 2003], but see also [Theorem 5.8; Dick, Kuo & Sloan 2013].
- Proof that randomly-shifted lattice rules achieve *optimal* error of $\mathcal{O}(1/N)$.
- Can be extended to the anchored case.
- A similar result for the worst-case error (i.e, for *unshifted* lattice rules) is **not** yet known.

Optimal CBC error (simplified version)

Corollary 5.2

Let N be prime, and let $\boldsymbol{\gamma}$ by a collection of product weights such that

$$\sum_{j=1}^{\infty} \gamma_j^{1/2} < \infty.$$

Then, for all $s \in \mathbb{N}$, the shift-averaged worst-case error for $\boldsymbol{z} \in \mathbb{U}_N^s$ given by the CBC algorithm satisfies

$$\widehat{e}_{s,\boldsymbol{\gamma},N}(\boldsymbol{z}) \leq C_\delta N^{-1+\delta}, \quad \text{for all } \delta > 0, \tag{5.7}$$

where C_δ is independent of s, but $C_\delta \to \infty$ as $\delta \to 0$.

Note: Randomly shifted lattice rules achieve dimension-independent *and* (almost) optimal error.

Proof. Without loss of generality assume $\gamma_j \leq 1$. For product weights, for any $c \in \mathbb{R}$, we have

$$\sum_{\mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}}^\lambda c^{|\mathfrak{u}|} = \sum_{\mathfrak{u} \subseteq \{1:s\}} \prod_{j \in \mathfrak{u}} c\gamma_j^\lambda = \prod_{j=1}^{s} (1 + c\gamma_j^\lambda) = \prod_{j=1}^{s} \exp\left(\log(1 + c\gamma_j^\lambda)\right).$$

Using $\log(1+x) \leq x$ gives

$$\sum_{\mathfrak{u} \subseteq \{1:s\}} \gamma_{\mathfrak{u}}^\lambda c^{|\mathfrak{u}|} \leq \exp\left(\sum_{j=1}^{s} c\gamma_j^\lambda\right) \leq \exp\left(c\sum_{j=1}^{\infty} \gamma_j^\lambda\right) \leq \exp\left(c\sum_{j=1}^{\infty} \gamma_j^{1/2}\right) < \infty.$$

since $\gamma_j^\lambda \leq \gamma_j^{1/2}$ for all $\lambda \in (1/2, 1]$.
Next, for N prime

$$\varphi(N) = N - 1 \geq \frac{N}{2} \quad \Longleftrightarrow \quad \frac{1}{\varphi(N)^{1/\lambda}} \leq \frac{2^{1/\lambda}}{N^{1/\lambda}}.$$

Hence, the result follows by taking $\lambda = 1/(2(1-\delta)) > 1/2$. □

Fast CBC

The bulk of the work at step j of the CBC is $G_N \boldsymbol{p}_{j-1}$ (cf. Step 4 Algorithm 3), which costs $\mathcal{O}(N^2)$.
However, $G_N \in \mathbb{R}^{\varphi(N)\times N}$ has a very special structure:

$$[G_N]_{i,j} \in \left\{ B_2(0), B_2\left(\frac{1}{N}\right), \ldots, B_2\left(\frac{N-1}{N}\right) \right\},$$

i.e., G_N only has N unique entries. So G_N can be permuted into a *circulant matrix*

$$G_N^{\text{circ}} = \begin{pmatrix} c_0 & c_1 & \cdots & c_{N-1} \\ c_1 & c_2 & \cdots & c_0 \\ \vdots & \vdots & \ddots & \vdots \\ c_{N-1} & c_0 & \cdots & c_{N-2} \end{pmatrix}, \quad (c_k = B_2(k/N)).$$

For circulant matrices matrix-vector multiplication can be performed by the FFT in $\mathcal{O}(N \log N)$ cost.
The *Fast CBC* [Nuyens, Cools 2006] uses this trick, and results in a total cost of $\mathcal{O}(sN \log N)$.

`MATLAB` code for the Fast CBC can be found on the website of **Prof. Dirk Nuyens** (KU Leuven, Belgium):

https://people.cs.kuleuven.be/~dirk.nuyens/

Embedded lattice rules [Cools, Kuo, Nuyens, 2006]

Let $N = b^m$ with b prime and $m \in \mathbb{N}$ within the range $m_{\min} \le m \le m_{\max}$. A modified version of the worst-case error that takes into account all N-point rules within this range, can be used in the CBC to obtain an *embedded lattice rule* generating vector. This embedded lattice rule can will work well for all $N = b^{m_{\min}}, b^{m_{\min}} + 1, \ldots, b^{m_{\max}}$.
The point sets for embedded lattice rules are nested:

$$\mathcal{P}_{b^{m_{\min}}} \subset \mathcal{P}_{b^{m_{\min}+1}} \subset \cdots \subset \mathcal{P}_{b^{m_{\max}}}.$$

When generating the new points we only need to add those that correspond to indices k that are not a multiple of b.
E.g., for $2^m \mapsto 2^{m+1} = N$ $(b = 2)$ the previous points are

$$\left\{ \frac{k\boldsymbol{z}}{N/2} \right\} = \left\{ \frac{(2k)\boldsymbol{z}}{N} \right\}, \quad \text{for } k = 0, 1, 2, \ldots, N/2 - 1,$$

and the new points will be given by odd indices

$$\left\{ \frac{2(k+1)\boldsymbol{z}}{N} \right\} \quad \text{for } k = 0, 1, 2, \ldots, \frac{N-3}{2}.$$

The theory for embedded lattice rules is unproven, but empirically it has been shown that the worst-case error increases by no more than 1.6.

Embedded lattice rule example

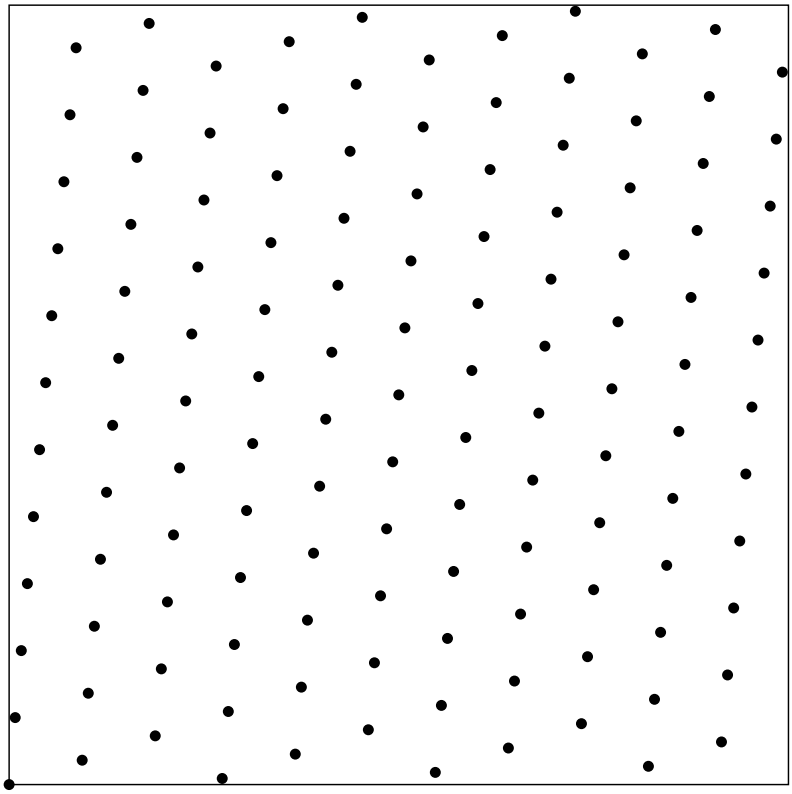


Figure: 2D embedded lattice rule in base 2 for $N = 1, 2, 4, 8, 16, 32, 64, 128$.

Summary

1. The CBC algorithm can be used to construct a randomly shifted lattice rule for which the worst-case error converges at the (almost) optimal rate of $\mathcal{O}(N^{-1+\delta})$.
2. If the square root of the weights are summable then this error is achieved independently of the dimension.
3. Fast CBC reduces the cost of constructing a generating vector to $\mathcal{O}(sN \log N)$.
4. Embedded lattice rules work well in practice for a range of N.

Off-the-shelf lattice rules

A collection of good precomputed generating vectors can be found on the website of **Prof. Frances Kuo** (UNSW Sydney, Australia):

https://web.maths.unsw.edu.au/~fkuo/lattice/index.html

6. Quasi-Monte Carlo finite element methods

Darcy flow problem with a (uniform) random coefficient

Let $D \subset \mathbb{R}^d$, for $d = 1, 2, 3$, be a bounded convex domain, and consider the elliptic PDE

$$\begin{aligned} -\nabla \cdot (a(\boldsymbol{x}, \boldsymbol{y}) \nabla u(\boldsymbol{x}, \boldsymbol{y})) &= f(\boldsymbol{x}), && \boldsymbol{x} \in D, \\ u(\boldsymbol{x}, \boldsymbol{y}) &= 0, && \boldsymbol{x} \in \partial D, \end{aligned} \tag{6.1}$$

where $\boldsymbol{x} \in D$ is the *physical variable* and $\boldsymbol{y} \sim \mathrm{Uni}([-\frac{1}{2}, \frac{1}{2}]^s)$ is a *random parameter* (i.e., $\Omega = [-\frac{1}{2}, \frac{1}{2}]$). Assume that the coefficient is of the form

$$a(\boldsymbol{x}, \boldsymbol{y}) = \phi_0(\boldsymbol{x}) + \sum_{j=1}^{s} y_j \phi_j(\boldsymbol{x}). \tag{6.2}$$

Goal: Compute the expected value of some quantity of interest (given as a linear functional $\mathcal{G}$ of the solution):

$$\mathbb{E}[\mathcal{G}(u)] = \int_{[-\frac{1}{2}, \frac{1}{2}]^s} \mathcal{G}(u(\boldsymbol{y}))\, \mathrm{d}\boldsymbol{y}. \tag{6.3}$$

Parametric weak solutions

Assumption 1

1. *There exist* $0 < a_{\min} < a_{\max} < \infty$ *such that*

$$a_{\min} \le a(\boldsymbol{x}, \boldsymbol{y}) \le a_{\max}, \quad \text{for all } \boldsymbol{x}, \in D, \boldsymbol{y} \in [-\tfrac{1}{2}, \tfrac{1}{2}]^s. \tag{6.4}$$

2. $\phi_j \in W^{1,\infty}(D)$ *for all* $j \in \mathbb{N}_0$, *and there exists a* $p \in (0, 1]$ *such that*

$$\sum_{j=1}^{\infty} \|\phi_j\|_{L^\infty(D)}^p < \infty. \tag{6.5}$$

3. $f \in L^2(D)$ *is deterministic.*
4. D *is convex.*

The parametric weak form is: Find $u(\boldsymbol{y}) \in V := H_0^1$ such that

$$\int_D a(\boldsymbol{x}, \boldsymbol{y}) \nabla u(\boldsymbol{x}, \boldsymbol{y}) \cdot \nabla v(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \int_D f(\boldsymbol{x}) v(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}, \quad \text{for all } v \in V, \tag{6.6}$$

which from Theorem II.7.1 admits a unique solution $u(\boldsymbol{y}) \in V$. Further, since $a_{\min}, a_{\max}$ are constant (i.e., in $L^\infty(\Omega)$), one can show that $u \in L^\infty(\Omega; H^2(D))$.

Approximation strategy

Finite element discretisation. Let $V_h \subset H_0^1(D)$ be a closed finite-dimensional subspace, e.g., FE space of piecewise polynomial (e.g., linear) functions corresponding to a triangulation $\mathcal{T}_h$ of D with mesh width $h > 0$ (Appendix B). The FE problem is: For $\boldsymbol{y} \in [-\frac{1}{2}, \frac{1}{2}]^s$, find $u_h(\boldsymbol{y}) \in V_h$ such that

$$\int_D a(\boldsymbol{x}, \boldsymbol{y}) \nabla u_h(\boldsymbol{x}, \boldsymbol{y}) \cdot \nabla v_h(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \int_D f(\boldsymbol{x}) v_h(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}, \quad \text{for all } v_h \in V_h, \tag{6.7}$$

which also admits a unique solution.

Quasi-Monte Carlo quadrature. Let $\mathcal{P}_N = \{\boldsymbol{t}_k\}$ be a QMC point set in $[0, 1]^s$, then apply the QMC rule to $\mathcal{G}(u)$, for $\mathcal{G} \in V^*$,

$$Q_{s,N}\mathcal{G}(u) = \frac{1}{N} \sum_{k=0}^{N-1} \mathcal{G}(u(\boldsymbol{t}_k - \tfrac{1}{2})). \tag{6.8}$$

We will use a randomly shifted lattice rule ($Q_{s,N}(\boldsymbol{\Delta})$), but in principle other QMC rules can be used.

Combined approximation:

$$\mathbb{E}[\mathcal{G}(u)] \approx Q_{s,N}\mathcal{G}(u_h) = \frac{1}{N} \sum_{k=0}^{N-1} \mathcal{G}\big(u_h(\boldsymbol{t}_k - \tfrac{1}{2})\big). \tag{6.9}$$

Error analysis

The mean-square error can be split using the triangle inequality into

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\Delta}}\left[|\mathbb{E}[\mathcal{G}(u)] - Q_{s,N}(\boldsymbol{\Delta})\mathcal{G}(u_h)|^2\right]\\ &\quad\lesssim \underbrace{\mathbb{E}_{\boldsymbol{\Delta}}\left[|\mathbb{E}[\mathcal{G}(u) - Q_{s,N}(\boldsymbol{\Delta})\mathcal{G}(u)|^2\right]}_{\text{QMC error}} + \underbrace{\mathbb{E}_{\boldsymbol{\Delta}}\left[|Q_{s,N}(\boldsymbol{\Delta})\mathcal{G}(u-u_h)|^2\right]}_{\text{FE error}}\end{aligned}$$

FE error. Again, since $1/a_{\min}, a_{\max} < \infty$ (6.4), by Theorem II.7.5, also eq. (II.7.5), for $\mathcal{G} \in L^2(D)$, for piecewise linear finite elements we have

$$\|u - u_h\|_{L^\infty(\Omega;H_0^1(D))} \le C_1 h \quad \text{and} \quad \|\mathcal{G}(u) - \mathcal{G}(u_h)\|_{L^\infty(\Omega;\mathbb{R})} \le C_2 h^2, \quad (6.10)$$

and under assumption (6.5) C_1, C_2 are independent of s.
Hence, by the triangle inequality and the uniform bound (6.10)

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\Delta}}\left[|Q_{s,N}(\boldsymbol{\Delta})\mathcal{G}(u-u_h)|^2\right] &\le \mathbb{E}_{\boldsymbol{\Delta}}\left[(Q_{s,N}(\boldsymbol{\Delta})|\mathcal{G}(u-u_h)|)^2\right]\\ &\le \|\mathcal{G}(u) - \mathcal{G}(u_h)\|^2_{L^\infty(\Omega;\mathbb{R})} \le C_2^2 h^4.\end{aligned}$$

QMC error. To use the CBC error bound (5.6), we must show that $\mathcal{G}(u) \in \mathcal{W}_{s,\boldsymbol{\gamma}}$.

Bounds on the stochastic derivatives

Theorem 6.1

Let Assumption 1 hold. Then, for all $\mathfrak{u} \subseteq \{1:s\}$, the stochastic derivatives satisfy

$$\sup_{\boldsymbol{y}\in[-\frac{1}{2},\frac{1}{2}]^s} \left\|\frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} u(\boldsymbol{y})\right\|_V \le \frac{\|f\|_{L^2(D)}}{a_{\min}} |\mathfrak{u}|! \prod_{j\in\mathfrak{u}} \beta_j, \quad (6.11)$$

where

$$\beta_j := \frac{\|\phi_j\|_{L^\infty(D)}}{a_{\min}}. \quad (6.12)$$

Furthermore, for $\mathcal{G} \in V^*$ we have $\mathcal{G}(u) \in \mathcal{W}_{s,\boldsymbol{\gamma}}$.

Proof. See accompanying notes, or [Cohen, DeVore & Schwab 2010]. □

Notes:

- Constant in derivative bounds are independent of dimension.
- Since $V_h \subset V$ the same result holds for u_h.
- Decay of importance of coefficient functions (6.4) implies the decaying importance of derivatives.

QMC error

Theorem 6.2 (Kuo, Schwab & Sloan 2012)

Let $\mathcal{G} \in V^*$, let Assumption 1 hold, and if $p = 1$, assume further that $\sum_{j=1}^{\infty} \beta_j < \sqrt{6}$. Define $\boldsymbol{\gamma}$ to be a collection of POD weights given by

$$\gamma_{\mathfrak{u}} = \left(|\mathfrak{u}|! \prod_{j \in \mathfrak{u}} \sqrt{6} \beta_j \right)^{2-p}, \quad \text{with} \quad \beta_j = \frac{\|\phi_j\|_{L^\infty(D)}}{a_{\min}}. \tag{6.13}$$

Then the RMS error of a CBC-generated randomly shifted lattice rule approximation satisfies

$$\sqrt{\mathbb{E}_{\boldsymbol{\Delta}} \left[|\mathbb{E}[\mathcal{G}(u)] - Q_{s,N}(\boldsymbol{\Delta}) \mathcal{G}(u)|^2 \right]} \leq C_\eta N^{-\eta}, \tag{6.14}$$

where C_η is independent of s and

$$\eta = \begin{cases} 1 - \delta \quad \text{for } 0 < \delta < p/2, & \text{if } p \in (0, 2/3], \\ 1/p - 1/2 & \text{if } p \in (2/3, 1]. \end{cases}$$

Sketch proof.

1. Start with the error bound $\widehat{e}_{s,\boldsymbol{\gamma},N}(\boldsymbol{z}) \|\mathcal{G}(u)\|_{s,\boldsymbol{\gamma}}$.
2. Substitute the CBC error bound (5.6) and the derivative bounds (6.11) to bound the norm.
3. The choice of weights (6.13) minimise this bound for a given λ.
4. Choose λ according to p such that assumption (6.5) ensures that the constant is independent of s.

Notes:

- The total QMC FE error is then $\mathcal{O}(h^2 + N^{-\eta})$, and the complexity is $\text{Cost} = \mathcal{O}(\varepsilon^{-1-\delta-d/2})$ (best case, $\delta > 0$).
- Key ingredients are the derivative bounds and the choice of the weights.
- POD weights are essential — theoretical and practical results with product weights or incorrect decay of weights are sub-optimal.
- The summability assumption (6.5) may seem restrictive, but is satisfied by random fields used in practice (cf. Section II.6), and also is common for other methods.
- A similar result also holds for the Gaussian random fields/log-normal coefficients as discussed in Section II.7.

QMCFEM example

$$\begin{aligned} -\nabla \cdot (a(\boldsymbol{x},\boldsymbol{y})\nabla u(\boldsymbol{x},\boldsymbol{y})) &= f(\boldsymbol{x}), && \boldsymbol{x} \in D = (0,1)^2, \\ u(\boldsymbol{x},\boldsymbol{y}) &= 0, && \boldsymbol{x} \in \partial D, \end{aligned}$$

where for $q > 4/3$

$$a(\boldsymbol{x},\boldsymbol{y}) = 1 + \sum_{j=1}^{s} y_j \phi_j(\boldsymbol{x}), \quad \phi_j(x_1,x_2) = \frac{1}{1+(j\pi)^q} \sin(j\pi x_1)\sin((j+1)\pi x_2).$$

We have

$$\|\phi_j\|_{L^\infty(D)} = \frac{1}{1+(j\pi)^q} < (j\pi)^{-q},$$

so Assumption 1 holds with $p > 1/q$.
Our goal is to compute

$$\mathbb{E}[\mathcal{G}(u)], \quad \text{where} \quad \mathcal{G}(v) = \int_{[\frac{1}{8},\frac{3}{8}]^2} v(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

using a P_1 FEM and a CBC generated randomly shifted lattice rule.

QMCFEM example

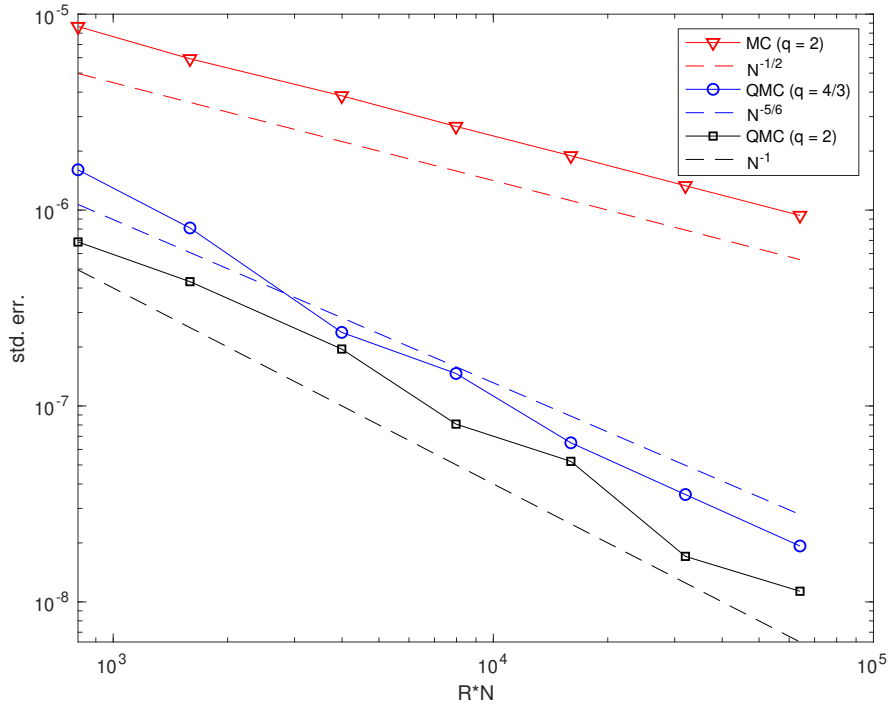


Figure: QMC and Monte Carlo convergence for PDE problem for $q = 4/3, 2$ ($s = 32$, $h = 1/64$, $R = 8$, $N = 101, 199, 499, 997, 1999, 4001, 8009$).

Summary

- QMC and FE methods can be combined to tackle stochastic PDE problems in UQ.
- Under a standard summability assumption on the stochastic coefficient the error behaves like $\mathcal{O}(h^2 + N^{-1+\delta})$.
- QMC FE methods have also been successfully applied to more general UQ problems: log-normal random fields [Graham, Kuo, Nuyens, **RS**, Sloan 2011]; eigenvalue problems [**AG**, Graham, Kuo, **RS**, Sloan 2019]; wave equations [Ganesh, Kuo, Sloan 2020] ...
- Stochastic PDE problems often admit higher regularity, which allows QMC to be applied, and perform better than MC.

7. QMC on $\mathbb{R}^s$

Integration on $\mathbb{R}^s$

Suppose we wish to compute

$$\int_{\mathbb{R}^s} g(\boldsymbol{y})\pi(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}, \tag{7.1}$$

where $\pi : \mathbb{R}^s \to \mathbb{R}^+$ is a probability density, e.g., a multivariate normal pdf.

How do we apply QMC to approximate (7.1)?

Map back to the unit cube by $\boldsymbol{\Phi} : \mathbb{R}^s \to [0,1]^s$:

$$\mathbb{R}^s \overset{\boldsymbol{\Phi}}{\mapsto} [0,1]^s, \quad \text{or equivalently } \boldsymbol{t}_k \mapsto \Phi^{-1}(\boldsymbol{t}_k),$$

e.g., take $\boldsymbol{\Phi}$ to be the Rosenblatt transform. Then by change of variables

$$\int_{\mathbb{R}^s} g(\boldsymbol{y})\pi(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} = \int_{[0,1]^s} \underbrace{g(\boldsymbol{\Phi}^{-1}(\boldsymbol{u}))|\det(D\boldsymbol{\Phi}^{-1}(\boldsymbol{u}))|\pi(\boldsymbol{\Phi}^{-1}(\boldsymbol{u}))}_{f(\boldsymbol{u})}\,\mathrm{d}\boldsymbol{u},$$

and we can apply QMC to the transformed integrand f.

Difficulties:

- For a general density π, computing $\boldsymbol{\Phi}^{-1}$ is extremely difficult.
- Typically, f does not belong to the weighted Sobolev spaces from Section 4.

Inverse cdf sampling

Consider (7.1) for the case of a product density $\pi = \prod_{j=1}^s \rho_j$, where each $\rho_j : \mathbb{R} \to \mathbb{R}^+$ is probability density with cdf given by $\Phi_j : \mathbb{R} \to [0,1]$

$$\Phi_j(x) = \int_{-\infty}^{x} \rho_j(t)\mathrm{d}t.$$

E.g., $\rho_j(y) = \exp(-y^2/2)/\sqrt{2\pi}$ (a standard normal pdf).

Let $\boldsymbol{\Phi}^{-1}$ be the inverse Rosenblatt (a.k.a. inverse cdf) transform

$$\boldsymbol{\Phi}^{-1}(\boldsymbol{u}) := (\Phi_1^{-1}(u_1), \Phi_2^{-1}(u_2), \ldots, \Phi_s^{-1}(u_s)),$$

then the change of variables simplifies to

$$\int_{\mathbb{R}^s} g(\boldsymbol{y})\left(\prod_{j=1}^s \rho_j(y_j)\right)\mathrm{d}\boldsymbol{y} = \int_{[0,1]^s} g(\boldsymbol{\Phi}^{-1}(\boldsymbol{u}))\,\mathrm{d}\boldsymbol{u}. \tag{7.2}$$

Mild conditions on g ensure that $g \circ \boldsymbol{\Phi}^{-1} \in \mathcal{W}_{s,\boldsymbol{\gamma}}$, such that we can apply our results for lattice from Section 5 (cf. [Nichols & Kuo, 2014]).

QMC sampling for multivariate normals

Let π be the pdf of a s-dimensional multivariate normal distribution, with mean $\boldsymbol{\mu} \in \mathbb{R}^s$ and s.p.d. covariance matrix Σ:

$$\pi(\boldsymbol{y}) \;=\; \frac{1}{\sqrt{(2\pi)^s \det(\Sigma)}} \exp\big(-\tfrac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\big)$$

π is not of product form, but we can factor the covariance matrix $\Sigma = AA^\top$ and then make the change of variables $\boldsymbol{x} = A^{-1}(\boldsymbol{y}-\boldsymbol{\mu})$. This gives

$$\int_{\mathbb{R}^s} g(\boldsymbol{y}) \frac{\exp\big(-\tfrac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\big)}{\sqrt{(2\pi)^s \det(\Sigma)}}\,\mathrm{d}\boldsymbol{y} \;=\; \int_{\mathbb{R}^s} g(A\boldsymbol{x}+\boldsymbol{\mu}) \frac{\exp(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{x})}{\sqrt{(2\pi)^s}}\,\mathrm{d}\boldsymbol{x},$$

which again fits into the product setting of (7.2).

Note that the factorisation is not unique, e.g., one could use a Cholesky, principal components or Brownian bridge factorisation.
The choice of A will of course affect the performance of a QMC rule.

Example II: Options pricing

The *Black–Scholes* model assumes that the price of an asset X_t at time t is given by the geometric Brownian motion

$$\mathrm{d}X_t \;=\; rX_t\mathrm{d}t + \sigma X_t \mathrm{d}W_t, \quad t>0, \quad X_0 \in \mathbb{R}, \tag{7.3}$$

where r is the risk-free interest rate, σ is the volatility and W_t is a standard Brownian motion ($W_t - W_s \overset{\text{i.i.d.}}{\sim} \mathrm{N}(0, t-s)$ for all $s<t$). The solution at $t \geq 0$ is

$$X_t \;=\; X_0 \exp\big((r-\tfrac{1}{2}\sigma^2)t + \sigma W_t\big). \tag{7.4}$$

Suppose we wish to estimate a fair price of some financial product involving the asset $\{X_t\}_{t\geq 0}$, e.g., a call option. This requires computing the "expected payoff",

$$\text{expected payoff} \;=\; \mathbb{E}[g(\{X_t\})],$$

where g is some "payoff" function of the asset path $\{X_t\}$.

Note: In more general settings when the Black–Scholes assumptions are not satisfied, the SDE modelling the asset price cannot be solved explicitly as in (7.4). In this case the SDE must be solved numerically as well.

Asian option

An Asian (average value) option compares the average value of the asset on $[0, T]$ to the strike price K. The payoff for a *call* (buy) option is

$$g(\{X_t\}) \;=\; C(\{X_t\}, T) \;=\; \max\left(\frac{1}{T}\int_0^T X_t \,\mathrm{d}t - K, 0\right), \tag{7.5}$$

and the payoff for a *put* (sell) option is

$$g(\{X_t\}) \;=\; P(\{X_t\}, T) \;=\; \max\left(K - \frac{1}{T}\int_0^T X_t \,\mathrm{d}t, 0\right). \tag{7.6}$$

The value of the option is the expected payoff, and the price of the option is taken as the discounted value of the option:

$$\mathbb{E}[\text{value}] \;=\; \mathbb{E}[g], \quad \text{and} \quad \text{price} \;=\; e^{-rT}\mathbb{E}[g].$$

Hence, to price an option we must compute the expected payoff, which will be a high-dimensional integral.

Note: The payoff's above do not have square-integrable mixed derivatives! Hence, such problems are not covered by the current theory (cf. Section 4). Nevertheless, we will see that QMC methods still work well.
This is typical of finance applications, because the value of an option cannot be negative

Generating paths and high-dimensional integrals

Paths $\{X_t\}_{t\geq 0}$ can be generated by discretising in time. For a stopping time $T > 0$ and $s \in \mathbb{N}$ time steps let

$$t_j \;=\; \frac{jT}{s}, \quad \text{for } j = 0, 1, 2, \ldots, s,$$

and let $(W_{t_j})_{j=1}^s \sim \mathrm{N}(0, \Sigma)$, where the covariance matrix $\Sigma \in \mathbb{R}^{s\times s}$ is

$$\Sigma_{ij} \;=\; \min(t_i, t_j) \;=\; \frac{T}{s}\min(i, j).$$

The expected payoff is then an s-dimensional integral:

$$\mathbb{E}[g] \;=\; \int_{\mathbb{R}^s} g(\boldsymbol{w}) \frac{\exp(-\frac{1}{2}\boldsymbol{w}^\top \Sigma^{-1}\boldsymbol{w})}{\sqrt{(2\pi)^s \det(\Sigma)}} \,\mathrm{d}\boldsymbol{w}. \tag{7.7}$$

Example. Asian option
For an Asian call option with payoff (7.5), using the formula (7.4) for the value of the asset, the time-discretised payoff function is

$$g_s(\boldsymbol{w}) \;=\; \max\left(\frac{1}{s}\sum_{j=1}^{s} X_0 \exp\left((r - \tfrac{1}{2}\sigma^2)\frac{jT}{s} + \sigma w_j\right) - K, 0\right) \tag{7.8}$$

Hence, to price an Asian option we must approximate (7.7) with g as in (7.8).

Pricing an Asian call option numerical results

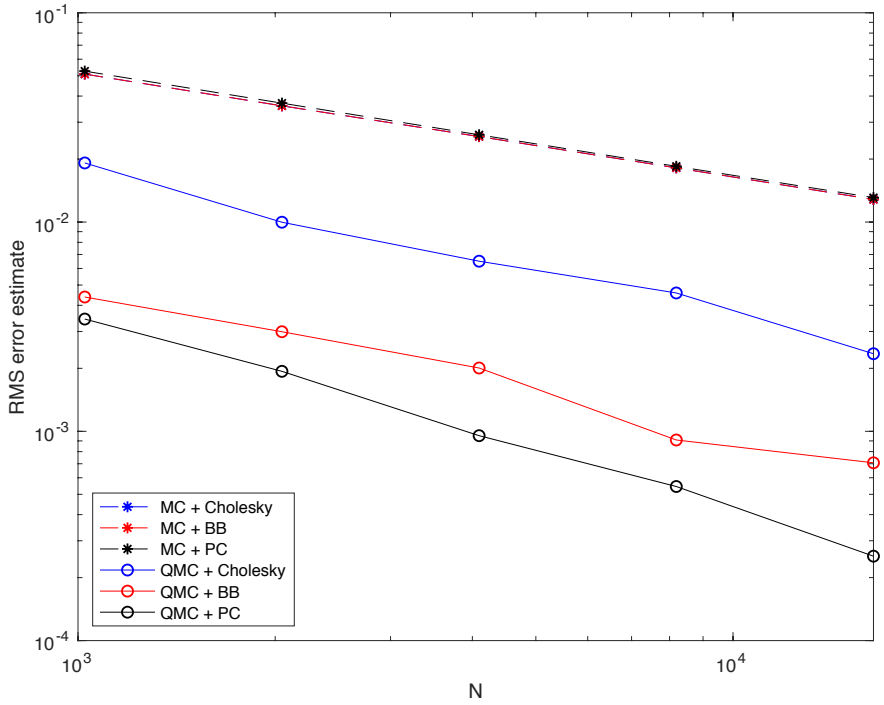


Figure: Convergence of MC and QMC (a randomly shifted lattice rule) for different factorisations of the covariance matrix (BB = Brownian bridge, PC = principal components). Asian call option with $X_0 = \$100$, $T = 256$ days and $K = \$100$.

Research in our group

- Integrals over $\mathbb{R}^s$ also arise in UQ appplications, e.g., in PDE's with a Gaussian random field as the coefficient. See: [Graham, Kuo, Nuyens, **RS** & Sloan 2011; Graham, Kuo, Nichols, **RS**, Schwab & Sloan 2015].
- As mentioned in II. Monte Carlo methods, our group is also interested in how to efficiently sample general densities π, e.g., by MCMC methods see or by constructing a surrogate approximation of the density. See: [Dodwell, Ketelsen, **RS** & Teckentrup 2015; Dolgov, Anaya-Izquierdo & Fox, **RS** 2019; Detommaso, Cui, Spantini, Marzouk & **RS** 2019].

Summary

1. QMC can be applied to integrals on $\mathbb{R}^s$ by mapping points from $[0,1]^s$.
2. The choice of mapping greatly affects the performance of the rule, and for general densities is a very difficult problem.
3. For product densities we can apply the inverse cdf componentwise, and conditions of the integrand allow us to apply our existing lattice rule results.
4. Multivariate normals ($\mathrm{N}(\boldsymbol{\mu}, \Sigma)$) can be also be handled by a factorisation $\Sigma = AA^\top$ and a change of variables.
5. The most common examples arise in computational finance, where QMC first found great success — despite such problems not being covered by the current theory.

8. Multilevel QMC

Multilevel QMC methods

As before: Let F be some quantity of interest that we cannot evaluate exactly, (e.g., the output of a PDE model), and let F_{h_ℓ} be a sequence of approximations based on a hierarchy of discretisations such that $F_{h_\ell} \to F$ as $\ell \to \infty$ (e.g., FE discretisations with meshwidth $h_\ell > 0$).
Letting

$$Y_0 \;=\; F_{h_0}, \quad Y_\ell \;=\; F_{h_\ell} - F_{h_{\ell-1}} \quad \text{for } \ell = 1, 2, \ldots$$

we again have the telescoping sum

$$\mathbb{E}[F_{h_L}] \;=\; \mathbb{E}[F_{h_0}] + \sum_{\ell=1}^{L} \mathbb{E}[F_{h_\ell} - F_{h_{\ell-1}}] \;=\; \sum_{\ell=0}^{L} \mathbb{E}[Y_\ell],$$

but now use a *QMC* rule with points $\mathcal{P}_N^{(\ell)} = \{\boldsymbol{t}_{\ell,k}\}$ to estimate the expectation of each level

$$\widehat{Y}_\ell \;\coloneqq\; Q_{s,N_\ell,\ell} Y_\ell \;=\; \frac{1}{N_\ell} \sum_{k=0}^{N_\ell - 1} Y_\ell(\boldsymbol{t}_{\ell,k}), \qquad \widehat{Q}^{\mathrm{ML}}_{L,\{N_\ell\}} F \;=\; \sum_{\ell=0}^{L} \widehat{Y}_\ell.$$

Key idea: If $\{Y_\ell\}$ are sufficiently smooth, then using QMC points should reduce the variance (RMSE) faster than Monte Carlo, and even less points are required on each level.

General multilevel QMC complexity theorem

Theorem 8.1

Let $\varepsilon < \exp(-1)$, let $F_{h_\ell} \in \mathcal{W}_{s,\boldsymbol{\gamma}}$ for $\ell = 0, 1, \ldots$, and assume that there are constants $\alpha, \beta, \eta > 0$ such that $\alpha \geq \frac{1}{2}\min\{\beta, \eta\}$ and, for all $\ell = 0, 1, \ldots$,

(M1) $|\mathbb{E}[F_{h_\ell}] - \mathbb{E}[F]| \;=\; \mathcal{O}(h_\ell^\alpha)$,

(M2') $\|Y_\ell\|^2_{s,\boldsymbol{\gamma}} = \mathcal{O}(h_\ell^\beta)$,

(M3) $\mathcal{C}_\ell = \mathsf{Cost}(Y_\ell) = \mathcal{O}(h_\ell^{-\eta})$.

Then there are L and $\{N_\ell = 2^{n_\ell}\}_{\ell=0}^L$, such that the MLQMC estimator using CBC-constructed randomly shifted lattice rules satisfies $\mathbb{E}\big[\big|\widehat{Q}^{\mathrm{ML}}_{L,\{N_\ell\}} F - \mathbb{E}[F]\big|^2\big] \leq \varepsilon^2$, and

$$\mathsf{Cost}(\widehat{Q}^{\mathrm{ML}}_{L,\{N_\ell\}}) \;=\; \begin{cases} \mathcal{O}(\varepsilon^{-1-\delta}) & \text{if } \beta > \eta, \\ \mathcal{O}(\varepsilon^{-1-\delta}|\log\varepsilon|^2), & \text{if } \beta = \eta, \\ \mathcal{O}(\varepsilon^{-1-\delta-(\eta-\beta)/\alpha}), & \text{if } \beta < \eta, \end{cases} \quad \text{for } \delta > 0. \tag{8.1}$$

Proof. Follow the minimisation argument from the proof of Theorem 5.2 in II. Monte Carlo Methods. □

Comments on MLQMC complexity

- For the PDE problem from Section 6 $\beta = 2\alpha = 4$ and $\eta \approx d$. Hence, the complexity is of the order

d	MC	MLMC	QMC	**MLQMC**
1	$\varepsilon^{-5/2}$	ε^{-2}	$\varepsilon^{-3/2-\delta}$	$\varepsilon^{-1-\delta}$
2	ε^{-3}	ε^{-2}	$\varepsilon^{-2-\delta}$	$\varepsilon^{-1-\delta}$
3	$\varepsilon^{-7/2}$	ε^{-2}	$\varepsilon^{-5/2-\delta}$	$\varepsilon^{-1-\delta}$

  See [Kuo, Schwab, Sloan 2015; Kuo, **RS**, Schwab, Sloan, Ullmann 2017].
- MLQMC improves upon MLMC by a power of ε across the board. However, the assumption (M2') is much stronger than (M2) from Theorem II.5.2.
  It requires the *spatial error* measured in the *QMC norm*. This often requires to study the mixed spatial and stochastic regularity simultaneously, e.g., for MLQMC based on P_1 FEs we require bounds on

$$\sup_{\boldsymbol{y}\in[0,1]^s} \left\| \Delta \frac{\partial^{|\mathrm{u}|} u}{\partial \boldsymbol{y}_{\mathrm{u}}} \right\|_{L^2(D)},$$

  which can be difficult to obtain in practice.
  Note that this is stronger than the bounds required for QMC FE methods, cf. Lemma 6.1, but the complexity is better than the single level QMC of $\mathcal{O}(\varepsilon^{-1-\delta-\eta/\alpha})$.

MLQMC for randomly shifted lattice rules

Strategy: Apply a randomly shifted lattice rule to Y_ℓ. Averaging over R random shifts and using an embedded lattice rule in base 2 ($N_\ell = 2^{m_\ell}$ for $m_\ell \in \mathbb{N}$) gives

$$\widehat{Y}_\ell := \frac{1}{R}\sum_{r=1}^{R} Q_{s,N_\ell}(\Delta_{\ell,r})Y_\ell, \qquad Q_{s,N_\ell}(\boldsymbol{\Delta}_{\ell,r})Y_\ell = \frac{1}{N_\ell}\sum_{k=0}^{N_\ell-1} Y_\ell\left(\left\{\frac{k\boldsymbol{z}}{N_\ell}+\boldsymbol{\Delta}_{\ell,r}\right\}\right).$$

Benefits:

- the same lattice rule (i.e., generating vector) can be used on each level ℓ,
- extra samples can be easily computed since the QMC points are nested,
- i.i.d. random shifts $\{\boldsymbol{\Delta}_{\ell,r}\}$ means that the approximations across levels are independent, and
- the variance can be estimated by the sample variances:

$$\widehat{V} = \sum_{\ell=0}^{L} \widehat{V}_\ell, \qquad \widehat{V}_\ell := \frac{1}{R(R-1)}\sum_{r=1}^{R} \left|\widehat{Y}_\ell - Q_{s,N_\ell}(\Delta_{\ell,r})Y_\ell\right|^2. \tag{8.2}$$

The bias can also be estimated by

$$\max\left\{\tfrac{1}{2}\left|\widehat{Y}_{L-1}\right|, \left|\widehat{Y}_L\right|\right\}. \tag{8.3}$$

Adaptive MLQMC for randomly shifted lattice rules

Algorithm 4 Adaptive MLQMC [Giles & Waterhouse 2007]

Given ε, an embedded lattice rule in base 2 and $R \in \mathbb{N}$.

$L \leftarrow 0$ and $N_0 \leftarrow 1$
Compute initial estimates $\widehat{Y}_0$ and $\widehat{V}_0$
while $L < 2$ or bias (8.3) $> \varepsilon/\sqrt{2}$ **do**
    while $\widehat{V} > \varepsilon^2/2$ **do**
        double N_ℓ for ℓ with largest $\widehat{V}_\ell/(2^\ell N_\ell)$
        compute $\widehat{Y}_\ell$ and $\widehat{V}_\ell$ ▷ compute new samples & **Var** estimate
    end while
    $L \leftarrow L + 1$ ▷ increase level to decrease bias
end while
$\widehat{Q}_L^{\mathrm{ML}} F = \sum_{\ell=0}^{L} \widehat{Y}_\ell$ ▷ final estimate

Note: for base 2 embedded lattice rules it is natural to double the number of samples on each level. However, this may lead to too much work being performed and so one could use a smaller factor, e.g., 1.2/1.5.

MLQMC: log-normal PDE example

$$
\begin{aligned}
-\nabla \cdot (a(\boldsymbol{x}, \boldsymbol{\omega}) \nabla u(\boldsymbol{x}, \boldsymbol{\omega})) &= f(\boldsymbol{x}), && \boldsymbol{x} \in D = (0,1)^2, \\
u(\boldsymbol{x}, \boldsymbol{\omega}) &= 0, && \boldsymbol{x} \in \partial D,
\end{aligned}
$$

where $a(\boldsymbol{x}, \boldsymbol{\omega})$ is a Gaussian random field (cf. Section II.6).

Goal: approximate

$$
\mathbb{E}[F], \qquad F(\omega) = \frac{1}{|D^*|} \int_{D^*} u(\boldsymbol{x}, \boldsymbol{\omega}) \, \mathrm{d}\boldsymbol{x}.
$$

where $D^* \subset (0,1)^2$, using a MLQMC estimator based on piecewise linear finite elements and a randomly shifted embedded lattice rule in base 2.

MLQMC numerical results for log-normal PDE problem

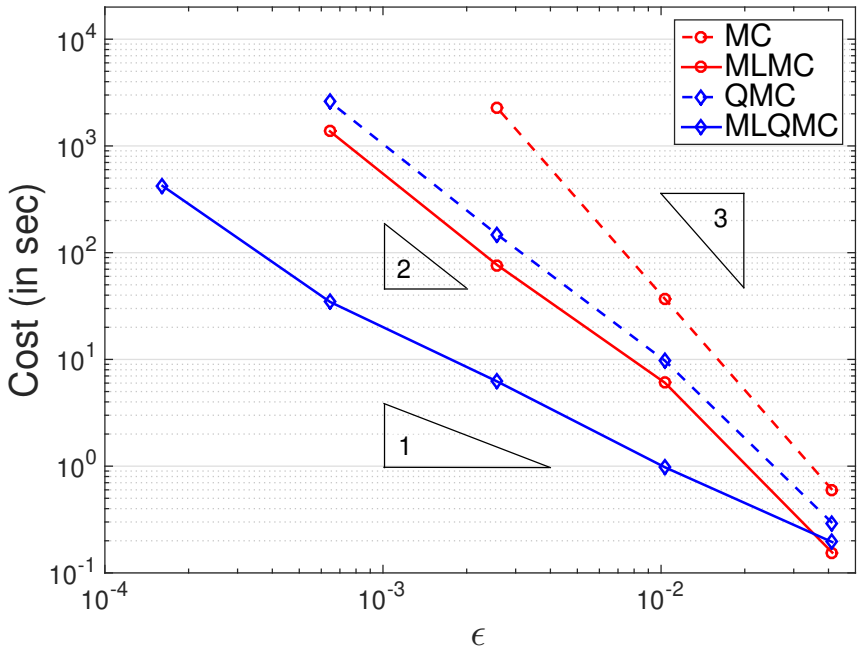


Figure: MC, MLMC, QMC and MLQMC complexity for approximating quantity of interest F for $\nu = 2.5$, $\sigma^2 = 1$ and $\lambda = 1$ [Kuo, **RS**, Schwab, Sloan & Ullmann 2017].

Summary

1. The multilevel framework allows for replacing MC samples with QMC points.
2. MLQMC gives an improved complexity when compared to (single level) QMC and also MLMC.
   Benefits of MLQMC are complementary: we gain from both the ML variance reduction *and* the faster QMC convergence.
3. Adaptive MLQMC algorithm is very useful in practice. The key ingredients are embedded lattice rules and random shifting.
4. Numerical results for the log-normal PDE problem from UQ illustrate the gains and match the theoretical complexity estimates.

9. Extensions and open problems

Extensions

Higher-order QMC [Dick 2008]

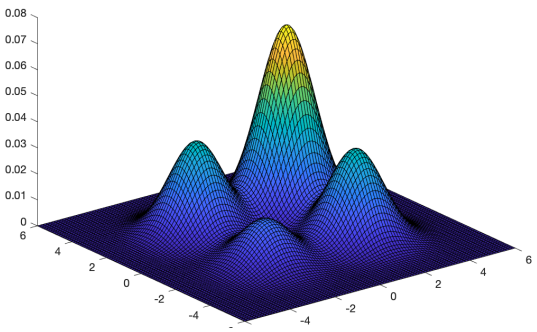
There exists polynomial lattice rules $\mathcal{P}_N \subset [0,1]^s$ that achieve

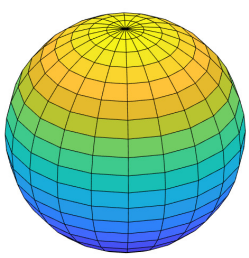
$$\left| \int_{[0,1]^s} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} - Q_{s,N} f \right| \lesssim N^{-r}, \quad \text{for some } r > 1.$$

The analysis also relies on weighted Sobolev spaces, but of *higher-order*, e.g., $\partial^{\boldsymbol{\alpha}} f$ is square-integrable for $\boldsymbol{\alpha} \in \mathbb{N}_0^s$ with $\alpha_j \leq r$.

QMC sampling for non-uniform measures and different domains

$$\int_{\mathbb{R}^s} f(\boldsymbol{y})\,\mathrm{d}\mu(\boldsymbol{y}) \qquad \text{or} \qquad \int_{\Omega} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}, \quad \Omega \subset \mathbb{R}^s$$





Open problems

1. QMC sampling for non-uniform measures and different domains
   Only certain measures and domains can be handled, e.g., product measures (as in Section 7), and simplex domains or spheres.
2. Higher-order QMC on $\mathbb{R}^s$
   Even for simple densities on $\mathbb{R}^s$ (e.g., product of standard Gaussian's) the inverse mapping back to $[0,1]^s$ destroys the smoothness required for higher order QMC.
   The current technique "truncates" $\mathbb{R}^s$ to a box $[-T,T]^s$, but results in a $\log(N)^s$ factor in the error.
3. QMC for simple discontinuities
   Developing methods to obtain optimal rate of N^{-1}, for functions that involve simple discontinuities or kinks, e.g.,

$$f(\boldsymbol{y}) = \max(g(\boldsymbol{y}),0) \quad \text{or} \quad f(\boldsymbol{y}) = \mathbb{1}(g(\boldsymbol{y})),$$

   where g smooth. Such problems are common in computational finance.

Summary

- QMC rules can achieve convergence rates of $\mathcal{O}(N^{-1})$ independent of dimension.
- Randomly shifted lattice rules that achieve this optimal rate can be efficiently constructed by the component-by-component algorithm, and they are simple to implement.
- The analysis relies on exploiting low-dimensional structure of the integrand, which is characterised by the weights γ that define the *weighted* Sobolev spaces $\mathcal{W}_{s,\gamma}$.
- QMC rules work well for integration problems coming from PDEs with random coefficients from UQ.
- QMC can be applied to integrals on $\mathbb{R}^s$ by mapping back to the unit cube. In particular, we can sample a multivariate normal with QMC points.
- Multilevel framework also works with QMC points, and here the gains are complementary.

References

Main references

[1] J. Dick, F. Y. Kuo, and I. H. Sloan. High-dimensional integration: The quasi-Monte Carlo way. *Acta Numerica*, **22**, 133–288, 2013.

[2] J. Dick, and F. Pillichshammer. *Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration*, Cambridge University Press, New York, NY, USA, 2010.

Further reading

[3] N. Aronszajn. Theory of reproducing kernels. *Trans. Am. Math. Soc.*, **68**, 337–404, 1950.

[4] A. Cohen, R. DeVore and Ch. Schwab. Convergence rates of best N-term approximations for a class of elliptic sPDEs. *Numer. Math.*, **10**, 615–646, 2010.

[5] R. Cools, F. Y. Kuo and D. Nuyens. Constructing embedded lattice rules for multivariate integration. *SIAM J. Sci. Comp.*, **28**, 2162–2188, 2006.

[6] J. Dick. Walsh spaces containing smooth functions and quasi-Monte Carlo rules of arbitrary high order. *SIAM J. Numer. Anal.*, **45**, 2141–2176, 2008.

[7] **A. D. Gilbert**, I. G. Graham, F. Y. Kuo, **R. Scheichl** and I. H. Sloan. Analysis of quasi-Monte Carlo methods for elliptic eigenvalue problems with stochastic coefficients. *Numer. Math.*, **142**, 863–915, 2019.

[8] M. B. Giles and B. J. Waterhouse. Multilevel quasi-Monte Carlo path simulation. In H. Albrecher et al. (eds.) *Advanced Financial Modelling*, Radon Series on Computational and Applied Mathematics, De Gruyter, Berlin-New York, 165–181, 2007.

[9] I. G. Graham, F. Y. Kuo, D. Nuyens, **R. Scheichl** and I. H. Sloan. Quasi-Monte Carlo methods for for elliptic PDEs with random coefficients and applications, *J. Comput. Phys.*, **230**, 3668-3694, 2011.

References (cont.)

[10] F. Y. Kuo, Ch. Schwab, and I. H. Sloan. Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients. *SIAM J. Numer. Anal.*, 50, 3351 – 3374, 2012.

[11] F. Y. Kuo, Ch. Schwab and I. H. Sloan. Multi-level quasi-Monte Carlo finite element methods for a class of elliptic PDEs with random coefficients. *Found. Comput. Math.*, **15**, 411–449, 2015.

[12] F. Y. Kuo, **R. Scheichl**, Ch. Schwab, I. H. Sloan and E. Ullmann. Multilevel quasi-Monte Carlo methods for log-normal diffusion problems. *Math. Comp.*, **86**, 2827–2860, 2017.

[13] D. Nuyens and R. Cools. Fast algorithms for component-by-component construction of ranl-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces. *Math. Comp.*, **75**, 903–920.

[14] S. Paskov and J. Traub. Faster valuation of financial derivatives. *J. Portfolio Management*, **22**, 113–120, 1995.

[15] I. H. Sloan and H. Woźniakowski. When are quasi-Monte Carlo algorithms efficient for high dimensional integration? *J. Complexity*, **14**, 1–33, 1998.

High-Dimensional Approximation and Applications in Uncertainty Quantification

IV. Sparse Grid Methods

Prof. Dr. Robert Scheichl
r.scheichl@uni-heidelberg.de

Dr. Alexander Gilbert
a.gilbert@uni-heidelberg.de

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Institut für Angewandte Mathematik, Universität Heidelberg

Summer Semester 2020

1. Sparse grid quadrature

Recap on quadrature in one dimension

Consider a sequence of 1D quadrature rules $\{Q_{1,\ell}\}$ given by

$$\int_0^1 f(y)\,\mathrm{d}y \approx Q_{1,\ell} f \;=\; \sum_{k=1}^{n_\ell} w_{\ell,k} f(t_{\ell,k})$$

where on level/precision $\ell \in \{1,2,\ldots\} (=: \mathbb{N})$

- $n_\ell \in \mathbb{N}$ is the number quadrature points (function evaluations). Typically $n_{\ell+1} > n_\ell$, e.g., $n_\ell = \mathcal{O}(2^\ell)$.
- $w_{\ell,k} \in \mathbb{R}$ for $k = 1, 2, \ldots, n_\ell$ are the quadrature *weights*, which satisfy

$$\sum_{k=1}^{n_\ell} w_{\ell,k} \;=\; 1, \quad \text{and}$$

- $t_{\ell,k} \in [0,1]$ for $k = 1, 2, \ldots, n_\ell$ are the quadrature *points*.
  As before define the *point set* or *grid* $\mathcal{P}_\ell := \{t_{\ell,k} : k = 1, 2, \ldots, n_\ell\}$.

Examples

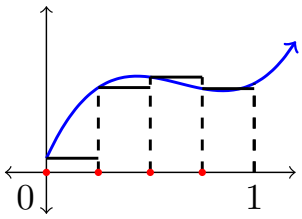
1. Monte Carlo and quasi-Monte Carlo: $w_{\ell,k} = 1/n_\ell$ and points are randomly or deterministically chosen.
2. *Polynomial-based rules*: rectangle rule, trapezoidal rule, Simpson's rule, Clenshaw–Curtis, Gauß–Legendre, etc.

Polynomial-based quadrature in one dimension

General idea: first approximate f by a piecewise polynomial interpolant p, and then integrate p exactly.
The level ℓ determines the number of points and/or also the polynomial degree, and hence, the *accuracy* or *precision* of the rule.

Rectangle rule (left)

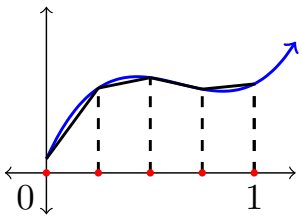
$n_\ell = 2^{\ell-1}$, $t_{\ell,k} = \dfrac{k-1}{n_\ell}$, $w_{\ell,k} = \dfrac{1}{n_\ell}$ for $\ell \in \mathbb{N}$.
If $f \in C^1[0,1]$ then error $= \mathcal{O}(n_\ell^{-1})$.



Trapezoidal rule

$n_1 = 1$, $t_{1,1} = 1/2$, $w_{1,1} = 1$, then for $\ell > 1$
$n_\ell = 2^{\ell-1} + 1$, $t_{\ell,k} = \dfrac{k-1}{n_\ell - 1}$ and

$$w_{\ell,k} = \begin{cases} \frac{1}{2(n_\ell-1)} & \text{for } k = 1, n_\ell, \\ \frac{1}{n_\ell - 1} & \text{for } k = 2, 3, \ldots, n_\ell - 1. \end{cases}$$



If $f \in C^2[0,1]$ then error $= \mathcal{O}(n_\ell^{-2})$.
Gauß–Legendre, Gauß–Patterson, Clenshaw–Curtis, Leja...

Tensor-product quadrature rules

An s-dimensional integral can be approximated by applying a 1D rule in each dimension:

$$\int_{[0,1]^s} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \approx Q^{\otimes}_{s,\boldsymbol{\ell}} f := \left(\bigotimes_{j=1}^{s} Q_{1,\ell_j} \right) f.$$

We call $Q^{\otimes}_{s,\boldsymbol{\ell}}$ the *tensor product* of the s 1D quadrature rules

$$Q_{1,\ell_j}\, g = \sum_{k_j=1}^{n_{\ell_j}} w_{\ell_j,k_j} g(t_{\ell_j,k_j}) \quad j = 1, 2, \ldots, s, \tag{1.1}$$

corresponding to the vector of levels $\boldsymbol{\ell} = (\ell_1, \ell_2, \ldots, \ell_s) \in \mathbb{N}^s$.
The tensor product quadrature rule is given explicitly by

$$\left(\bigotimes_{j=1}^{s} Q_{1,\ell_j} \right) f := \sum_{k_1=1}^{n_{\ell_1}} \sum_{k_2=1}^{n_{\ell_2}} \cdots \sum_{k_s=1}^{n_{\ell_s}} w_{\ell_1,k_1} w_{\ell_2,k_2} \cdots w_{\ell_s,k_s} f(t_{\ell_1,k_1}, t_{\ell_2,k_2}, \ldots, t_{\ell_s,k_s}). \tag{1.2}$$

Tensor product quadrature rules

Letting $\boldsymbol{n} = (n_{\ell_1}, n_{\ell_2}, \ldots, n_{\ell_s})$ and $\boldsymbol{k} = (k_1, k_2, \ldots, k_s) \in \mathbb{N}^s$ we can write

$$Q^{\otimes}_{s,\boldsymbol{\ell}} f \;=\; \sum_{\boldsymbol{k} \le \boldsymbol{n}} \boldsymbol{w}^{\otimes}_{\boldsymbol{\ell},\boldsymbol{k}} f(\boldsymbol{t}_{\boldsymbol{\ell},\boldsymbol{k}}), \qquad \boldsymbol{w}^{\otimes}_{\boldsymbol{\ell},\boldsymbol{k}} \;:=\; \prod_{j=1}^{s} w_{\ell_j,k_j}, \;\; \boldsymbol{t}_{\boldsymbol{\ell},\boldsymbol{k}} \;:=\; (t_{\ell_j,k_j})_{j=1}^{s}. \quad (1.3)$$

Notes:

- $N = \prod_{j=1}^{s} n_{\ell_j}$ points in total.
- *Full tensor product* is given by $\ell_j = \ell$ for $j = 1, 2, \ldots, s$. Then the total number of points is $N = n_\ell^s$.
- Total error is given by the error of the 1D quadrature rules, and so tensor product rules suffer from the curse of dimensionality!
  As an example, for the full tensor product and $f \in C^2([0,1]^s)$ the error is $\mathcal{O}(n_\ell^{-2}) = \mathcal{O}(N^{-2/s})$, e.g., using the trapezoidal rule.
- Key problem: Error in higher dimensions is given by the the error in 1D, but taking the product increases N exponentially!

Key idea of sparse grids: is it possible to maintain the order 1D convergence without using the "full" tensor product?

Full tensor product grid vs. sparse grid

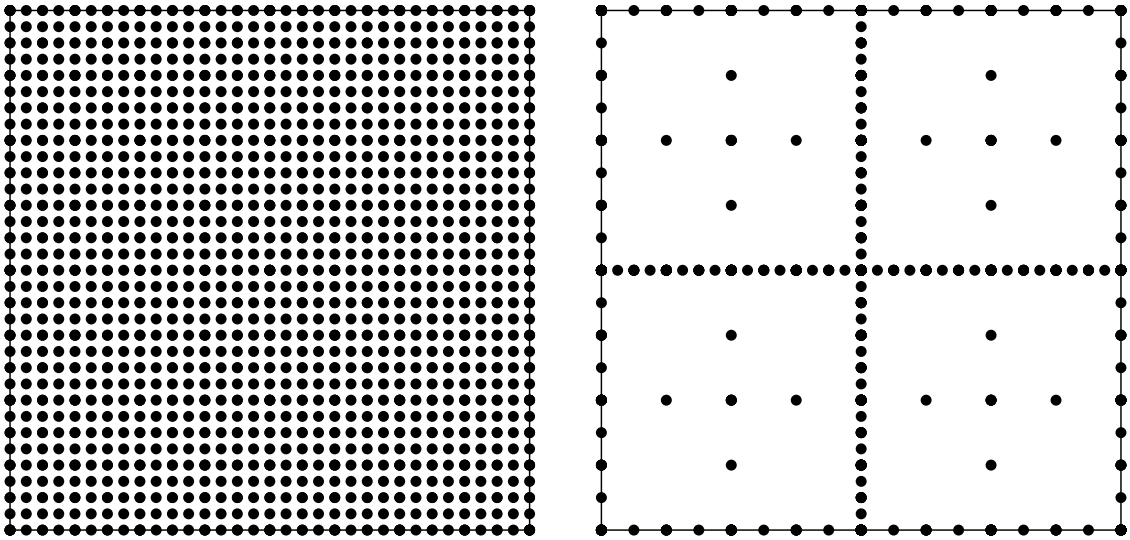


Figure: Trapezoidal rule: full tensor product grid (L) and sparse grid (R) for level $\ell = 6$ ($N_{\text{full}} = 33 \times 33 = 1089$ vs $N_{\text{SG}} = 145$).

Sparse grid quadrature

Definition 1.1 (Smolyak Operator)

Let $\{Q_{1,\ell}\}_{\ell\in\mathbb{N}}$ be a sequence of 1D quadrature rules, and for $\ell = 1, 2, \ldots$ define the *difference* quadrature rules

$$\Delta_\ell \coloneqq Q_{1,\ell} - Q_{1,\ell-1}, \qquad \text{with } Q_{1,0} \equiv 0.$$

Define the Smolyak (quadrature) operator applied to $f \in C([0,1]^s)$ by

$$Q_{s,\ell} f = \sum_{|\boldsymbol{k}|\le \ell+s-1} \left(\bigotimes_{j=1}^{s} \Delta_{k_j} \right) f, \tag{1.4}$$

where $\boldsymbol{k} \in \mathbb{N}^s$, $|\boldsymbol{k}| = \sum_{j=1}^s k_j$ and the tensor product quadrature rule is as defined in (1.2).

Two key ideas:

1. Construct an approximation that is the sum of tensor products of *differences* of 1D quadrature rules, and
2. do not use the full order ℓ tensor product grid, but restrict the grid so that the *total order* of each tensor product is at most ℓ.

Properties of Smolyak/sparse grid quadrature

- (1.4) is called both *Smolyak* quadrature and *sparse grid* quadrature. Such a quadrature approximation was first proposed in [Smolyak 1963], and the modern development uses sparse grids e.g., [Bungartz & Griebel 2004; Gerstner & Griebel 1998].
- The grid, or quadrature point set, $\mathcal{P}_{s,\ell}$ corresponding to (1.4) is called a *sparse grid*.
- A 1D rule is called *nested* if $\mathcal{P}_\ell \subset \mathcal{P}_{\ell+1}$ for all $\ell \in \mathbb{N}$. It is called *nonnested* otherwise. Similarly, a sparse grid/Smolyak approximation is called nested (nonnested) if it is based on nested (nonnested) 1D rules.
- If each order ℓ_j 1D rule uses n_{ℓ_j} points, then the total number of points is

$$N_\ell = \begin{cases} \displaystyle\sum_{|\boldsymbol{k}|\le \ell+s-1} (n_{k_1} - n_{k_1-1})(n_{k_2} - n_{k_2-1}) \cdots (n_{k_s} - n_{k_s-1}), & \text{nested,} \\ \displaystyle\sum_{|\boldsymbol{k}|\le \ell+s-1} n_{k_1} \cdot n_{k_2} \cdots n_{k_s} & \text{nonnested,} \end{cases}$$

e.g., for a nested rule if $n_\ell = \mathcal{O}(2^\ell)$, then $N = \mathcal{O}(2^\ell \cdot \ell^{s-1})$, which is a drastic reduction compared to $\mathcal{O}(2^{\ell s})$ for the full tensor product.

Properties of Smolyak/sparse grid quadrature

- The first non-trivial (nonzero) sparse grid approximation ($\ell = 1$) is

$$Q_{s,1} f = \left(\sum_{|\boldsymbol{k}| \le s} \bigotimes_{j=1}^{s} \Delta_{\ell_j} \right) f = \left(\bigotimes_{j=1}^{s} Q_{1,1} \right) f$$

  and uses $N_1 = n_1^s$ points.
  In practice it is important to use 1D rules with $n_1 = 1$, otherwise any sparse grid approximation will use an exponential number of points (since $N_\ell \ge N_1$).
- A naive sparse grid implementation, i.e., computing the sum as it is formulated in (1.4), will evaluate the function at the same points multiple times, but with different weights (due to the differences, but especially if the quadrature rules are nested).
  In practice (see Section 2), one collects the weights into a single weight, and the evaluates the function at each unique point *once only*:

$$Q_{s,\ell} f = \sum_{k=1}^{N_\ell} \widetilde{\boldsymbol{w}}_{\ell,k} f(\widetilde{\boldsymbol{t}}_{\ell,k}).$$

- Power of sparse grids: For f sufficiently smooth a sparse grid approximation has a similar order error as the *full tensor product* (cf. Section 3).

Examples of sparse grids

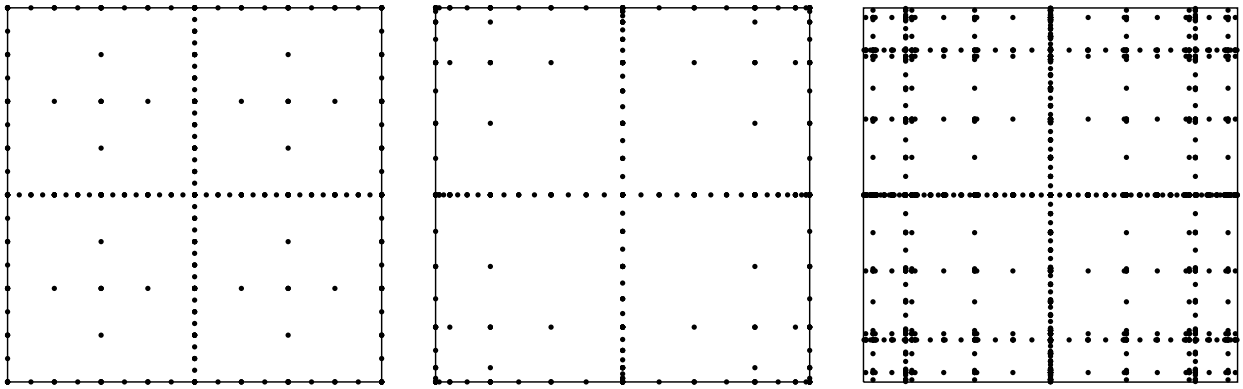


Figure: 2D sparse grids corresponding to trapezoidal, Clenshaw–Curtis and Gauß–Legendre rules for $\ell = 6$.

Numerical example — integral equation

A simplified transport problem in 1D, which models the behaviour of a particle through the one-dimensional rod $[0, 1]$, is given by the integral equation

$$x(t) \;=\; t + \int_t^1 \alpha x(y)\,\mathrm{d}y, \qquad t \in [0, 1].$$

The solution is known exactly, but can alternatively be approximated by the integral of a truncated series expansion

$$x(t) \;\approx\; x_s(t) \;=\; \int_{[0,1]^s} \sum_{k=0}^{s-1} F_k(t, \boldsymbol{y})\,\mathrm{d}\boldsymbol{y},$$

where

$$F_k(t, \boldsymbol{y}) \;=\; \alpha^k (1-t)^k \left(\prod_{j=1}^{k-1} y_j^{k-j} \right) \left(1 - (1-t) \prod_{j=1}^{k} y_j \right).$$

Sparse grid numerical results

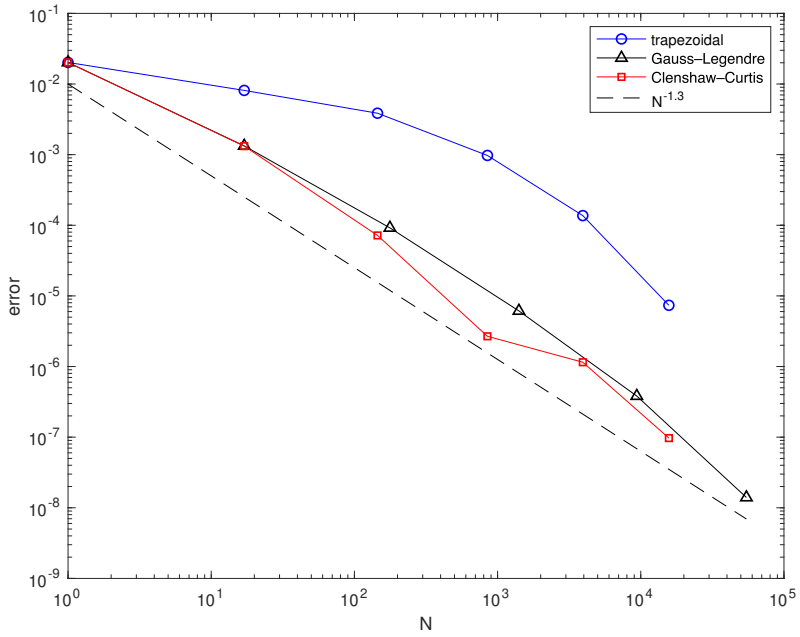


Figure: Convergence of sparse grids for the approximate solution $x_s(0)$ of integral equation with $s = 8$. Maximum level $\ell = 6$, and using trapezoidal, Clenshaw–Curtis and Gauß–Legendre 1D rules. [Source: Gerstner & Griebel 1998].

Summary

- Polynomial based quadrature rules provide great results in 1D, but when generalising to higher dimensions tensor product rules suffer the curse of dimensionality.
- Sparse grids drastically reduce the number of functions evaluations, and if the integrand is sufficiently smooth they can achieve similar convergence as the full tensor product grid.
- Two key ideas of sparse grids:
  1. Instead of taking the tensor product of order ℓ in each dimension, sparse grids restrict the quadrature grids such that the total order of the tensor product is at most ℓ.
  2. Instead of taking the tensor product of 1D rules, the Smolyak (sparse grid) rule takes the sum of tensor products of the differences.

2. Implementation of sparse grid quadrature

Comparison to full tensor product

Recall that a sparse grid approximation is given by

$$Q_{s,\ell} f \;=\; \sum_{|\boldsymbol{k}|\le \ell+s-1} \left(\bigotimes_{j=1}^{s} \Delta_{k_j} \right) f.$$

The full tensor product can be also be written in terms of the difference rules:

$$\left(\bigotimes_{j=1}^{s} Q_{1,\ell} \right) f \;=\; \sum_{|\boldsymbol{k}|_\infty \le \ell} \left(\bigotimes_{j=1}^{s} \Delta_{k_j} \right) f. \tag{2.1}$$

where $|\boldsymbol{k}|_\infty := \max_{j=1}^{s} k_j$.

The key difference is how $\boldsymbol{k}$ is restricted:

sparse grid: $|\boldsymbol{k}| = \sum_{j=1}^{s} k_j \le \ell + s - 1$

i.e., order of all 1D rules *combined* is less than $\ell + s - 1$.
E.g., if $k_i = \ell$ then $k_j = 1$ for all $j \neq i$.

full tensor product: $|\boldsymbol{k}|_\infty = \max_{j\in\{1:s\}} k_j \le \ell$.
i.e., order of the 1D rule in each dimension is less than ℓ.

A naive sparse grid implementation

A *naive* sparse grid implementation evaluates the sparse grid approximation exactly as it is formulated in (1.4), i.e.,

Algorithm 1 A naive sparse grid implementation

Given $s \in \mathbb{N}$, $\ell \in \mathbb{N}$, $\{Q_{1,k}\}_{k\in\mathbb{N}}$ and f:

1: Initialise: $Q_{s,\ell} f \leftarrow 0$
2: **for** $\boldsymbol{k} \in \mathbb{N}^s$ such that $|\boldsymbol{k}| \le \ell + s - 1$ **do**
3: $\quad Q_{s,\ell} f \leftarrow Q_{s,\ell} f + \left(\bigotimes_{j=1}^{s} \Delta_{k_j} \right) f$
4: **end for**

Problem: Due to the differences overlapping on different levels and because the 1D rules are nested, almost all quadrature points will be evaluated multiple times, but with different weights.
E.g., for the trapezoidal rule (with 1D points $\{1/2, 0, 1, 1/4, 3/4, \ldots\}$) in s dimensions the first point is $t_{\ell,1} = (1/2, 1/2, \ldots, 1/2)$ for all $\ell \in \mathbb{N}$. Hence, for every $\boldsymbol{k}$ one evaluates $f(1/2, 1/2, \ldots, 1/2)$ but with different weights.

Goal: Formulate the sparse grid approximation (1.4) such that the integrand is evaluated each unique quadrature point *once only*.

Difference grids

Define the *difference grids* $\Theta_\ell \subset [0,1]$ to be the "new" points corresponding to the 1D difference $\Delta_\ell = Q_{1,\ell} - Q_{1,\ell-1}$:

$$\Theta_\ell \;=\; \mathcal{P}_\ell \setminus \mathcal{P}_{\ell-1}, \qquad \text{with } \mathcal{P}_0 = \emptyset.$$

Note that if the rules are nonnested then $\mathcal{P}_\ell \cap \mathcal{P}_{\ell-1} = \emptyset$ so $\Theta_\ell = \mathcal{P}_\ell \setminus \mathcal{P}_{\ell-1} = \mathcal{P}_\ell$. Let the number of points in the difference grid be

$$m_\ell \;=\; \begin{cases} n_\ell - n_{\ell-1}, & (n_0 = 0) \quad \text{nested}, \\ n_\ell & \text{nonnested}. \end{cases}$$

and let the points and weights for each difference grid be

$$\Theta_\ell = \{\tau_{\ell,1}, \tau_{\ell,2}, \ldots, \tau_{\ell,m_\ell}\} \quad \text{and} \quad \omega_{\ell,1}, \omega_{\ell,2}, \ldots, \omega_{\ell,m_\ell}.$$

With this notation we can reformulate (1.4) equivalently as

$$Q_{s,\ell} f \;=\; \sum_{|\boldsymbol{k}| \le \ell+s-1} \sum_{i_1=1}^{m_{k_1}} \sum_{i_2=1}^{m_{k_2}} \cdots \sum_{i_s=1}^{m_{k_s}} \omega_{\boldsymbol{k},\boldsymbol{i}} f(\boldsymbol{\tau}_{\boldsymbol{k},\boldsymbol{i}}) \tag{2.2}$$

where $\boldsymbol{\tau}_{\boldsymbol{k},\boldsymbol{i}} = (\tau_{k_1,i_1}, \tau_{k_2,i_2}, \ldots, \tau_{k_s,i_s})$ and $\omega_{\boldsymbol{k},\boldsymbol{i}}$ is to be specified.

Computing the weights

The weight $\omega_{\boldsymbol{k},\boldsymbol{i}}$ can be computed by summing all of the weights corresponding to the unique point $\boldsymbol{\tau}_{\boldsymbol{k},\boldsymbol{i}}$ that will occur in the formulation (1.4).

Nested case

$$\omega_{\boldsymbol{k},\boldsymbol{i}} \;=\; \sum_{|\boldsymbol{k}+\boldsymbol{h}| \le \ell+2s-1} v_{(k_1+h_1),i_1} \cdot v_{(k_2+h_2),i_2} \cdots v_{(k_s+h_s),i_s}$$

where $\boldsymbol{h} \in \mathbb{N}^s$, and

$$v_{(k+h),i} \;=\; \begin{cases} \omega_{k,i} & \text{for } h = 1, \\ \omega_{(k+h-1),q} - \omega_{(k+h-2),r} & \text{for } h > 1, \text{ and } q, r \in \mathbb{N} \text{ s.t.} \\ & \tau_{k,i} = \tau_{(k+h-1),q} = \tau_{(k+h-2),r}. \end{cases}$$

Nonnested case

$$\omega_{\boldsymbol{k},\boldsymbol{i}} \;=\; \sum_{\substack{\boldsymbol{h} \in \{0,1\}^s \\ |\boldsymbol{k}+\boldsymbol{h}| \le \ell+s-1}} \prod_{j=1}^{s} (-1)^{h_j} \omega_{k_j,i_j}$$

Efficient sparse grid implementation remarks

- By dealing with the difference grids Θ_ℓ, the reformulation (2.2) only evaluates f at each unique quadrature point *once only*.
- In practice, the points and weights can be precomputed.
- The weights $\omega_{\boldsymbol{k},\boldsymbol{i}}$ may be negative in practice.
- The formulation (2.2) is not extensible in ℓ, because increasing ℓ changes *all* of the weights, i.e, to perform the approximation for $\ell+1$ one must go back and evaluate f at previous points with the new weights.
  Any sparse grid approximation suffers from this problem.
- Numerical experiments [Gerstner & Griebel 1998] suggest that it is more numerically stable to compute a sparse grid approximation dimension-wise instead of summing up to ℓ, i.e, instead of computing

$$\sum_{\kappa=s}^{\ell+s-1} \sum_{|\boldsymbol{k}|=\kappa} \sum_{i_1=1}^{m_{k_1}} \sum_{i_2=1}^{m_{k_2}} \cdots \sum_{i_s=1}^{m_{k_s}} \omega_{\boldsymbol{k},\boldsymbol{i}} f(\boldsymbol{\tau}_{\boldsymbol{k},\boldsymbol{i}}),$$

  one should compute

$$\sum_{k_1=1}^{\ell} \sum_{k_2=1}^{\ell-k_1} \cdots \sum_{k_s=1}^{\ell-k_1-\ldots-k_{s-1}} \sum_{i_1=1}^{m_{k_1}} \sum_{i_2=1}^{m_{k_2}} \cdots \sum_{i_s=1}^{m_{k_s}} \omega_{\boldsymbol{k},\boldsymbol{i}} f(\boldsymbol{\tau}_{\boldsymbol{k},\boldsymbol{i}}).$$

Combination technique

The sparse grid approximation (1.4) can equivalently be written as the sum over the tensor product of (full) 1D rules Q_{1,k_j}, instead of the differences Δ_{k_j}. The *combination technique* is the formulation of a sparse grid approximation given by

$$Q_{s,\ell} f \;=\; \sum_{\ell \le |\boldsymbol{k}| \le \ell+s-1} (-1)^{\ell+s-|\boldsymbol{k}|-1} \binom{s-1}{|\boldsymbol{k}|-\ell} \left(\bigotimes_{j=1}^{s} Q_{1,k_j} \right) f, \qquad (2.3)$$

Notes:

- Dealing only with the 1D rules Q_{1,k_j} instead of the differences Δ_{k_j} is simpler in practice.
- But still requires evaluating f at the same quadrature points multiple times with different weights. In particular, by necessity it requires more function evaluations than the equivalent formulation (2.2).
- Not extensible in ℓ, because again the "combination" weights change.

Efficiently generating indices

Sparse grid, and tensor product, rules require to generate indices (vectors) in $\boldsymbol{k}, \boldsymbol{i} \in \mathbb{N}^s$ such that $|\boldsymbol{k}| \leq \ell + s - 1$ or $i_j \leq \ell_j$.
One could of course write such sums as s nested loops, but in practice we want the flexibility to handle different dimensions.
Some basic combinatorics helps us to do this.

Tensor products

Goal: to generate all

$$\boldsymbol{i} \in \mathbb{N}^s \quad \text{such that} \quad i_j \leq n_j.$$

The indices $\boldsymbol{i} \in \mathbb{N}^s$ for the tensor product correspond to

$$\boldsymbol{i} \in \bigotimes_{j=1}^{s} \{1, 2, \ldots, n_j\},$$

which can be generated by taking all s-dimensional combinations of the vectors $(1, 2, \ldots, n_j)$ for $j = 1, 2, \ldots, s$. E.g., see the `MATLAB` function `combvec`.

Efficiently generating indices for sparse grids

Goal: to generate all

$$\boldsymbol{k} \in \mathbb{N}^s \quad \text{such that} \quad |\boldsymbol{k}| = \sum_{j=1}^{s} k_j \leq \ell + s - 1.$$

Naive method

Generate all the vectors corresponding to the full tensor product $\boldsymbol{k} \in \{1, 2, \ldots, \ell\}^s$, and then check $|\boldsymbol{k}| \leq \ell + s - 1$.
This will be very inefficient in higher dimensions.

Efficient alternative

$\boldsymbol{k} \in \mathbb{N}^s$ with $|\boldsymbol{k}| = \ell + s - 1$ is equivalent to $\boldsymbol{\kappa} \in \mathbb{N}_0^s$ with $|\boldsymbol{\kappa}| = \ell - 1$, which can in turn be represented as a collection of $\ell - 1$ stars, $\star$, and $s - 1$ bars, $|$.
E.g., for $s = 4$, $\ell = 6$ we have

$$(2, 1, 4, 2) \iff (1, 0, 3, 1) \iff \star || \star\star\star | \star$$

Hence, one can generate all $\boldsymbol{k} \in \mathbb{N}^s$ such that $|\boldsymbol{k}| = \ell + s - 1$ by generating all $(s-1)$-combinations of the integers up to $\ell + s - 2$ (see [Algorithm T, p. 5, Knuth 2005].

Summary

- The sparse grid method and a full tensor product essentially differ in how the sum over the indices $\boldsymbol{k} \in \mathbb{N}^s$ is restricted:
  - tensor products allow 1D rules up to ℓ in every dimension simultaneously, whereas
  - sparse grids restrict $\boldsymbol{k}$ such that the total order of the 1D rules in all dimensions is less than or equal to $\ell + s - 1$.
- The sparse grid approximation can be reformulated such that the function is evaluated at each unique quadrature point once only.
- The combination technique gives a reformulation of the sparse grid approximation that is given in terms of tensor products of 1D quadrature rules only, instead of the differences.
- Combinatorics tricks are often useful for generating the index vectors for sparse grids and tensor products in practice.

3. Analysis of sparse grid quadrature

Setting for sparse grid error analysis

Goal: derive bounds on the sparse grid error:

$$e_{s,\ell}(\mathcal{X}, Q_{s,\ell}) := \sup_{f\in\mathcal{X},\|f\|_{\mathcal{X}}\le 1} \left| \int_{[0,1]^s} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} - Q_{s,\ell} f \right|$$

for all f in a suitable function space $\mathcal{X}$, and ideally $e_{s,\ell} = \mathcal{O}(e_{1,\ell})$.

Function spaces

Since a sparse grid approximation is constructed by tensor products, the natural spaces in which to analyse sparse grids are tensor product spaces.
In particular, we consider the *Sobolev spaces of dominating mixed smoothness* of order $r \in \mathbb{N}$ on $[0,1]^s$ $H^{\boldsymbol{r}}_{\mathrm{mix}}([0,1]^s)$ (cf. Section III.3), which recall are the tensor product of 1D Sobolev spaces:

$$H^{\boldsymbol{r}}_{\mathrm{mix}}([0,1]^s) = \bigotimes_{j=1}^{s} H^r[0,1].$$

Alternate formulations for the Smolyak operator

Lemma 3.1

Let $s \in \mathbb{N}$ and $\ell \in \mathbb{N}$, then the Smolyak approximation operator (1.4) is also given by the combination technique

$$Q_{s,\ell} f = \sum_{\ell\le|\boldsymbol{k}|\le\ell+s-1} (-1)^{\ell+s-|\boldsymbol{k}|-1} \binom{s-1}{|\boldsymbol{k}|-\ell} \left(\bigotimes_{j=1}^{s} Q_{1,k_j} \right) f,$$

and the following dimension recursive formulas

$$Q_{s,\ell} f = \sum_{k=1}^{\ell} \left(\Delta_k \otimes Q_{s-1,\ell-k+1}\right) f, \tag{3.1}$$

$$Q_{s+1,\ell} f = \sum_{|\boldsymbol{k}|\le\ell+s-1} \left(\Delta_{k_1} \otimes \Delta_{k_2} \otimes \cdots \otimes \Delta_{k_s} \otimes Q_{1,\ell+s-|\boldsymbol{k}|}\right) f. \tag{3.2}$$

Proof. See accompanying notes. □

Abstract error bounds for Smolyak approximation

Lemma 3.2 (Wasilkowski & Woźniakowski 1995)

Let $\mathcal{H}_1$ be a Hilbert space of functions $f : [0,1] \to \mathbb{R}$, and let $\{Q_{1,\ell}\}_{\ell\in\mathbb{N}}$ be a sequence of quadrature rules such that for $\alpha \in (0,1)$ and all $\ell \in \mathbb{N}$

$$\text{(S1)} \quad e_{1,0}(\mathcal{H}_1, Q_{1,0}) := \sup_{f\in\mathcal{H}_1, \|f\|_{\mathcal{H}_1}\le 1} \left| \int_0^1 f(y)\,\mathrm{d}y \right| \le C_1,$$

$$\text{(S2)} \quad e_{1,\ell}(\mathcal{H}_1, Q_{1,\ell}) \le C_2\alpha^\ell,$$

$$\text{(S3)} \quad \|\Delta_\ell\| = \sup_{f\in\mathcal{H}_1, \|f\|_{\mathcal{H}_1}\le 1} |\Delta_\ell f| \le C_3\alpha^\ell.$$

Then, for $s \in \mathbb{N}$, the worst-case error of the Smolyak approximation (1.4) in the tensor product Hilbert space $\mathcal{H}_s = \otimes_{j=1}^s \mathcal{H}_1$ is bounded by

$$e_{s,\ell}(\mathcal{H}_s, Q_{s,\ell}) \le C_2 \max\{C_1, C_3\alpha\}^{s-1} \binom{\ell+s-1}{s-1} \alpha^\ell. \tag{3.3}$$

Proof. See [Lemma 2, Wasilkowski & Woźniakowski 1995] or accompanying notes. □

Explicit error bounds

Theorem 3.3

Let $s \in \mathbb{N}$, and let $\{Q_{1,\ell}\}_{\ell\in\mathbb{N}}$ be a sequence of 1D quadrature rules such that $n_\ell = \mathcal{O}(2^\ell)$, and their worst-case errors in $H^r[0,1]$, $r \in \mathbb{N}$, satisfy

$$e_{1,\ell}(H^r[0,1], Q_{1,\ell}) = \mathcal{O}(n_\ell^{-r}) = \mathcal{O}(2^{-\ell r}).$$

Then, for $f \in H^{\boldsymbol{r}}_{\text{mix}}([0,1]^s)$ the error of the sparse grid approximation (1.4) of order ℓ is bounded by

$$\begin{aligned} \left| \int_{[0,1]^s} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} - Q_{s,\ell} f \right| &\le C 2^{-\ell r} \ell^{s-1} \|f\|_{H^{\boldsymbol{r}}_{\text{mix}}} \\ &= \mathcal{O}\big(N_\ell^{-r} \log(N_\ell)^{(r+1)(s-1)}\big), \end{aligned} \tag{3.4}$$

where $C < \infty$ may depend on s.

Proof. The proof follows from Lemma 3.2 with $\alpha = 2^{-r}$, and then that the total number of points is $N_\ell = \mathcal{O}(2^\ell \ell^{s-1})$. □

Comments on sparse grid error

- As an example, for $f \in H^2[0,1]$ the error of the trapezoidal rule is $\mathcal{O}(n_\ell^{-2})$. Hence, for $f \in H^2_{\text{mix}}([0,1]^s)$ the error of a sparse grid approximation based on trapezoidal rules is $\mathcal{O}(N_\ell^{-2} \log(N_\ell)^{3(s-1)})$.
- Error bounds are asymptotically better than MC and QMC, but they still depend on the dimension.
  The sparse grid approximation (1.4) is *isotropic*, i.e., each dimension is treated the same (similarly for H^r_{mix}). Since we are not exploiting any dimensionwise structure of f, the dependence on the dimension is to be expected.
- We have only considered integration on $[0,1]^s$, however, Smolyak's construction can be used for many more problems, e.g., integration on $\mathbb{R}^s$ by using Gauß–Hermite quadrature in 1D, or function approximation/interpolation by taking the tensor product of 1D polynomial interpolants (cf. Section 4).
  In particular, Lemma 3.2 holds in much more general settings.

Summary

- Since a sparse grid approximation is based on tensor products, the correct setting to study their errors are tensor product spaces.
- For $f \in H^r_{\text{mix}}([0,1]^s)$ the error of a sparse grid approximation is $\mathcal{O}(N_\ell^{-r} \log(N_\ell)^{(r+1)(s-1)})$.
- A key abstract result give that the error of a sparse grid approximation is of the same order as the 1D errors.
- Since the sparse grids we have studied so far are isotropic, the errors still depend on the dimension.

4. Sparse grid interpolation

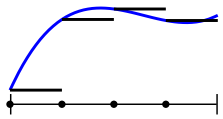
Function interpolation in one dimension

Now we wish to approximate $f : [0,1] \to \mathbb{R}$ in some finite-dimensional subspace $V_{1,\ell}$ with dimension n_ℓ, e.g., piecewise polynomials corresponding to a given grid. Define the *interpolation operator* $A_{1,\ell} : C[0,1] \to V_{1,\ell}$ by

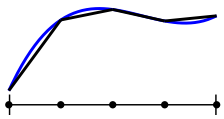
$$f(y) \approx A_{1,\ell} f(y) := \sum_{k=1}^{n_\ell} f(t_{\ell,k}) \phi_{\ell,k}(y), \tag{4.1}$$

where $\{\phi_{\ell,k}\}_{k=1}^{n_\ell}$ are a basis for $V_{1,\ell}$ and $\mathcal{P}_\ell = \{t_{\ell,k}\}_{k=1}^{n_\ell}$ is the *grid*.

Piecewise constant



Piecewise linear



Other examples

Higher order Lagrange interpolation, Legendre polynomials, Hermite polynomials on $\mathbb{R}$, ...

Approximation vs interpolation

$A_{1,\ell} f$ *interpolates* f on the grid $\mathcal{P}_\ell = \{t_{\ell,k}\}_{k=1}^{n_\ell}$:

$$A_{1,\ell} f(t_{\ell,k}) = f(t_{\ell,k}) \quad \text{for } k = 1, 2, \ldots, n_\ell.$$

But one could also consider more general *approximation* operators that give the best approximation in $V_{1,\ell}$, e.g., least-squares, best n-term ...

Piecewise linear interpolation

Let $\mathcal{P}_\ell$ be an equidistant grid on $[0,1]$ with

$$n_\ell = 2^\ell + 1, \quad h_\ell = 2^{-\ell} \text{ (meshwidth or gridsize)} \quad \text{and} \quad t_{\ell,k} = (k-1)h_\ell.$$

Then let $V_{1,\ell}$ be the space of piecewise linear functions

$$V_{1,\ell} = \left\{ v \in C[0,1] : v|_{(t_{\ell,k}, t_{\ell,k+1})} \in P_1^1(t_{\ell,k}, t_{\ell,k+1}) \text{ for } k = 1, 2, \ldots n_\ell - 1 \right\}.$$

Error in one dimension

For $f \in H^2[0,1]$, it is well-known that the interpolation errors satisfy

$$\begin{aligned} \|f - A_{1,\ell} f\|_{H^1[0,1]} &\leq C_1 h_\ell |f|_{H^2[0,1]}, \\ \|f - A_{1,\ell} f\|_{L^2[0,1]} &\leq C_2 h_\ell^2 |f|_{H^2[0,1]}, \end{aligned}$$

where $|f|_{H^2[0,1]} = \|f''\|_{L^2[0,1]}$ is the H^2 semi-norm.

Tensor product interpolation

Let $\{V_{1,\ell}\}_{\ell \in \mathbb{N}}$ be a sequence finite-dimensional subspaces of $C[0,1]$, let $n_\ell = \dim(V_{1,\ell})$ and let the corresponding 1D interpolation operators $A_{1,\ell} : C[0,1] \to V_{1,\ell}$ be given by

$$A_{1,\ell} f = \sum_{k=1}^{n_\ell} f(t_{\ell,k}) \phi_{\ell,k}.$$

Let $\boldsymbol{\ell} \in \mathbb{N}^s$ be a multiindex of levels. The *tensor product approximation space* is

$$V_{s,\boldsymbol{\ell}}^{\otimes} = \bigotimes_{j=1}^{s} V_{1,\ell_j} = \operatorname{span}\left\{ \prod_{j=1}^{s} \phi_{\ell_j,k_j} : k_j = 1, 2, \ldots, n_{\ell_j}, j = 1, 2, \ldots, s \right\},$$

and the *tensor product approximation operator* is given by

$$A_{s,\boldsymbol{\ell}}^{\otimes} f = \left(\bigotimes_{j=1}^{s} A_{1,\ell_j} \right) f = \sum_{k_1=1}^{n_{\ell_1}} \sum_{k_2=1}^{n_{\ell_2}} \cdots \sum_{k_s=1}^{n_{\ell_s}} f(\boldsymbol{t}_{\boldsymbol{\ell},\boldsymbol{k}}) \prod_{j=1}^{s} \phi_{\ell_j,k_j}, \tag{4.2}$$

where $\boldsymbol{t}_{\boldsymbol{\ell},\boldsymbol{k}} = (t_{\ell_1,k_1}, t_{\ell_2,k_2}, \ldots, t_{\ell_s,k_s})$.

Properties of tensor product interpolation

- The dimension of the tensor product approximation space is

$$N_{\boldsymbol{\ell}}^{\otimes} = \dim\left(V_{s,\boldsymbol{\ell}}^{\otimes}\right) = \prod_{j=1}^{s} \dim\left(V_{1,\ell_j}\right) = \prod_{j=1}^{s} n_{\ell_j},$$

  which also corresponds to the total number of points in the tensor product grid.
- The full tensor product is given by $\ell_j = \ell$ for $j = 1, 2, \ldots, s$, and then $\dim(\otimes_{j=1}^{s} V_1, \ell) = n_\ell^s$.
- As with quadrature, the error of the tensor product interpolation is given by the error of interpolation in one dimension.
  E.g, for $f \in H^2([0,1]^s)$ the L^2-error of the full tensor product piecewise linear interpolant is

$$\|f - A_{s,\ell}^{\otimes} f\|_{L^2([0,1]^s)} = \mathcal{O}(h_\ell^2) = \mathcal{O}\big(n_\ell^{-2}\big) = \mathcal{O}\big((N_\ell^{\otimes})^{-2/s}\big).$$

  And so tensor product interpolation also suffers from the curse of dimensionality!

Sparse grid interpolation

Definition 4.1 (Smolyak Operator)

Let $\{A_{1,\ell}\}_{\ell \in \mathbb{N}}$ be a sequence of 1D interpolations defined on the finite-dimensional subspaces $V_{1,\ell} \subset C[0,1]$, and for $\ell = 1, 2, \ldots$ define now the *difference interpolation* operators

$$\Delta_\ell := A_{1,\ell} - A_{1,\ell-1}, \qquad \text{with } A_{1,0} \equiv 0.$$

Define the Smolyak (interpolation) operator applied to $f \in C([0,1]^s)$ by

$$A_{s,\ell} f = \sum_{|\boldsymbol{k}| \le \ell+s-1} \left(\bigotimes_{j=1}^{s} \Delta_{k_j} \right) f, \tag{4.3}$$

where $\boldsymbol{k} \in \mathbb{N}^s$, $|\boldsymbol{k}| = \sum_{j=1}^{s} k_j$ and the tensor product approximation is as defined in (4.2). The corresponding approximation subspace is

$$V_{s,\ell} = \bigoplus_{|\boldsymbol{k}| \le \ell+s-1} \bigotimes_{j=1}^{s} V_{1,k_j} = \operatorname{span}\left\{ \prod_{j=1}^{s} v_{k_j} : v_{k_j} \in V_{1,k_j} \text{ for } |\boldsymbol{k}| \le \ell + s - 1 \right\}.$$

Properties of sparse grid interpolation

Define the 1D *difference* subspaces

$$W_{1,\ell} := V_{1,\ell} \setminus V_{1,\ell-1},$$

and let $m_\ell = \dim\left(W_{1,\ell}\right)$. The 1D interpolation rules are *nested* if $V_{1,\ell} \subset V_{1,\ell+1}$ and *nonnested* otherwise.
Then we can write the Smolyak approximation space as

$$V_{s,\ell} = \bigoplus_{|\boldsymbol{k}|\leq \ell+s-1} \bigotimes_{j=1}^{s} W_{1,k_j}.$$

The total number of degrees of freedom (also the number of points in the grid) is

$$N_\ell = \dim\left(V_{s,\ell}\right) = \sum_{|\boldsymbol{k}|\leq \ell+s-1} \prod_{j=1}^{s} m_{k_j}.$$

Alternate formulas for interpolation

The following formulas also hold for the Smolyak interpolation operator (4.3):

- Combination technique:

$$A_{s,\ell} f = \sum_{\ell\leq|\boldsymbol{k}|\leq \ell+s-1} (-1)^{\ell+s-|\boldsymbol{k}|-1} \binom{s-1}{|\boldsymbol{k}|-\ell} \left(\bigotimes_{j=1}^{s} A_{1,k_j}\right) f. \tag{4.4}$$

- Dimension recursive formulas:

$$A_{s,\ell} f = \sum_{k=1}^{\ell} \left(\Delta_k \otimes A_{s-1,\ell-k+1}\right) f, \tag{4.5}$$

$$A_{s+1,\ell} f = \sum_{|\boldsymbol{k}|\leq \ell+s-1} \left(\Delta_{k_1} \otimes \Delta_{k_2} \otimes \cdots \otimes \Delta_{k_s} \otimes A_{1,\ell+s-|\boldsymbol{k}|}\right) f, \tag{4.6}$$

  where $k \in \mathbb{N}$ and $\boldsymbol{k} \in \mathbb{N}^s$.

- Full tensor product in terms of the differences:

$$\left(\bigotimes_{j=1}^{s} A_{1,\ell}\right) f = \sum_{|\boldsymbol{k}|_\infty \leq \ell} \left(\bigotimes_{j=1}^{s} \Delta_{k_j}\right) f, \tag{4.7}$$

  where $|\boldsymbol{k}|_\infty = \max_{j=1}^{s}\left\{k_j\right\}$.

Analysis of sparse grid interpolation

Goal: derive bounds on the sparse grid interpolation error:

$$\|f - A_{s,\ell} f\|$$

in some appropriate norm, e.g., L^∞ or L^2.

As before, we can define the *worst-case error* for approximation by

$$e_{s,\ell}(\mathcal{X}, \mathcal{Y}, A_{s,\ell}) := \sup_{f \in \mathcal{X}, \|f\|_{\mathcal{X}} \leq 1} \|f - A_{s,\ell} f\|_{\mathcal{Y}},$$

where $f \in \mathcal{X}$, and we measure the error in the $\mathcal{Y}$ norm.
Again we want $e_{s,\ell} = \mathcal{O}(e_{1,\ell})$.

Function spaces

Again, we consider $f \in H^r_{\text{mix}}([0,1]^s)$, the Sobolev space of dominating mixed smoothness and we will look at the L^2-error:

$$\|f - A_{s,\ell} f\|_{L^2([0,1]^s)}.$$

Abstract error bounds for Smolyak interpolation error

Lemma 4.2 (Wasilkowski & Woźniakowski 1995)

Let $\mathcal{H}_1$ be a Hilbert space of functions $f : [0,1] \to \mathbb{R}$ and let $\mathcal{Y}_1$ be a Banach space such that $\mathcal{H}_1 \subset \mathcal{Y}_1$. Let $\{A_{1,\ell}\}_{\ell \in \mathbb{N}}$ be a sequence of interpolation rules such that for $\alpha \in (0,1)$ and all $\ell \in \mathbb{N}$

(S1) $e_{1,0}(\mathcal{H}_1, \mathcal{Y}_1, A_{1,0}) := \sup_{f \in \mathcal{H}_1, \|f\|_{\mathcal{H}_1} \leq 1} \|f\|_{\mathcal{Y}_1} \leq C_1,$

(S2) $e_{1,\ell}(\mathcal{H}_1, \mathcal{Y}_1, A_{1,\ell}) \leq C_2 \alpha^\ell,$

(S3) $\|\Delta_\ell\| = \sup_{f \in \mathcal{H}_1, \|f\|_{\mathcal{H}_1} \leq 1} \|\Delta_\ell f\|_{\mathcal{Y}_1} \leq C_3 \alpha^\ell.$

Then, for $s \in \mathbb{N}$, the worst-case error of the Smolyak interpolation (4.3) in the tensor product Hilbert space $\mathcal{H}_s = \otimes_{j=1}^s \mathcal{H}_1$, measured in $\mathcal{Y}_s = \otimes_{j=1}^s \mathcal{Y}_1$, is bounded by

$$e_{s,\ell}(\mathcal{H}_s, \mathcal{Y}_s, A_{s,\ell}) \leq C_2 \max\{C_1, C_3 \alpha\}^{s-1} \binom{\ell + s - 1}{s - 1} \alpha^\ell. \tag{4.8}$$

Proof. The proof is the same as for Lemma 3.2. In fact, the original statement [Lemma 2, Wasilkowski & Woźniakowski 1995] is more general. □

Explicit error bound for sparse grid interpolation

Theorem 4.3

Let $s \in \mathbb{N}$, and let $\{A_{1,\ell}\}_{\ell\in\mathbb{N}}$ be a sequence of 1D interpolation rules such that $n_\ell = \mathcal{O}(2^\ell)$, and their worst-case L^2-errors in $H^r[0,1]$, $r \in \mathbb{N}$, satisfy

$$e_{1,\ell}(H^r[0,1], L^2[0,1], A_{1,\ell}) \;=\; \mathcal{O}(n_\ell^{-r}) \;=\; \mathcal{O}(2^{-\ell r}).$$

Then, for $f \in H^{\boldsymbol{r}}_{\mathrm{mix}}([0,1]^s)$ the L^2-error of the sparse grid interpolation (4.3) of order ℓ is bounded by

$$\begin{aligned} \|f - A_{s,\ell} f\|_{L^2([0,1]^s)} &\le C 2^{-\ell r} \ell^{s-1} \|f\|_{H^{\boldsymbol{r}}_{\mathrm{mix}}} \\ &= \mathcal{O}\big(N_\ell^{-r} \log(N_\ell)^{(r+1)(s-1)}\big), \end{aligned} \tag{4.9}$$

where $C < \infty$ may depend on s.

Proof. The proof follows from Lemma 4.2 with $\alpha = 2^{-r}$, and then that the total number of points/degrees of freedom is $N_\ell = \mathcal{O}(2^\ell \ell^{s-1})$. □

Comments on sparse grid interpolation

- As an example, for $f \in H^2[0,1]$ the L^2 error of piecewise linear interpolation is $\mathcal{O}(h_\ell^2) = \big(n_\ell^{-2}\big)$. Hence, for $f \in H^2_{\mathrm{mix}}([0,1]^s$ the error of a piecewise linear sparse grid interpolant is $\mathcal{O}(N_\ell^{-2} \log(N_\ell)^{3(s-1)})$.
  In general, the L^2 error in $H^r[0,1]$ of 1D piecewise Lagrangian interpolation of order $r-1$ is $\mathcal{O}(h_\ell^r) = \mathcal{O}(n_\ell^{-r})$, and so a sparse grid based on Lagrangian interpolation achieves the error bound (4.9).
- We can also take more general approximation methods for our 1D rules, e.g., interpolation on $\mathbb{R}$ using Hermite polynomials, least-squares approximation, best n term approximation, wavelets ...
- How do we compute the difference approximations $\Delta_\ell = A_{1,\ell} - A_{1,\ell-1}$?

Basis systems for linear interpolation

Consider again piecewise linear interpolation on an equidistant grid $\mathcal{P}_\ell = \{t_{\ell,k}\}_{k=1}^{n_\ell} \subset [0,1]$, with $n_\ell = 2^{\ell-1}+1$, $h_\ell = 2^{-(\ell-1)}$ and $t_{\ell,k} = (k-1)h_\ell$. The grids are nested $\mathcal{P}_\ell \subset \mathcal{P}_{\ell+1}$, and so $V_{1,\ell} \subset V_{1,\ell+1}$ as well, but the basis functions *are not necessarily* nested.

Hat functions

$$\phi_{\ell,k}(y) = \begin{cases} \frac{y - t_{\ell,k-1}}{h_\ell} & y \in (t_{\ell,k-1}, t_{\ell,k}] \\ \frac{t_{\ell,k+1} - y}{h_\ell} & y \in (t_{\ell,k}, t_{\ell,k+1}) \\ 0 & \text{otherwise.} \end{cases}$$

Hierarchical basis

k odd:

$$\phi_{\ell,k} = \phi_{\ell-1,(k+1)/2}$$

k even:

$$\phi_{\ell,k}(y) = \begin{cases} \frac{y - t_{\ell,k-1}}{h_\ell} & y \in (t_{\ell,k-1}, t_{\ell,k}] \\ \frac{t_{\ell,k+1} - y}{h_\ell} & y \in (t_{\ell,k}, t_{\ell,k+1}) \\ 0 & \text{otherwise.} \end{cases}$$

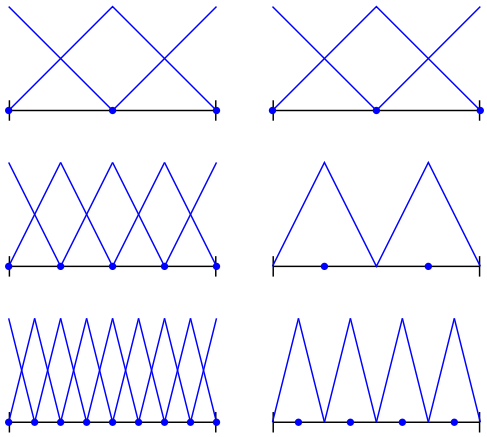


Figure: Basis functions for piecewise linear interpolation: hat functions (L) and hierarchical basis (R).

Hierarchical interpolation

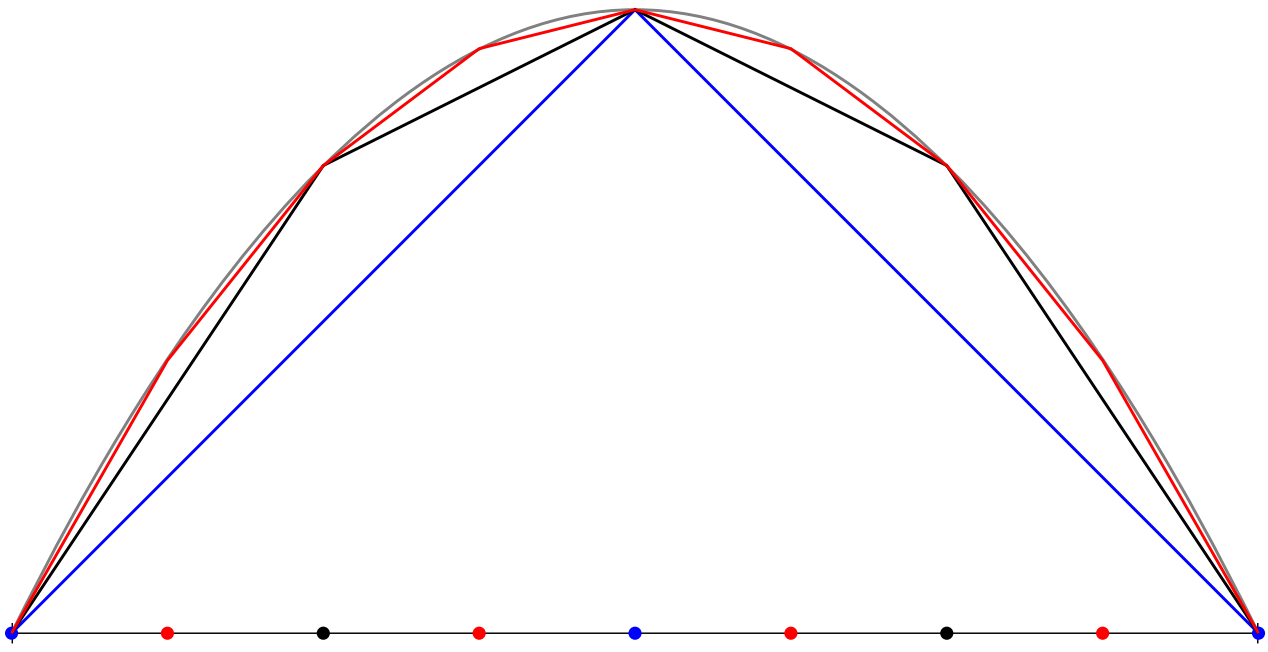


Figure: Hierarchical piecewise linear interpolation of a parabola.

Hierarchical interpolation in one-dimension

The difference spaces $\{W_{1,\ell}\}_{\ell\in\mathbb{N}}$ can be represented by the hierarchical basis as

$$W_{1,\ell} \;=\; V_{1,\ell} \setminus V_{1,\ell-1} \;=\; \operatorname{span}\big\{\phi_{\ell,2i} : i = 1, 2, \ldots, (n_\ell - 1)/2 = m_\ell\big\}.$$

Then the interpolation rule $A_{1,\ell} : C[0,1] \to V_{1,\ell}$ can be constructed *hierarchically*

$$A_{1,\ell} f \;=\; \sum_{k=1}^{\ell} \sum_{i=1}^{m_k} \alpha_{k,i} \phi_{k,2i},$$

where $\alpha_{k,i} \in \mathbb{R}$ is the *coefficient* of the *basis function* $\phi_{k,2i} \in W_{1,k}$.
The coefficients are given by $\alpha_{k,1} = f(0)$, $\alpha_{k,m_k} = f(1)$ and

$$\begin{aligned}
\alpha_{k,i} \;&=\; f(t_{k,2i}) - \frac{1}{2}\big(f(t_{k,2i} - h_k) + f(t_{k,2i} + h_k)\big) \\
&=\; f(t_{k,2i}) - \frac{1}{2}\big(f(t_{k,2i-1}) + f(t_{k,2i+1})\big) \\
&=\; f(t_{k,2i}) - \frac{1}{2}\big(f(t_{k-1,i}) + f(t_{k-1,i+1})\big),
\end{aligned}$$

for $i = 2, 3, \ldots, m_k - 1$.

Tensor product of hierarchical 1D interpolation

Let $s > 1$ and $\boldsymbol{\ell} \in \mathbb{N}^s$, then the *hierarchical tensor product* is given by

$$\left(\bigotimes_{j=1}^{s} A_{1,\ell_j}\right) f \;=\; \sum_{k_1=1}^{\ell_1} \sum_{i_1=1}^{m_{k_1}} \sum_{k_2=1}^{\ell_2} \sum_{i_2=1}^{m_{k_2}} \cdots \sum_{k_s=1}^{\ell_s} \sum_{i_s=1}^{m_{k_s}} \alpha_{\boldsymbol{k},\boldsymbol{i}} \prod_{j=1}^{s} \phi_{k_j,2i_j},$$

where the coefficient $\alpha_{\boldsymbol{k},\boldsymbol{i}} \in \mathbb{R}$ is constructed as follows.
Define the one-dimensional *interpolation coefficient operator (stencil)* $\Xi_{k,i} : C[0,1] \to \mathbb{R}$, which maps $f_1 \mapsto \alpha_{k,i}$, and is given by

$$\Xi_{k,i} f \;=\; -\frac{1}{2} f(t_{k,2i-1}) + f(t_{k,2i}) - \frac{1}{2} f(t_{k,2i+1}) \;=\; \big[-\tfrac{1}{2} \quad 1 \quad -\tfrac{1}{2}\big]_{k,2i}\, f.$$

The coefficient is then given by the product of the 1D interpolation coefficient operators/stencils

$$\alpha_{\boldsymbol{k},\boldsymbol{i}} \;=\; \left(\prod_{j=1}^{s} \Xi_{k_j,i_j}\right) f \;=\; \left(\prod_{j=1}^{s} \big[-\tfrac{1}{2} \quad 1 \quad -\tfrac{1}{2}\big]_{k_j,2i_j}\right) f.$$

General hierarchical tensor product interpolation

Let $\{V_{1,\ell}\}_{\ell\in\mathbb{N}}$ be a hierarchy of finite-dimensional interpolation spaces

$$V_{1,\ell} \;=\; \operatorname{span}\{\phi_{\ell,i} : i = 1, 2, \ldots, n_\ell\} \;=\; \bigoplus_{k=1}^{\ell} W_{1,k}.$$

Let the index set for the "new" basis functions for the difference space $W_{1,k}$ be $\mathcal{M}_k \subset \{1, 2, \ldots, n_\ell\}$, so that $m_k = \dim\left(W_{1,k}\right) = |\mathcal{M}_k|$ and

$$W_{1,k} \;=\; V_{1,k} \setminus V_{1,k-1} \;=\; \operatorname{span}\{\phi_{k,i} : i \in \mathcal{M}_k\}.$$

Let the 1D interpolation operator $A_{s,\ell} : C[0,1] \to V_{1,\ell}$ be given by

$$A_{1,\ell} f \;=\; \sum_{k=1}^{\ell} \sum_{i\in\mathcal{M}_k} \left(\Xi_{k,i} f\right)\phi_{k,i},$$

where $\Xi_{k,i}$ is the interpolation coefficient operator for the basis function $\phi_{k,i}$.
For $\boldsymbol{\ell} \in \mathbb{N}^s$, the hierarchical representation of the tensor product interpolant is

$$\left(\bigotimes_{j=1}^{s} A_{1,\ell_j}\right) f \;=\; \sum_{\boldsymbol{k}\le\boldsymbol{\ell}} \sum_{\boldsymbol{i}\in\mathcal{M}_{\boldsymbol{k}}} \alpha_{\boldsymbol{k},\boldsymbol{i}} \prod_{j=1}^{s} \phi_{k_j,i_j}.$$

where $\boldsymbol{k}, \boldsymbol{i} \in \mathbb{N}^s$, $\mathcal{M}_{\boldsymbol{k}} = \otimes_{j=1}^{s} \mathcal{M}_{k_j} \subset \mathbb{N}^s$ and $\alpha_{\boldsymbol{k},\boldsymbol{i}} = \left(\prod_{j=1}^{s} \Xi_{k_j,i_j}\right) f$.

Hierarchical representation of sparse grid interpolation

The sparse grid interpolant (4.3) can also be written in hierarchical form

$$A_{s,\ell} f \;=\; \sum_{|\boldsymbol{k}|\le \ell+s-1} \sum_{\boldsymbol{i}\in\mathcal{M}_{\boldsymbol{k}}} \alpha_{\boldsymbol{k},\boldsymbol{i}} \prod_{j=1}^{s} \phi_{k_j,i_j} \tag{4.10}$$

where $\alpha_{\boldsymbol{k},\boldsymbol{i}} \;=\; \left(\prod_{j=1}^{s} \Xi_{k_j,i_j}\right) f$ (and we assume the same setting as before).

Example

The sparse grid interpolant based on piecewise linear interpolation is given explicitly by

$$A_{s,\ell} f \;=\; \sum_{|\boldsymbol{k}|\le \ell+s-1} \sum_{i_1=1}^{m_{k_1}} \sum_{i_2=1}^{m_{k_2}} \cdots \sum_{i_s=1}^{m_{k_s}} \alpha_{\boldsymbol{k},\boldsymbol{i}} \prod_{j=1}^{s} \phi_{k_j,2i_j}$$

where

$$\alpha_{\boldsymbol{k},\boldsymbol{i}} \;=\; \left(\prod_{j=1}^{s} \begin{bmatrix} -\frac{1}{2} & 1 & -\frac{1}{2} \end{bmatrix}_{k_j,2i_j}\right) f.$$

Comments on hierarchical representations

- In practice, one can evaluate and store the function evaluations at all of the grid points, then applies the product stencils to obtain the coefficients $\alpha_{\boldsymbol{k},\boldsymbol{i}}$.
- Approximations based on higher order interpolants or different polynomial basis systems can be hierarchically constructed in the same way as for the piecewise linear case. One just needs to identify the indices of the new basis functions and construct the stencils to obtain the coefficients in 1D.

Summary

- The sparse grid formulation can also be used for interpolation in high dimensions.
- The error of sparse grid interpolation in higher dimensions is of the same order as the error in one dimension and the full tensor product, but the sparse grid formulation uses drastically less degrees of freedom than the full tensor product.
- In practice, to compute sparse grid interpolants efficiently it is useful to work with hierarchical basis systems for the 1D interpolation spaces.

5. Sparse grid stochastic collocation

Stochastic PDE problem again

Let $D \subset \mathbb{R}^d$, for $d = 1, 2, 3$, be a bounded convex domain, and consider the again the stochastic PDE

$$\begin{aligned} -\nabla \cdot (a(\boldsymbol{x}, \boldsymbol{y}) \nabla u(\boldsymbol{x}, \boldsymbol{y})) &= f(\boldsymbol{x}), & \boldsymbol{x} &\in D, \\ u(\boldsymbol{x}, \boldsymbol{y}) &= 0, & \boldsymbol{x} &\in \partial D, \end{aligned} \tag{5.1}$$

where $\boldsymbol{x} \in D$ is the *physical variable* and $\boldsymbol{y} \sim \mathrm{Uni}([-\frac{1}{2}, \frac{1}{2}]^s)$ is a *random parameter* (i.e., $\Omega = [-\frac{1}{2}, \frac{1}{2}]$). Assume $f \in L^2(D)$ and that the coefficient,

$$a(\boldsymbol{x}, \boldsymbol{y}) = a_0(\boldsymbol{x}) + \sum_{j=1}^{s} y_j a_j(\boldsymbol{x}), \tag{5.2}$$

satisfies Assumption III.1, so that $u \in L^\infty\big(\Omega^s; H_0^1(D)\big)$.

Goal: Approximate

1. $\mathcal{G}(u(\boldsymbol{y}))$ on Ω^s for a linear functional $\mathcal{G} \in H^{-1}(D)$, or
2. the *full* solution u on $D \times \Omega^s$.

Approximation strategy

Finite element discretisation. Let $V_h^D \subset H_0^1(D)$ be a FE subspace corresponding to a triangulation $\mathscr{T}_h$ of D with mesh width $h > 0$ (Appendix B).
For each $\boldsymbol{y}$ the FE approximation is $u(\boldsymbol{y}) \approx u_h(\boldsymbol{y}) \in V_h^D$.

Interpolation on the stochastic domain Ω^s. Let $\{V_{s,\ell}^\Omega \subset L^2(\Omega^s\}_{\ell\in\mathbb{N}}$ be a sequence of interpolation spaces on $\Omega^s = [-\frac{1}{2},\frac{1}{2}]^s$ and let $A_{s,\ell} : C(\Omega^s) \to V_{s,\ell}^\Omega$ be the interpolation rules, e.g., a tensor product interpolant (4.2) or a sparse grid interpolant (4.3).
We can then interpolate $u(\boldsymbol{y}) \in H_0^1(D)$ on Ω^s by applying $A_{s,\ell}$:

$$\mathcal{G}(u(\boldsymbol{y})) \approx A_{s,\ell}\mathcal{G}(u(\boldsymbol{y})) \quad \text{and} \quad u(\boldsymbol{y}) \approx A_{s,\ell}u(\boldsymbol{y}).$$

Combined approximations

$$\mathcal{G}(u(\boldsymbol{y})) \approx A_{s,\ell}\mathcal{G}(u_h(\boldsymbol{y})) \quad \text{and} \quad u(\boldsymbol{x},\boldsymbol{y}) \approx u_{h,\ell}(\boldsymbol{x},\boldsymbol{y}) := A_{s,\ell}u_h(\boldsymbol{x},\boldsymbol{y}).$$

Key points:

1. $\mathcal{G}(u) \in C(\Omega^s)$ is a function of $\boldsymbol{y}$ only, and so we can directly apply the interpolation methods and results from Section 4. Now, "function evaluations" are PDE solves.
2. $u \in L^\infty(\Omega^s; H_0^1(D))$ and so we are applying an interpolation rule to $u(\boldsymbol{y})$ which takes values in $H_0^1(D)$ (or V_h^D). This is stochastic collocation.

Direct application of sparse grids to $\mathcal{G}(u(\boldsymbol{y}))$

For $\mathcal{G} \in H^{-1}(D)$, the quantity of interest $\mathcal{G}(u) \in C(\Omega^s)$ is a function of $\boldsymbol{y}$ only.
So we can immediately approximate $\mathcal{G}(u(\boldsymbol{y}))$ using the methods from Section 4:

$$\mathcal{G}(u) \approx A_{s,\ell}\mathcal{G}(u).$$

E.g., a sparse piecewise linear interpolant is given explicitly by

$$A_{s,\ell}\mathcal{G}(u) = \sum_{|\boldsymbol{k}|\le\ell+s-1} \sum_{i_1=1}^{m_{k_1}} \sum_{i_2=1}^{m_{k_2}} \cdots \sum_{i_s=1}^{m_{k_s}} \left(\prod_{j=1}^{s} \left[-\tfrac{1}{2} \quad 1 \quad -\tfrac{1}{2}\right]_{k_j,2i_j}\right) \mathcal{G}(u) \left(\prod_{j=1}^{s} \varphi_{k_j,2i_j}\right)$$

where $\varphi_{k,i}(y) = \phi_{k,i}(y+\frac{1}{2})$ and the 1D gridpoints are $\tau_{k,i} = (i-1)2^{-(k-1)} - \frac{1}{2}$.
Similarly, we can approximate the expectation by a sparse grid quadrature rule

$$\mathbb{E}[\mathcal{G}(u)] = \int_{[-\frac{1}{2},\frac{1}{2}]^s} \mathcal{G}(u(\boldsymbol{y}))\,\mathrm{d}\boldsymbol{y} \approx Q_{s,\ell}\mathcal{G}(u).$$

Total error bound

$$\|\mathcal{G}(u) - A_{s,\ell}\mathcal{G}(u_h)\|_{L^2(\Omega^s)} \le \underbrace{\|\mathcal{G}(u) - \mathcal{G}(u_h)\|_{L^2(\Omega^s)}}_{\text{FE error}} + \underbrace{\|\mathcal{G}(u_h) - A_{s,\ell}\mathcal{G}(u_h)\|_{L^2(\Omega^s)}}_{\text{SG error}} \tag{5.3}$$

Analytic regularity in the stochastic domain

Theorem 5.1

For $\boldsymbol{\nu} \in \mathbb{N}^s$, the order $\boldsymbol{\nu}$ derivative with respect to $\boldsymbol{y}$ is bounded by

$$\left\| \frac{\partial^{|\boldsymbol{\nu}|}}{\partial \boldsymbol{y}^{\boldsymbol{\nu}}} u(\boldsymbol{y}) \right\|_{L^\infty(\Omega^s; H^1(D))} \leq \frac{\|f\|_{L^2(D)}}{a_{\min}} |\boldsymbol{\nu}|! \prod_{j=1}^{s} \left(\frac{\|a_j\|_{L^\infty(D)}}{a_{\min}} \right)^{\nu_j}.$$

and hence, $\mathcal{G}(u) \in H^{\boldsymbol{r}}_{\mathrm{mix}}(\Omega^s)$ for all $r \in \mathbb{N}$.
Furthermore, for all $\boldsymbol{y}_{-j} := (y_i : i \in \{1:s\} \setminus \{j\}) \in \Omega^{s-1}$ the one-parameter function $u(\cdot; \boldsymbol{y}_{-j}) \in C(\Omega; H^1_0(D))$ admits a unique analytic extension $u(\xi, \boldsymbol{y}_{-j})$ for all $\xi \in \Sigma(\Omega; \rho_j) \subset \mathbb{C}$, and the domain of analyticity is

$$\Sigma(\Omega; \rho_j) := \left\{ \xi \in \mathbb{C} : \mathrm{dist}(\xi, \Omega) \leq \rho_j \right\} \quad \text{where } \rho_j = \frac{a_{\min}}{4\|a_j\|_{L^\infty(D)}}.$$

Since $V_h^D \subset H^1_0(D)$, the same results also hold for $u_h(\boldsymbol{y}) \in V_h^D$.

Proof. The derivative bounds are proved as in Lemma III.6.1, and the analytic extension then follows by the derivative bounds (see [Lemma 3.2 Babuška, Nobile & Tempone 2007]). □

Total error for sparse grid interpolation linear functional

Theorem 5.2

Let $V_h^D \subset H^1_0(D)$ be the piecewise linear FE space for a triangulation of D with meshwidth $h > 0$, and let $\{V^\Omega_{1,\ell}\}_{\ell \in \mathbb{N}}$ be a sequence of 1D interpolation spaces on Ω with $n_\ell = \dim\left(V^\Omega_{1,\ell}\right) = \mathcal{O}(2^\ell)$. Also, let the worst-case L^2 error in $H^r(\Omega)$ of the corresponding 1D interpolation rules $A_{1,\ell} : C(\Omega) \to V^\Omega_{1,\ell}$ satisfy

$$e_{1,\ell}(H^r(\Omega), L^2(\Omega), A_{1,\ell}) = \mathcal{O}(n_\ell^{-r}) = \mathcal{O}(2^{-\ell r}).$$

Then, for $\mathcal{G} \in L^2(D)$ the total error the combined FE sparse grid interpolation rule applied to $\mathcal{G}(u)$ is bounded by

$$\|\mathcal{G}(u) - A_{s,\ell}\mathcal{G}(u_h)\|_{L^2(\Omega^s)} \leq C\left(h^2 + N_\ell^{-r} \log(N_\ell)^{(r+1)(s-1)}\right)$$

where $C < \infty$ may depend on s, but is independent of h, ℓ and N_ℓ.

Proof. By the triangle inequality

$$\|\mathcal{G}(u) - A_{s,\ell}\mathcal{G}(u_h)\|_{L^2(\Omega^s)} \leq \|\mathcal{G}(u) - \mathcal{G}(u_h)\|_{L^2(\Omega^s)} + \|\mathcal{G}(u_h) - A_{s,\ell}\mathcal{G}(u_h)\|_{L^2(\Omega^s)}.$$

The 1st term (FE) can be bounded by (II.7.5). By Theorem IV.5.1 $\mathcal{G}(u_h) \in H^{\boldsymbol{r}}_{\mathrm{mix}}(\Omega^s)$ so the 2nd term (SG) can be bounded by Theorem IV.4.3. □

Stochastic collocation: Approximating the full solution

Collocation is a method of solving differential or integral equations by solving the equation at a collection of *collocation points* in the domain and then reconstructing the full solution.

Stochastic collocation is a method for approximating the full parametric solution of a stochastic PDE (e.g., (5.1)) by collocation in the parameter domain, i.e., applying interpolation on function space.
As an example, for an interpolation space $V_{s,\ell}^{\Omega} = \operatorname{span}\{\phi_k\}$ and grid $\{\boldsymbol{t}_k\}$ on Ω^s

$$u(\boldsymbol{x},\boldsymbol{y}) \approx \sum_k u(\boldsymbol{x},\boldsymbol{t}_k)\phi_k(\boldsymbol{y}),$$

e.g., tensor product or sparse grid polynomial interpolation.
Basic idea: Apply a sparse grid interpolation rule to efficiently handle the high-dimensional parameter domain.
The collocation points are the sparse grid points, and the coefficients in the interpolation are now functions in $H_0^1(D)$ (or V_h^D).

Note: In practice, we must also approximate the u in the spatial domain by a FE solution u_h. But as we saw for $\mathcal{G}(u)$, we can handle the FE error by the triangle inequality, and $V_h^D \subset H_0^1(D)$ so the stochastic regularity results also hold for u_h.

Stochastic collocation for one parameter

Let $s = 1$, so that for $y \in \Omega$ the stochastic PDE is

$$-\nabla \cdot \big((a_0(\boldsymbol{x}) + y a_1(\boldsymbol{x}))\nabla u(\boldsymbol{x},y)\big) = f(\boldsymbol{x}), \qquad \text{for } \boldsymbol{x} \in D.$$

The stochastic collocation based on the interpolation rule $A_{1,\ell} : C(\Omega) \to V_{1,\ell}^{\Omega}$ is

$$u_\ell(\boldsymbol{x},y) = A_{1,\ell}u(\boldsymbol{x},y) = \sum_{k=1}^{n_\ell} u(\boldsymbol{x},t_{\ell,k})\phi_{\ell,k}(y),$$

where $\{t_{\ell,k}\}_{k=1}^{n_\ell}$ are the collocation points and $\{\phi_{\ell,k} \in V_{1,\ell}^{\Omega}\}_{k=1}^{n_\ell}$ is the basis.

E.g., the FE and stochastic collocation (using hierarchical piecewise linear interpolation) approximation is

$$u_{h,\ell}(\boldsymbol{x},\boldsymbol{y}) = \sum_{k=1}^{\ell}\sum_{i=1}^{m_k}\big[-\tfrac{1}{2}u_h(\boldsymbol{x},t_{k,2i-1}) + u_h(\boldsymbol{x},t_{k,2i}) - \tfrac{1}{2}u_h(\boldsymbol{x},t_{k,2i+1})\big]\phi_{k,2i}(y),$$

where the collocation points are $\{t_{k,i} = (i-1)2^{-(k-1)} - \frac{1}{2}\}$ and $\{\phi_{k,i}\}$ are the corresponding hierarchical basis functions.

Sparse grid stochastic collocation

Let $A_{s,\ell} : C(\Omega^s) \to V^\Omega_{s,\ell}$ be a sparse grid interpolation rule as in (4.10).
The sparse grid stochastic collocation approximation $u_\ell \in H^1_0(D) \otimes V^\Omega_{s,\ell}$ is then

$$u_\ell(\boldsymbol{x},\boldsymbol{y}) \;=\; A_{s,\ell}u(\boldsymbol{x},\boldsymbol{y}) \;=\; \sum_{|\boldsymbol{k}|\le \ell+s-1} \sum_{\boldsymbol{i}\in\mathcal{M}_{\boldsymbol{k}}} \alpha_{\boldsymbol{k},\boldsymbol{i}}(\boldsymbol{x}) \prod_{j=1}^{s} \phi_{k_j,i_j}(y_j), \qquad (5.4)$$

where now $\alpha_{\boldsymbol{k},\boldsymbol{i}} \in H^1_0(D)$ is given by

$$\alpha_{\boldsymbol{k},\boldsymbol{i}}(\boldsymbol{x}) \;=\; \left(\prod_{j=1}^{s} \Xi_{k_j,i_j}\right) u(\boldsymbol{x},\cdot).$$

E.g., for piecewise linear interpolation the FE and sparse grid stochastic collocation approximation is given explicitly by

$$u_{h,\ell}(\boldsymbol{x},\boldsymbol{y}) = \sum_{|\boldsymbol{k}|\le \ell+s-1} \sum_{i_1=1}^{m_{k_1}} \cdots \sum_{i_s=1}^{m_{k_s}} \left(\prod_{j=1}^{s} \left[-\tfrac{1}{2} \quad 1 \quad -\tfrac{1}{2}\right]_{k_j,2i_j}\right) u_h(\boldsymbol{x},\cdot) \prod_{j=1}^{s} \phi_{k_j,2i_j}(y_j).$$

Sparse grid stochastic collocation error

Theorem 5.3 (Nobile, Tempone & Webster 2007)

Let $\{A_{1,\ell}\}_{\ell\in\mathbb{N}}$ be a sequence of 1D interpolation rules based on Gaussian abscissas with $n_1 = 1$ and $n_\ell = 2^{\ell-1}+1$. Then the error of the sparse grid stochastic collocation (5.4) *in the Bochner space $L^2(\Omega^s, H^1_0(D))$ satisfies*

$$\|u - A_{s,\ell}u\|_{L^2(\Omega^s,H^1_0(D))} \;\le\; C_1 N_\ell^{-\eta_1}$$

where, for $\rho_{\min} = \min_{j=1}^s \rho_j = \min_{j=1}^s a_{\min}/(4\|a_j\|_{L^\infty(D)})$,

$$\eta_1 \;=\; \frac{e\log(2)}{1+(1+\log_2(3/2))\log(2)+\log(s)}\rho_{\min}.$$

Furthermore, if $\ell > s/\log(2)$ then the error satisfies the subexponential bound

$$\|u - A_{s,\ell}u\|_{L^2(\Omega^s,H^1_0(D))} \;\le\; C_2 \exp\left(-\frac{s\rho_{\min}}{2^{1/s}} N_\ell^{\eta_2}\right),$$

where

$$\eta_2 \;=\; \frac{\log(2)}{s[1+(1+\log_2(3/2))\log(2)+\log(s)]} \;<\; 1.$$

Here $C_1, C_2 < \infty$ may depend on s but are independent of N_ℓ.

Summary

- Sparse grid quadrature and interpolation can be used to tackle difficult high dimensional stochastic PDE problems.
- For a quantity of interest $\mathcal{G}(u)$, we can directly apply sparse grid interpolation/quadrature, and the error analysis follows from our previous results in Sections 3 and 4.
- Stochastic collocation approximates the full parametric solution by interpolation in function space on the stochastic domain, and can easily be combined with FE methods for the spatial domain.
  Essentially, the only difference is that the interpolation coefficients are now functions.
- The solution u is analytic in the stochastic parameter, and the "smoothness" can be measured by the size of the analytic extension into the complex plane.
- The error analysis of sparse grid stochastic collocation relies on this analytic regularity. The asymptotic convergence rate is subexponential, and the preasymptotic rate is algebraic. Both convergence rates depend on the radius of the analytic extension.
- $\log$-normal problem can also be handled by using interpolation on $\mathbb{R}$, e.g., using Hermite polynomials.

6. Adaptive sparse grids

General sparse grids

$$\int_{[0,1]^s} f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \approx Q_{s,\ell} f = \sum_{|\boldsymbol{k}|\le \ell+s-1} \left(\bigotimes_{j=1}^{s} \Delta_{k_j} \right) f$$

The sparse grid quadrature rule is *isotropic*, i.e., each dimension j is treated equally, and as such the sparse grid errors still depend on the dimension s.

The key to the performance of sparse grids in high dimensions is the structure of the *index set*

$$\{\boldsymbol{k} \in \mathbb{N}^s : |\boldsymbol{k}| \le \ell + s - 1\},$$

but is this suitable for *all* problems?
Let $\mathscr{I} \subset \mathbb{N}^s$, then a *general* or *anisotropic* sparse grid quadrature rule is

$$Q_{s,\mathscr{I}} f = \sum_{\boldsymbol{k}\in\mathscr{I}} \left(\bigotimes_{j=1}^{s} \Delta_{k_j} \right) f. \tag{6.1}$$

Examples

1. isotropic sparse grid: $\mathscr{I} = \{\boldsymbol{k} \in \mathbb{N}^s : |\boldsymbol{k}| \le \ell + s - 1\}$,
2. full tensor product: $\mathscr{I} = \{\boldsymbol{k} \in \mathbb{N}^s : |\boldsymbol{k}|_\infty \le \ell\}$,
3. *weighted* sparse grid: $\mathscr{I} = \{\boldsymbol{k} \in \mathbb{N}^s : \boldsymbol{\gamma} \cdot \boldsymbol{k} \le \ell + s - 1\}$ where $\boldsymbol{\gamma} \in \mathbb{R}^s_+$,
4. hyperbolic cross: $\mathscr{I} = \{\boldsymbol{k} \in \mathbb{N}^s : \prod_{j=1}^{s} \max\{k_j, 1\} \le \ell\}$.

Admissible index sets

Definition 6.1

An index set $\mathscr{I} \subset \mathbb{N}^s$ is called *downward closed* or *admissible* if for all $\boldsymbol{k} \in \mathscr{I}$

$$\boldsymbol{k} - \boldsymbol{e}_j \in \mathscr{I}, \quad \text{for } j = 1, 2, \ldots, s \quad \text{such that } k_j > 1,$$

where $\boldsymbol{e}_j \in \{0,1\}^s$ is the jth unit vector.

Key idea: Index sets that are downward closed preserve the telescoping property of the differences in the Smolyak operator.

Examples

For $s \in \mathbb{N}$, $\ell \in \mathbb{N}$ and $\boldsymbol{\gamma} \in \mathbb{N}^s$:

1. $\{\boldsymbol{k} \in \mathbb{N}^s : |\boldsymbol{k}| \le \ell + s - 1\}$ is admissible,
2. $\{\boldsymbol{k} \in \mathbb{N}^s : |\boldsymbol{k}|_\infty \le \ell\}$ is admissible,
3. $\{\boldsymbol{k} \in \mathbb{N}^s : \boldsymbol{\gamma} \cdot \boldsymbol{k} \le \ell + s - 1\}$ is admissible,
4. $\{\boldsymbol{k} \in \mathbb{N}^s : |\boldsymbol{k}| = \ell + s - 1\}$ is not admissible.

Properties of general sparse grids

- Anisotropic sparse grids allow to handle each dimension differently, i.e., for the more important dimensions we can use a higher precision rule with more points and less points for the less important dimensions.
- General sparse grids are difficult to analyse, and so the convergence results are limited.
- Constructing suitable index sets for given problems can be difficult— we need to know *a priori* the important dimensions, and then how this information translates into the choice of index set.
- After computing a given general sparse grid approximation, how do we extend the index set to best increase the accuracy?

To overcome the last three difficulties we will choose the index set *automatically*.

Adaptive sparse grids

Key idea: Given f, choose the index set adaptively *while* computing the sparse grid approximation.

Goal: Compute an index set by sequentially adding index vectors such that

- the new index set is still admissible,
- the error is significantly reduced without a drastic increase in cost, and
- only information given by the previous computations and the cost of differential quadrature approximations are used.

Basic structure of the adaptive algorithm:

1. Start with $\mathscr{I} = \{(1, 1, \ldots, 1)\}$.
2. Compute the set of *new admissible indices* to be considered $\{\boldsymbol{k} + \boldsymbol{e}_j, \boldsymbol{k} \in \mathscr{I}$ and $j = 1, 2, \ldots, s\}$.
3. Select the new admissible index that gives the "best" error reduction, and add it to $\mathscr{I}$.
4. Update the quadrature approximation.
5. Repeat steps 2–4 until the total error estimate is below a given tolerance, or the maximum work is exceeded.

Main ingredients for adaptive algorithm

- Error tolerance $\varepsilon > 0$,
- *differential* quadrature rule $\Delta_{\boldsymbol{k}} := \bigotimes_{j=1}^{s} \Delta_{k_j}$ for $\boldsymbol{k} \in \mathbb{N}^s$,
- *forward neighbourhood* of $\boldsymbol{k}$: $\{\boldsymbol{k} + \boldsymbol{e}_j : j = 1, 2, \ldots, s\}$,
- *backward neighbourhood* of $\boldsymbol{k}$: $\{\boldsymbol{k} - \boldsymbol{e}_j : j = 1, 2, \ldots, s \text{ and } k_j > 1\}$,
- $\mathscr{A}$ the *active* index set, which holds the index vectors whose forward neighbourhoods are currently being considered for inclusion,
- $\mathscr{O}$ the *old* index set, which the holds the index vectors $\boldsymbol{k}$ that have already been considered,
- an index $\boldsymbol{i}$ is called *admissible* if the backward neighbourhood of $\boldsymbol{i}$ is included in the old index set, i.e., $\boldsymbol{i} - \boldsymbol{e}_j \in \mathscr{O}$ for all $j = 1, 2, \ldots, s$ with $i_j > 1$,
- a *local error estimator* $\eta_{\boldsymbol{k}}$, which uses the computation $\Delta_{\boldsymbol{k}} f$ and information about the work,
- the global error estimate $\eta = \sum_{\boldsymbol{k} \in \mathscr{A}} \eta_{\boldsymbol{k}}$.

Dimension-adaptive sparse grid quadrature

Algorithm 2 [Gerstner & Griebel 2003]

Given $f \in C([0,1]^s)$ and an error tolerance $\varepsilon > 0$:

1: Initialise: $\boldsymbol{k} = (1, 1, \ldots, 1)$, $\mathscr{O} = \emptyset$, $\mathscr{A} \leftarrow \{\boldsymbol{k}\}$, $Q \leftarrow \Delta_{\boldsymbol{k}} f$, and $\eta \leftarrow \eta_{\boldsymbol{k}}$.
2: **while** $\eta > \varepsilon$ **do**
3: select $\boldsymbol{k} \in \mathscr{A}$ with largest $\eta_{\boldsymbol{k}}$
4: $\mathscr{A} \leftarrow \mathscr{A} \setminus \{\boldsymbol{k}\}$ ▷ Remove $\boldsymbol{k}$ from active indices
5: $\mathscr{O} \leftarrow \mathscr{O} \cup \{\boldsymbol{k}\}$ ▷ then add $\boldsymbol{k}$ to old indices
6: $\eta \leftarrow \eta - \eta_{\boldsymbol{k}}$ ▷ and update total error estimate.
7: **for** $j = 1, 2, \ldots, s$ **do** ▷ Add new admissible indices
8: $\boldsymbol{i} \leftarrow \boldsymbol{k} + \boldsymbol{e}_j$ ▷ from forward neighbourhood of $\boldsymbol{k}$.
9: **if** $\boldsymbol{i}$ is admissible **then**
10: $\mathscr{A} \leftarrow \mathscr{A} \cup \{\boldsymbol{i}\}$ ▷ Add index to active indices then update
11: Compute $\Delta_{\boldsymbol{i}} f$ and $\eta_{\boldsymbol{i}}$
12: $Q \leftarrow Q + \Delta_{\boldsymbol{i}} f$
13: $\eta \leftarrow \eta + \eta_{\boldsymbol{i}}$
14: **end if**
15: **end for**
16: **end while**

Comments on dimension-adaptive algorithm

- The final index set is $\mathscr{I} = \mathscr{O} \cup \mathscr{A}$, since whenever we add an index $\boldsymbol{i}$ to $\mathscr{A}$ we compute $\Delta_{\boldsymbol{i}} f$ in order to calculate the error estimate $\eta_{\boldsymbol{i}}$. Once we have computed the differential quadrature it should immediately be included in the total approximation.
- For each index $\boldsymbol{k} \in \mathscr{I}$, we must compute the full differential quadrature rule $\bigotimes_{j=1}^{s} \Delta_{k_j}$ instead of just working on the difference grid as in (2.2). This must be done to ensure the weights are correct throughout the algorithm.
- The basic structure of the algorithm can be used for more general problems, e.g., quadrature on $\mathbb{R}^s$, interpolation or stochastic collocation.
- Obviously the total and local errors cannot be computed exactly since the integral is unknown. Hence, we must use the computations that have previously been performed to estimate the error.

Illustration of the evolution of the adaptive algorithm

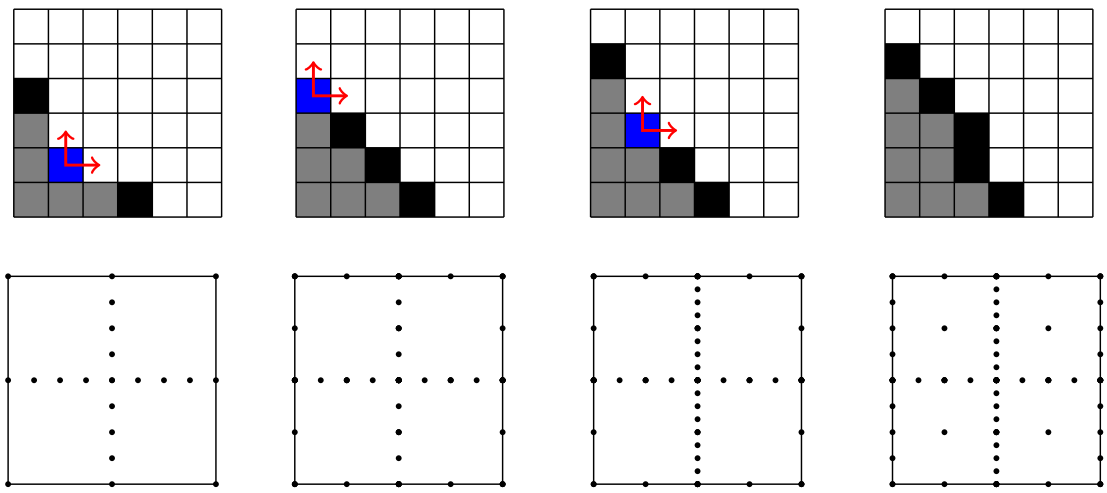


Figure: Three steps of the adaptive sparse grid algorithm in two dimensions. Above depicts the index sets. Each small square corresponds to an index $\boldsymbol{k}$: grey denotes the old index set, black and blue denote the active index set, blue denotes the current index (i.e., $\boldsymbol{k} \in \mathscr{A}$ with the greatest $\eta_{\boldsymbol{k}}$), white is not in the index set, and red arrows denote the forward neighbourhood of $\boldsymbol{k}$. Below are the corresponding sparse grids based on the trapezoidal rule. Source [Gerstner & Griebel 2003].

Error estimation

Goal: to efficiently estimate the error reduction for a given index $\boldsymbol{k}$, *and* take into account the increase in the cost of the approximation.

Error estimate.

Assuming that f is smooth enough such the differential quadrature approximations are roughly decreasing as the order of the index increases, then as an estimate of the local error reduction we can take the differential quadrature approximation $|\Delta_{\boldsymbol{k}} f|$ directly.
The intuition here is that $\Delta_{\boldsymbol{k}} f$ can be thought of as the difference between and more accurate approximation with a less accurate approximation, and if f is smooth enough this gives a good estimate of the error reduction.

Cost estimate

As an estimate of the cost we simply take the total number of function evaluations corresponding to the differential quadrature rule.
Let the total number of points in the differential quadrature rule $\Delta_{\boldsymbol{k}}$ be

$$N_{\boldsymbol{k}} \coloneqq \begin{cases} \displaystyle\prod_{j=1}^{s} n_{k_j}, & \text{if nested,} \\ \displaystyle\prod_{j=1}^{s} \left(n_{k_j} + n_{k_j - 1}\right) & \text{if nonnested.} \end{cases}$$

Error estimation

For $\xi \in [0, 1]$, define the *local error estimator* for an index $\boldsymbol{k} \in \mathbb{N}^s$ by

$$\eta_{\boldsymbol{k}} \coloneqq \max\left\{\xi \frac{|\Delta_{\boldsymbol{k}} f|}{|\Delta_{\boldsymbol{1}} f|}, (1-\xi)\frac{N_{\boldsymbol{1}}}{N_{\boldsymbol{k}}}\right\} \tag{6.2}$$

- In Algorithm 2 we choose the index with the greatest $\eta_{\boldsymbol{k}}$, which here relates to choosing either the greatest error reduction or the cheapest approximation.
- If $\Delta_{\boldsymbol{1}} f = 0$, then we simply take another reference value or approximation.
- $\xi = 1$ gives a greedy algorithm that ignores the cost and chooses the index that gives the greatest error reduction.
- $\xi = 0$ ignores the error and chooses the most efficient approximation. This case corresponds to the usual isotropic sparse grids (1.4).
- This estimate works well if we assume that f is smooth enough such that the differential quadrature rules are decreasing as $\boldsymbol{k}$ "increases".
  Note that there are cases where this may not be true, which can "trick" the adaptive algorithm, but such cases are also not smooth enough to be handled well by other methods, e.g., QMC.
- Numerical results [Gerstner & Griebel 2003] suggest that for this choice of local error estimators the total error η underestimates the true error. But that the underestimation is by a constant factor, i.e., true error $= C \cdot \eta$ for $C > 1$.

Summary

- Anisotropic sparse grids, which are based on general index sets, allow more flexibility to account for structure of the integrand. However, theoretically and practically they can be hard to work with, requiring *a priori* knowledge of the integrand.
- Adaptive sparse grids compute a general index set at the same time as computing the sparse grid approximation.
- One of the key components is a reliable local error estimator, which balances the error reduction and the added cost, and is based on the size of a differential quadrature approximation and the cost to compute that approximation.

7. Extensions

- Sparse grid quadrature/interpolation on $\mathbb{R}^s$ can be performed using Hermite polynomials.
- More general 1D approximation rules can be used within the sparse grid framework, e.g., least-squares approximation, best n-term approximation, trigonometric polynomials...
- Sparse grids may also be used to efficiently approximate the solution to high-dimensional PDEs, by setting up a sparse mesh corresponding to the sparse grid and then computing a Galerkin approximation on the sparse approximation space $V_{s,\ell}$ (as defined in Definition 4.1), see [Bungartz & Griebel 2004].
- Adaptive sparse grids can also be used for stochastic collocation, in which case it is natural to combine them with adaptive FEM, see [Lang, **RS** & Sylvester 2019] and others.
- Sparse grids can be made dimension independent.
  In particular, they also be applied to infinite-dimensional problems (i.e, $s \to \infty$), by using them within a multivariate decomposition algorithm [Kuo, Nuyens, Plaskota, Sloan & Wasilkowski 2017; **AG**, Kuo, Nuyens & Wasilkowski 2018] or by appropriately choosing a general index set [Zech & Schwab 2020], and others.

References

Main references

[1] H-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, **13**, 147–169, 2004.

[2] J. Garcke. Sparse grids in a nutshell. *In: J. Garcke, M. Griebel (eds) Sparse Grids and Applications. Lecture Notes in Computational Science and Engineering* **88**, 55–80, Springer, Berlin-Heidelberg, 2012.
`https://ins.uni-bonn.de/media/public/publication-media/sparse_grids_nutshell.pdf?pk=639`

[3] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numer. Algorithms*, **18**, 209–232, 1998.

Further reading

[4] I. Babuška, F. Nobile and R. Tempone. A stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*., **45**, 1005–1034, 2007.

[5] A. Cohen, R. DeVore and Ch. Schwab. Convergence rates of best N-term approximations for a class of elliptic sPDEs. *Numer. Math.*, **10**, 615–646, 2010.

[6] T. Gerstner and M. Griebel. Dimension-adaptive tensor-product quadrature. Computing, **71**, 65–87, 2003.

[7] **A. D. Gilbert**, F. Y. Kuo, D. Nuyens and G. W. Wasilkowski. Efficient implementations of the Multivariate Decomposition Method for approximating infinite-variate integrals. *SIAM J. Sci. Comp.*, **40**, A3240–A3266, 2018.

[8] D. E. Knuth. *The Art of Computer Programming*, Vol. 4A, Addison–Wesley, 2005.

[9] F. Y. Kuo, D. Nuyens, L. Plaskota, I. H. Sloan and G. W. Wasilkowski. Infinite-dimensional integration and the multivariate decomposition method. *J. Comput. Appl. Math.*, **326**, 217–234, 2017.

References (cont.)

[10] J. Lang, **R. Scheichl** and D. Sylvester. A fully adaptive multilevel stochastic collocation trategy for solving elliptic PDEs with random coefficients. *arXiv:* `https://arxiv.org/abs/1902.03409`, 2019.

[11] F. Nobile, R. Tempone and C. G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, **46**, 2309–2345, 2008.

[12] S. A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl/ Akad. Nauk SSSR 4*, 240–243, 1963.

[13] G. W. Wasilkowski and H. Woźniakowski. Explicit cost bounds of algorithms for multivariate tensor product problems. *J. Complexity*, **11**, 1–56, 1995.

[14] G. von Winckel. Legendre-Gauss quadrature weights and nodes. *MATLAB Central File Exchange*, `https://www.mathworks.com/matlabcentral/fileexchange/4540-legendre-gauss-quadrature-weights-and-nodes`, retrieved June 6, 2020.

[15] J. Zech and Ch. Schwab. Convergence rates of high dimensional Smolyak approximation. *ESAIM—Math. Model. Num.*, **54**, 1259–1307, 2020.

V. Stochastic collocation for $s = \infty$

Jakob Zech

July 6, 2020

1 Sparse-grid interpolation in infinite dimensions

We now generalize our previous notions of multiindices and interpolation operators to the infinite dimensional case $s = \infty$.

1.1 Infinite dimensional multiindices

In the following we write $\boldsymbol{k} = (k_j)_{j\in\mathbb{N}} \in \mathbb{N}_0^{\mathbb{N}}$ (where $\mathbb{N}_0 = \{0, 1, 2, \dots\}$) for a multiindex. Contrary to our previous notation for $s < \infty$, in case $s = \infty$ it is more convenient to consider multiindices with $k_j \in \mathbb{N}_0$ (instead of $k_j \in \mathbb{N}$ as before). We only consider multiindices such that $|\boldsymbol{k}| := \sum_{j\in\mathbb{N}} k_j < \infty$. That means $\boldsymbol{k}$ has finite support in the sense that $\operatorname{supp}(\boldsymbol{k}) = \{j \in \mathbb{N} : k_j \neq 0\} \subseteq \mathbb{N}$ has finite cardinality. We denote the set of such multiindices by

$$\mathcal{F} := \{\boldsymbol{k} \in \mathbb{N}_0^{\mathbb{N}} : |\boldsymbol{k}| < \infty\}.$$

For $\boldsymbol{m}, \boldsymbol{k} \in \mathcal{F}$ we write $\boldsymbol{m} \leq \boldsymbol{k}$ iff $m_j \leq k_j$ for all $j \in \mathbb{N}$, and the multiindex with all zero entries is denoted by $\boldsymbol{0} = (0)_{j\in\mathbb{N}} \in \mathcal{F}$.

Definition 1.1. *We call a set $\mathcal{J} \subseteq \mathcal{F}$* downward closed *if $\boldsymbol{k} - \boldsymbol{e}_j \in \mathcal{J}$ for each $\boldsymbol{k} \in \mathcal{J}$ and each $j \in \operatorname{supp} \boldsymbol{k}$.*

Throughout, for $\boldsymbol{k} \in \mathcal{F}$ and $\boldsymbol{\rho} = (\rho_j)_{j\in\mathbb{N}}$, a sequence of real numbers, we denote

$$\boldsymbol{\rho}^{\boldsymbol{k}} := \prod_{j\in\mathbb{N}} \rho_j^{k_j} = \prod_{j\in\operatorname{supp}\boldsymbol{k}} \rho_j^{k_j}.$$

1.2 Lagrange interpolation

We begin by recalling the one dimensional Lagrange interpolation operators. For $n \in \mathbb{N}_0$ and $f \in C^0(\mathbb{R})$ we denote the one dimensional *Lagrange interpolant*

$$I_n[f](x) = \sum_{j=0}^{n} f(\chi_{n,j})\ell_{n,j}(x), \qquad \ell_{n,j}(x) = \prod_{\substack{i=0\\ i\neq j}}^{n} \frac{x - \chi_{n,i}}{\chi_{n,j} - \chi_{n,i}}, \tag{1.1}$$

where for each $n \in \mathbb{N}$, $(\chi_{n,j})_{j=0}^n$ is a distinct sequence of points in $[-1, 1]$ (the "interpolation nodes"). Additionally, in case $n = 0$ we let $\ell_{0,0} \equiv 1$ be the constant 1 function.

There hold the following facts:

- I_n is a well-defined map from $C^0([-1,1])$ to the space $\mathbb{P}_n$ of polynomials of degree n,
- $I_n[f](\chi_{n,i}) = f(\chi_{n,i})$ for all $i \in \{0,\dots,n\}$,
- $I_n[f] = f$ whenever $f \in \mathbb{P}_n$.

1.3 Multivariate Lagrange interpolation

1.3.1 Tensorized interpolation

We wish to approximate functions f mapping from $[-1,1]^{\mathbb{N}}$ to $\mathbb{R}$, i.e. f assigns a real value to each sequence $\boldsymbol{y} = (y_j)_{j\in\mathbb{N}}$ with $y_j \in [-1,1]$. For a multiindex $\boldsymbol{k} \in \mathcal{F}$ introduce the tensorized interpolation operator

$$I_{\boldsymbol{k}} = \otimes_{j\in\mathbb{N}} I_{k_j}.$$

By definition, I_{k_j} acts on the j th variable, i.e. for $f : [-1,1]^{\mathbb{N}} \to \mathbb{R}$: For $\boldsymbol{x} = (x_j)_{j\in\mathbb{N}} \in [-1,1]^{\mathbb{N}}$

$$I_{\boldsymbol{k}}[f](\boldsymbol{x}) := \sum_{i_1=0}^{k_1} \sum_{i_2=0}^{k_2} \dots f(\chi_{k_1,i_1}, \chi_{k_2,i_2}, \dots) \prod_{j\in\mathbb{N}} \ell_{k_j,i_j}(x_j).$$

Since we chose $\boldsymbol{k} \in \mathcal{F}$, the infinite number of sums and the infinite product are in fact finite: First note that $k_j = 0$ for every $j \notin \operatorname{supp}\boldsymbol{k}$, and since $\ell_{0,0} \equiv 1$, we have $\prod_{j\in\mathbb{N}} \ell_{k_j,i_j}(x_j) = \prod_{j\in\operatorname{supp}\boldsymbol{k}} \ell_{k_j,i_j}(x_j)$ which is a finite product (for $\boldsymbol{k} = \boldsymbol{0}$, we have $\operatorname{supp}\boldsymbol{k} = \emptyset$, and this case we use the convention $\prod_{j\in\operatorname{supp}\boldsymbol{k}} \ell_{k_j,i_j}(x_j) := 1$ for this empty product). Furthermore, since $k_j = 0$ for all $j > J := \max\{i \,:\, i \in \operatorname{supp}\boldsymbol{k}\}$, we can write

$$I_{\boldsymbol{k}}[f](\boldsymbol{x}) = \sum_{i_1=0}^{k_1} \cdots \sum_{i_J=0}^{k_J} f(\chi_{k_1,i_1}, \dots, \chi_{k_J,i_J}, \chi_{0,0}, \chi_{0,0}, \dots) \prod_{j\in\operatorname{supp}\boldsymbol{k}} \ell_{k_j,i_j}(x_j).$$

Finally, a more compact way to write this is

$$I_{\boldsymbol{k}}[f](\boldsymbol{x}) = \sum_{\{\boldsymbol{m}\in\mathcal{F}\,:\,\boldsymbol{m}\le\boldsymbol{k}\}} f((\chi_{k_j,m_j})_{j\in\mathbb{N}}) \prod_{j\in\operatorname{supp}\boldsymbol{m}} \ell_{k_j,m_j}(x_j).$$

We emphasize again that the set $\{\boldsymbol{m} \in \mathcal{F} \,:\, \boldsymbol{m} \le \boldsymbol{k}\}$ is finite, so that $I_{\boldsymbol{k}}[f](\boldsymbol{x})$ can be evaluated and computed.

Let us point out that for $\boldsymbol{m} \le \boldsymbol{k}$ holds

$$I_{\boldsymbol{k}}[\boldsymbol{y}^{\boldsymbol{m}}](\boldsymbol{x}) = \prod_{j\in\operatorname{supp}\boldsymbol{m}} I_{k_j}[y_j^{m_j}](x_j) = \prod_{j\in\operatorname{supp}\boldsymbol{m}} x_j^{m_j}. \tag{1.2}$$

Here we used that $I_{\boldsymbol{k}} = \otimes_{j\in\mathbb{N}} I_{k_j}$, where I_{k_j} only acts on the jth variable y_j. Furthermore we used $I_n[f] = f$ whenever $f \in \mathbb{P}_n$. Thus, (1.2) shows $I_{\boldsymbol{k}}[\boldsymbol{y}^{\boldsymbol{m}}] = \boldsymbol{y}^{\boldsymbol{m}}$ whenever $\boldsymbol{m} \le \boldsymbol{k}$.

1.3.2 Sparse-grid interpolation

Let $\mathcal{J} \subseteq \mathcal{F}$ be a finite downward closed index set. Then we introduce a further interpolation operator

$$I_{\mathcal{J}} = \sum_{\boldsymbol{k} \in \mathcal{J}} \otimes_{j \in \mathbb{N}} (I_{k_j} - I_{k_j - 1}), \tag{1.3}$$

with the convention $I_{-1} \equiv 0$.

There is another representation of this operator as a linear combination of the tensorized operators $I_{\boldsymbol{k}}$ for $\boldsymbol{k} \in \mathcal{J}$: It holds

$$I_{\mathcal{J}} = \sum_{\boldsymbol{k} \in \mathcal{J}} c_{\boldsymbol{k},\mathcal{J}} I_{\boldsymbol{k}} \qquad \text{with the coefficients} \qquad c_{\boldsymbol{k},\mathcal{J}} = \sum_{\{\boldsymbol{e} \in \{0,1\}^{\mathbb{N}} : \boldsymbol{k} + \boldsymbol{e} \in \mathcal{J}\}} (-1)^{|\boldsymbol{e}|}. \tag{1.4}$$

This is known as the combination formula.

Exercise 1.2. *Show* (1.4).

1.4 Properties of $I_{\mathcal{J}}$

For $\boldsymbol{y} = (y_j)_{j \in \mathbb{N}} \in [-1,1]^{\mathbb{N}}$ and $\boldsymbol{k} = (k_j)_{j \in \mathbb{N}} \in \mathbb{N}_0^{\mathbb{N}}$ set

$$\boldsymbol{y}^{\boldsymbol{k}} = \prod_{j \in \mathbb{N}} y_j^{k_j} = \prod_{j \in \operatorname{supp} \boldsymbol{k}} y_j^{k_j}.$$

For a finite set $\mathcal{J}$ define

$$\mathbb{P}_{\mathcal{J}} := \operatorname{span}\{\boldsymbol{y}^{\boldsymbol{k}} \,:\, \boldsymbol{k} \in \mathcal{J}\}.$$

The next lemma justifies the definition of $I_{\mathcal{J}}$ in the previous section. Recall that the one-dimensional Lagrange interpolation operator I_n has the property of reproducing all polynomials of degree at most n, i.e. $I_n[f] = f$ whenever $f \in \mathbb{P}_n$. We now show that the same holds for $I_{\mathcal{J}}$, in the sense that $I_{\mathcal{J}}[f] = f$ whenever $f \in \mathbb{P}_{\mathcal{J}}$.

Lemma 1.3. *Let $\mathcal{J} \subseteq \mathcal{F}$ be finite and downward closed. Then $I_{\mathcal{J}}[f] = f$ for all polynomials $f \in \mathbb{P}_{\mathcal{J}}$.*

Proof. In the following we show $I_{\mathcal{J}}[\boldsymbol{y}^{\boldsymbol{m}}] = \boldsymbol{y}^{\boldsymbol{m}}$ for all $\boldsymbol{m} \in \mathcal{J}$. The linearity of $I_{\mathcal{J}}$ then implies $I_{\mathcal{J}}[f] = f$ for all $f \in \operatorname{span}\{\boldsymbol{y}^{\boldsymbol{m}} \,:\, \boldsymbol{m} \in \mathcal{J}\}$.

Let $\boldsymbol{n} \geq \boldsymbol{k}$ be two multiindices and assume that there exists j such that $n_j > k_j$. For $r \geq s \in \mathbb{N}_0$ we have $I_r[y^s] = y^k$ since I_r is the identity on the space of polynomials with degree $r \geq s$. This implies that $(I_r - I_{r-1})[y^s] \equiv 0$ whenever $r > s$.

Hence, if there exists i such that $n_i > k_i$, then

$$(\otimes_{j \in \mathbb{N}} (I_{n_j} - I_{n_j - 1}))[\boldsymbol{y}^{\boldsymbol{k}}](\boldsymbol{x}) = \prod_{j \in \mathbb{N}} (I_{n_j} - I_{n_j - 1})[y_j^{k_j}](x_j) \equiv 0.$$

Here we used that $(I_{n_j} - I_{n_j - 1})$ only acts on the jth variable y_j, so that, applied to the product $\boldsymbol{y}^{\boldsymbol{k}} = \prod_{j \in \mathbb{N}} y_j^{k_j}$, we obtained a product of functions $\prod_{j \in \mathbb{N}} (I_{n_j} - I_{n_j - 1})[y_j^{k_j}]$, which equals 0 since for the ith term it holds $(I_{n_i} - I_{n_i - 1})[y_i^{k_i}] \equiv 0$. We conclude with (1.3) that

$$I_{\mathcal{J}}[\boldsymbol{y}^{\boldsymbol{m}}] = \sum_{\boldsymbol{k} \in \mathcal{J}} (\otimes_{j \in \mathbb{N}} (I_{k_j} - I_{k_j - 1}))[\boldsymbol{y}^{\boldsymbol{m}}] = \sum_{\{\boldsymbol{k} \in \mathcal{J} \,:\, \boldsymbol{k} \leq \boldsymbol{m}\}} (\otimes_{j \in \mathbb{N}} (I_{k_j} - I_{k_j - 1}))[\boldsymbol{y}^{\boldsymbol{m}}].$$

Next note that the downward closedness of $\mathcal{J}$ actually means that $\{\boldsymbol{k} \in \mathcal{J} \,:\, \boldsymbol{k} \leq \boldsymbol{m}\} = \{\boldsymbol{k} \in \mathcal{F} \,:\, \boldsymbol{k} \leq \boldsymbol{m}\}$ whenever $\boldsymbol{m} \in \mathcal{J}$. Thus for $\boldsymbol{m} \in \mathcal{J}$

$$I_{\mathcal{J}}[\boldsymbol{y}^{\boldsymbol{m}}] = \sum_{\{\boldsymbol{k}\in\mathcal{F}\,:\,\boldsymbol{k}\leq\boldsymbol{m}\}} (\otimes_{j\in\mathbb{N}}(I_{k_j} - I_{k_j-1}))[\boldsymbol{y}^{\boldsymbol{m}}].$$

To conclude the proof we note that with $J = \max\{j \,:\, j \in \operatorname{supp} \boldsymbol{m}\}$

$$\begin{aligned}
\sum_{\{\boldsymbol{k}\in\mathcal{F}\,:\,\boldsymbol{k}\leq\boldsymbol{m}\}} (\otimes_{j\in\mathbb{N}}(I_{k_j} - I_{k_j-1})) &= \sum_{k_1=0}^{m_1} \cdots \sum_{k_J=0}^{m_J} \otimes_{j\leq J}(I_{k_j} - I_{k_j-1}) \otimes_{j>J} I_0 \\
&= \sum_{k_2=0}^{m_2} \cdots \sum_{k_J=0}^{m_J} (I_{m_1} - \underbrace{I_{-1}}_{=0}) \otimes_{j\leq J} (I_{k_j} - I_{k_j-1}) \otimes_{j>J} I_0 \\
&= \cdots = \otimes_{j\in\mathbb{N}} I_{m_j} = I_{\boldsymbol{m}}.
\end{aligned}$$

where we used the multilinearity of the tensor product operator. The statement follows by (1.2). □

1.5 The Lebesgue constant

For I_n in (1.1), the Lebesgue constant is defined as its operator norm as a function mapping from $C^0([-1,1])$ to itself:

$$\Lambda_n := \sup_{\|f\|_{C^0([-1,1])}=1} \|I_n[f]\|_{C^0([-1,1])}$$

where $\|f\|_{C^0([-1,1])} = \max_{x\in[-1,1]} |f(x)|$. Note that Λ_n is a function of the interpolation nodes $(\chi_{n,i})_{i=0}^n$, and in general they should be chosen such as to minimize Λ_n. The reason is that $(1+\Lambda_n)$ provides a bound on how far the interpolant is from the best approximating polynomial: Since I_n is linear and satisfies $I_n[p] = p$ for all $p \in \mathbb{P}_n$, it holds for any $f \in C^0([-1,1])$

$$\begin{aligned}
\|f - I_n[f]\|_{C^0([-1,1])} &= \inf_{q\in\mathbb{P}_n} \|f - I_n[f - q + q]\|_{C^0([-1,1])} \\
&= \inf_{q\in\mathbb{P}_n} \|f - q - I_n[f - q]\|_{C^0([-1,1])} \\
&\leq (1 + \Lambda_n) \inf_{q\in\mathbb{P}_n} \|f - q\|_{C^0([-1,1])}.
\end{aligned}$$

This argument can be generalized to the operators $I_{\mathcal{J}}$, to obtain an error bound on $f - I_{\mathcal{J}}[f]$. For our purpose, it will be sufficient to understand the behaviour of $I_{\mathcal{J}}$ on the multivariate monomials $\boldsymbol{y}^{\boldsymbol{m}}$. In the following, we'll assume that:

$$\exists \tau > 0 : \qquad \Lambda_n \leq (1+n)^\tau \qquad \forall n \in \mathbb{N}. \tag{1.5}$$

We point out again, that this is an assumption on the interpolation nodes. For example, for the Chebyscheff nodes, the Lebesgue constant Λ_n grows merely logarithmically in n (and in particular there exists $\tau > 0$ such that (1.5) holds).

We will need the following bound (which we do not prove here) on the multivariate interpolant $I_{\mathcal{J}}$ applied to the multivariate monomials $\boldsymbol{y}^{\boldsymbol{m}}$. A proof can be found in [1, Lemma 3.1].

Lemma 1.4. *Let $\mathcal{J}$ be downward closed and assume that (1.5) holds. Then for all $\boldsymbol{m} \in \mathcal{F}$*

$$\sup_{\boldsymbol{x}\in[-1,1]^{\mathbb{N}}} |I_{\mathcal{J}}[\boldsymbol{y}^{\boldsymbol{m}}](\boldsymbol{x})| \leq |\mathcal{J}|^{1+\tau}$$

where $|\mathcal{J}|$ denotes the cardinality of the set $\mathcal{J}$.

2 UQ for $s = \infty$

In this section we want to approximate the solution $u(\boldsymbol{y}) \in H_0^1(D)$ to the parametric PDE

$$-\operatorname{div}(a(\boldsymbol{y})\nabla u(\boldsymbol{y})) = f, \qquad u(\boldsymbol{y})|_{\partial D} = 0, \tag{2.1}$$

depending on $\boldsymbol{y} \in [-1,1]^{\mathbb{N}}$. Here all functions depend on $x \in D$, and we assume the diffusion coefficient to be given by

$$a(\boldsymbol{y}, x) := 1 + \sum_{j\in\mathbb{N}} y_j \psi_j(x) \qquad x \in D. \tag{2.2}$$

For simplicity, we use both notations $a(\boldsymbol{y})$ and $a(\boldsymbol{y}, x)$, and omit the x argument unless we wish to emphasize the dependence on x. The norm on $H_0^1(D)$ is given by

$$\|v\|^2_{H_0^1(D)} := \int_D \nabla v \cdot \nabla v \mathrm{d}x,$$

and we work under the following assumptions:

(i) $D \subseteq \mathbb{R}^d$, $d \in \{2, 3\}$, is a bounded Lipschitz domain,

(ii) $f \in H^{-1}(D)$,

(iii) $\psi_j : D \to \mathbb{R}$ such that $\sum_{j\in\mathbb{N}} \|\psi_j\|_{L^\infty(D)} < 1$.

The last condition guarantees that $a(\boldsymbol{y}) \in L^\infty(D)$ is well-defined since for $\boldsymbol{y} \in [-1,1]^{\mathbb{N}}$

$$\|a(\boldsymbol{y})\|_{L^\infty(D)} \le 1 + \sum_{j\in\mathbb{N}} |y_j| \|\psi_j\|_{L^\infty(D)} \le 1 + \sum_{j\in\mathbb{N}} \|\psi_j\|_{L^\infty(D)} < \infty.$$

Additionally $a(\boldsymbol{y})$ satisfies for all $\boldsymbol{y} \in [-1,1]^{\mathbb{N}}$

$$\operatorname*{ess\,inf}_{x\in D} a(\boldsymbol{y}, x) \ge 1 - \sum_{j\in\mathbb{N}} \|\psi_j\| > 0.$$

By the Lax-Milgram lemma we conclude that the following proposition holds:

Proposition 2.1. *For every $\boldsymbol{y} \in [-1,1]^{\mathbb{N}}$ there exists a unique solution $u(\boldsymbol{y}) \in H_0^1(D)$ to (2.1)-(2.2).*

2.1 Linear and multilinear maps

In this section we denote by X and Y two Banach spaces with norms $\|\cdot\|_X$ and $\|\cdot\|_Y$. Later we will choose $X = H_0^1(D)$ and $Y = H^{-1}(D)$, but the following discussion holds in more generality.

2.1.1 Linear maps

We write $L(X,Y)$ to denote the set of bounded linear operators from X to Y. Hence if $A \in L(X,Y)$, then $A : X \to Y$ is linear and

$$\|A\|_{L(X,Y)} = \sup_{\|x\|_X=1} \|Ax\|_Y < \infty.$$

In this case $\|Ax\|_Y \leq \|A\|_{L(X,Y)}\|x\|_X$ for all $x \in X$. For two linear operators $A_1 \in L(X_1, X_2)$, $A_2 \in L(X_2, X_3)$ we write A_2A_1 to denote the composition $A_2 \circ A_1$. The definition of the norm immediately implies $\|A_2A_1x\| \leq \|A_2\|_{L(X_2,X_3)}\|A_1\|_{L(X_1,X_2)}\|x\|_{X_1}$ and thus

$$\|A_2A_1\|_{L(X_1,X_3)} \leq \|A_1\|_{L(X_1,X_2)}\|A_2\|_{L(X_2,X_3)} < \infty$$

so that $A_2A_1 \in L(X_1, X_3)$. Similarly, if $A \in L(X,X)$, then $A^n \in L(X,X)$ stands for n fold composition of A with itself, i.e. $A^n = A \circ \cdots \circ A$, and $\|A^n\|_{L(X,X)} \leq \|A\|^n_{L(X,X)}$.

We call $A \in L(X,Y)$ an *isomorphism*, in case $A : X \to Y$ is bijective. Recall that the open mapping theorem in this case implies $A^{-1} \in L(Y,X)$, that is, the inverse of A is also a bounded linear map.

Example 2.2. *Under our current assumptions (on D), the Laplace operator $-\Delta : H^1_0(D) \to H^{-1}(D)$ is an isomorphism: First of all, $-\Delta : H^1_0(D) \to H^{-1}(D)$ is well-defined. Second, by the Lax-Milgram lemma, for every $g \in H^{-1}(D)$ there exists a unique $v \in H^1_0(D)$ such that $-\Delta v = g$, which shows that $-\Delta$ is bijective. The operator has norm 1 (and is thus bounded) due to*

$$\begin{aligned}
\sup_{\|v\|_{H^1_0(D)}=1} \|-\Delta v\|_{H^{-1}(D)} &= \sup_{\|v\|_{H^1_0(D)}=1} \sup_{\|w\|_{H^1_0(D)}=1} \langle -\Delta v, w\rangle \\
&= \sup_{\|v\|_{H^1_0(D)}=1} \sup_{\|w\|_{H^1_0(D)}=1} \int_D \nabla v \cdot \nabla w \mathrm{d}x \\
&= \sup_{\|v\|_{H^1_0(D)}=1} \int_D \nabla v \cdot \nabla v \mathrm{d}x = 1.
\end{aligned}$$

Here we used that $H^{-1}(D)$ is the dual space of $H^1_0(D)$, so that $\|g\|_{H^{-1}(D)} = \sup_{\|w\|_{H^1_0(D)}=1}\langle g, w\rangle$ for all $g \in H^{-1}(D)$. Also the inverse operator $-\Delta : H^{-1}(D) \to H^1_0(D)$ has norm 1: Let $-\Delta v = g$, i.e. $v = (-\Delta)^{-1}g$. Then

$$\|(-\Delta)^{-1}g\|^2_{H^1_0(D)} = \|v\|^2_{H^1_0(D)} = \int_D \nabla v \cdot \nabla v \mathrm{d}x = \langle g, v\rangle \leq \|v\|_{H^1_0(D)}\|g\|_{H^{-1}(D)},$$

and thus $\|(-\Delta)^{-1}g\|_{H^1_0(D)} \leq \|g\|_{H^{-1}(D)}$.

Theorem 2.3 (Neumann series)**.** *Let $A \in L(X,Y)$ be an isomorphism and $B \in L(X,Y)$ such that $\|B\|_{L(X,Y)} < \|A^{-1}\|^{-1}_{L(Y,X)}$. Then $A - B \in L(X,Y)$ is an isomorphism and*

$$(A-B)^{-1} = \sum_{n\in\mathbb{N}_0} (A^{-1}B)^n A^{-1} \in L(Y,X).$$

Proof. Since A is invertible, we note that $A - B \in L(X, Y)$ is invertible iff $A^{-1}(A - B) \in L(X, X)$ is invertible, and in this case

$$(A - B)^{-1}A = (A^{-1}(A - B))^{-1} = (I_X - A^{-1}B)^{-1}, \tag{2.3}$$

with I_X denoting the identity on X. Thus it suffices to check that $I_X - A^{-1}B$ is boundedly invertible.

Set $C := A^{-1}B$. By assumption

$$\|C\|_{L(X,X)} = \|A^{-1}B\|_{L(X,X)} \leq \|A^{-1}\|_{L(Y,X)}\|B\|_{L(X,Y)} < 1. \tag{2.4}$$

With $C^0 := I_X$ set

$$D := \sum_{n\in\mathbb{N}_0} C^n = \sum_{n\in\mathbb{N}_0} (A^{-1}B)^n,$$

and note that $D \in L(X, X)$ is well-defined due to

$$\|D\|_{L(X,X)} \leq \sum_{n\in\mathbb{N}_0} \|(A^{-1}B)^n\|_{L(X,X)} \leq \sum_{n\in\mathbb{N}_0} \|A^{-1}B\|^n_{L(X,X)} < \infty,$$

where we used (2.4). We claim that $D = (I_X - A^{-1}B)^{-1}$. To verify this we show $(I_X - A^{-1}B)D = I_X$ and $D(I_X - A^{-1}B) = I_X$. The first equality holds by

$$(I_X - A^{-1}B)D = (I_X - C)\sum_{n\in\mathbb{N}_0} C^n = \sum_{n\in\mathbb{N}_0} C^n - \sum_{n\in\mathbb{N}} C^n = C^0 = I_X.$$

Similarly one checks $D(I_X - A^{-1}B) = I_X$.

Finally, by (2.3)

$$(A - B)^{-1} = (I_X - A^{-1}B)^{-1}A^{-1} = \sum_{n\in\mathbb{N}_0} (A^{-1}B)^n A^{-1}. \qquad \square$$

2.1.2 Multilinear maps

Definition 2.4. *We call $M_n : X \times \cdots \times X \to Y$ a multilinear map (or n-linear map) if for $(x_1, \ldots, x_n) \in X^n$ and each $j \in \{1, \ldots, n\}$ the map*

$$x_j \mapsto M_n(x_1, \ldots, x_n) \in Y$$

is linear. We denote

$$\|M_n\| := \sup_{\|x_j\|_X \leq 1 \ \forall j} \|M(x_1, \ldots, x_n)\|_Y.$$

The definition of the norm immediately implies that for any $(x_1, \ldots, x_n)$

$$\|M_n(x_1, \ldots, x_n)\|_Y \leq \|M_n\| \prod_{j=1}^{n} \|x_j\|_X.$$

2.2 Expanding $u(\boldsymbol{y})$

We now get back to the solution $u(\boldsymbol{y})$ of problem (2.1)-(2.2). With $X := H_0^1(D)$ and $Y := H^{-1}(D)$ let $A := -\Delta \in L(X,Y)$ and

$$B_j := \operatorname{div}(\psi_j \nabla \cdot) \in L(X,Y), \qquad B(\boldsymbol{y}) := \sum_{j\in\mathbb{N}} y_j B_j \in L(X,Y).$$

Then by definition of $u(\boldsymbol{y})$ we can write

$$u(\boldsymbol{y}) = (A - B(\boldsymbol{y}))^{-1} f,$$

since $A - B(\boldsymbol{y}) = -\operatorname{div}((1 + \sum_{j\in\mathbb{N}} y_j\psi_j)\nabla\cdot)$. Here $f \in Y$ is the right-hand side in (2.1). By Thm. 2.3 it holds

$$u(\boldsymbol{y}) = \sum_{n\in\mathbb{N}_0} (A^{-1}B(\boldsymbol{y}))^n A^{-1} f. \tag{2.5}$$

With $A^{-1} \in L(Y,X)$, define for arbitrary $C_1, \dots, C_n \in L(X,Y)$

$$M_n(C_1,\dots,C_n) := A^{-1} \underbrace{C_1}_{\in L(X,Y)} \cdots A^{-1} \underbrace{C_n}_{\in L(X,Y)} A^{-1} \underbrace{f}_{\in Y} \in X.$$

Then M_n is an n-linear map from $L(X,Y)\times\cdots\times L(X,Y) \to X$, and (2.5) becomes

$$u(\boldsymbol{y}) = \sum_{n\in\mathbb{N}_0} M_n(B(\boldsymbol{y}),\dots,B(\boldsymbol{y})). \tag{2.6}$$

The idea in the following is to use the multilinearity of each M_n to write this as an expansion in terms of $\boldsymbol{y}$. To this end, for every $\boldsymbol{k} \in \mathcal{F}$ define

$$S_{\boldsymbol{k}} := \{(i_1,\dots,i_{|\boldsymbol{k}|}) \,:\, |\{r \,:\, i_r = j\}| = k_j \ \forall j \in \mathbb{N}\},$$

where $|\{r \,:\, i_r = j\}|$ denotes the cardinality of the set $|\{r \,:\, i_r = j\}|$. For example

$$\text{if} \qquad \boldsymbol{k} = (0,1,2,0,0,\dots) \in \mathcal{F} \qquad \text{then} \qquad S_{\boldsymbol{k}} = \{(2,3,3),(3,2,3),(3,3,2)\}.$$

Note that the cardinality of $S_{\boldsymbol{k}}$ equals

$$|S_{\boldsymbol{k}}| = \frac{|\boldsymbol{k}|!}{\boldsymbol{k}!} \qquad \text{where} \qquad \boldsymbol{k}! := \prod_{j\in\mathbb{N}} k_j!.$$

Now set

$$t_{\boldsymbol{k}} := \sum_{(i_1,\dots,i_{|\boldsymbol{k}|})\in S_{\boldsymbol{k}}} M_{|\boldsymbol{k}|}(B_{i_1},\dots,B_{i_{|\boldsymbol{k}|}}) \in X.$$

We obtain the upper bound

$$\begin{aligned}
\|t_{\boldsymbol{k}}\|_X &\le \sum_{(i_1,\dots,i_{|\boldsymbol{k}|})\in S_{\boldsymbol{k}}} \|M_{|\boldsymbol{k}|}\| \prod_{r=1}^{|\boldsymbol{k}|} \|B_{i_r}\|_{L(X,Y)} \\
&= \sum_{(i_1,\dots,i_{|\boldsymbol{k}|})\in S_{\boldsymbol{k}}} \|M_{|\boldsymbol{k}|}\| \prod_{j\in\mathbb{N}} \|B_j\|_{L(X,Y)}^{k_j} \\
&\le \frac{|\boldsymbol{k}|!}{\boldsymbol{k}!} \|M_{|\boldsymbol{k}|}\| \prod_{j\in\mathbb{N}} \|B_j\|_{L(X,Y)}^{k_j}
\end{aligned} \tag{2.7}$$

where we used the definition of $S_{\boldsymbol{k}}$.

Lemma 2.5. *It holds*

$$\|B_j\|_{L(X,Y)} = \|-\operatorname{div}(\psi_j \nabla \cdot)\|_{L(X,Y)} \leq \|\psi_j\|_{L^\infty(D)}.$$

Proof. Exercise. Hint: Use that $Y = H^{-1}(D)$ is the dual space of $X = H_0^1(D)$ so that $\|v\|_Y = \sup_{\|w\|_X=1} \langle v, w \rangle$. □

Lemma 2.6. *For all* $n \in \mathbb{N}_0$

$$\|M_n\| \leq \|f\|_Y.$$

Proof. For all $C_j \in L(X,Y)$

$$\|A^{-1} C_1 \cdots C_n A^{-1} f\|_Y \leq \|A^{-1}\|_{L(Y,X)} \|C_1\|_{L(X,Y)} \cdots \|C_1\|_{L(X,Y)} \|A^{-1}\|_{L(Y,X)} \|f\|_Y.$$

The lemma follows by definition of $\|M_n\|$ and the fact that $\|A^{-1}\|_{L(Y,X)} \leq 1$ by Example 2.2. □

Theorem 2.7. *Assume* $\sum_{j\in\mathbb{N}} \|\psi_j\|_{L^\infty(D)} < 1$. *Then for all* $\boldsymbol{y} \in [-1,1]^{\mathbb{N}}$

$$u(\boldsymbol{y}) = \sum_{\boldsymbol{k}\in\mathcal{F}} t_{\boldsymbol{k}} \boldsymbol{y}^{\boldsymbol{k}}. \tag{2.8}$$

Moreover, if $\boldsymbol{\rho} = (\rho_j)_{j\in\mathbb{N}}$ *is a sequence of positive numbers such that* $\sum_{j\in\mathbb{N}} \|\psi_j\|_{L^\infty(D)} \rho_j < 1$, *then*

$$\sum_{\boldsymbol{k}\in\mathcal{F}} \boldsymbol{\rho}^{\boldsymbol{k}} \|t_{\boldsymbol{k}}\|_X < \infty. \tag{2.9}$$

Proof. We start with (2.9). By (2.7) and Lemma 2.5

$$\begin{aligned}
\sum_{\boldsymbol{k}\in\mathcal{F}} \boldsymbol{\rho}^{\boldsymbol{k}} \|t_{\boldsymbol{k}}\|_X &= \sum_{n\in\mathbb{N}_0} \sum_{|\boldsymbol{k}|=n} \boldsymbol{\rho}^{\boldsymbol{k}} \|t_{\boldsymbol{k}}\|_X \\
&\leq \|f\|_Y \sum_{n\in\mathbb{N}_0} \sum_{|\boldsymbol{k}|=n} \frac{n!}{\boldsymbol{k}!} \boldsymbol{\rho}^{\boldsymbol{k}} \prod_{j\in\mathbb{N}} \|B_j\|^{k_j} \\
&= \|f\|_Y \sum_{n\in\mathbb{N}_0} \sum_{|\boldsymbol{k}|=n} \frac{n!}{\boldsymbol{k}!} \prod_{j\in\mathbb{N}} (\rho_j \|\psi_j\|_{L^\infty(D)})^{k_j}.
\end{aligned}$$

Now observe that

$$\sum_{|\boldsymbol{k}|=n} \frac{n!}{\boldsymbol{k}!} \prod_{j\in\mathbb{N}} (\rho_j \|\psi_j\|_{L(D)})^{k_j} = \left(\sum_{j\in\mathbb{N}} \rho_j \|\psi_j\|_{L^\infty(D)} \right)^n,$$

since each of the product terms on the left-hand side occurs exactly $\frac{n!}{\boldsymbol{k}!}$ times on the right-hand side (this is known as the *multinomial theorem*, which is a generalization of the binomial theorem). Our assumptions imply that

$$r := \sum_{j\in\mathbb{N}} \rho_j \|\psi_j\|_{L^\infty(D)} < 1,$$

and thus ultimately

$$\sum_{\boldsymbol{k}\in\mathcal{F}} \|t_{\boldsymbol{k}}\|_X \boldsymbol{\rho}^{\boldsymbol{k}} \leq \|f\|_Y \sum_{n\in\mathbb{N}_0} r^n < \infty.$$

Equation (2.8) formally follows by expanding (2.6):

$$\begin{aligned} u(\boldsymbol{y}) = \sum_{n\in\mathbb{N}_0} M_n(B(\boldsymbol{y}),\dots,B(\boldsymbol{y})) &= \sum_{n\in\mathbb{N}_0} M_n\Big(\sum_{j_1\in\mathbb{N}} y_{j_1}B_{j_1},\dots,\sum_{j_n\in\mathbb{N}} y_{j_n}B_{j_n}\Big) \\ &= \sum_{n\in\mathbb{N}_0}\sum_{j_1\in\mathbb{N}}\cdots\sum_{j_n\in\mathbb{N}} M_n(B_{j_1},\dots,B_{j_n})\prod_{i=1}^{n} y_{j_i} \\ &= \sum_{n\in\mathbb{N}_0}\sum_{\{\boldsymbol{k}\in\mathcal{F}\,:\,|\boldsymbol{k}|=n\}}\sum_{(j_1,\dots,j_n)\in S_{\boldsymbol{k}}} M_n(B_{j_1},\dots,B_{j_n})\boldsymbol{y}^{\boldsymbol{k}} \\ &= \sum_{\boldsymbol{k}\in\mathcal{F}} \boldsymbol{y}^{\boldsymbol{k}} t_{\boldsymbol{k}}. \end{aligned}$$

This formal reordering of the sum is justified if the last sum is absolutely convergent. This holds by (2.9) with the choice $\rho_j := 1$ for all j, since for $\boldsymbol{y}\in[-1,1]^{\mathbb{N}}$

$$\sum_{\boldsymbol{k}\in\mathcal{F}} |\boldsymbol{y}^{\boldsymbol{k}}|\|t_{\boldsymbol{k}}\|_X = \sum_{\boldsymbol{k}\in\mathcal{F}} |\boldsymbol{y}^{\boldsymbol{k}}|\|t_{\boldsymbol{k}}\|_X\boldsymbol{\rho}^{\boldsymbol{k}} \le \sum_{\boldsymbol{k}\in\mathcal{F}} \|t_{\boldsymbol{k}}\|_X\boldsymbol{\rho}^{\boldsymbol{k}} < \infty$$

due to the assumption $\sum_{j\in\mathbb{N}}\|\psi_j\|_{L^\infty(D)}\rho_j = \sum_{j\in\mathbb{N}}\|\psi_j\|_{L^\infty(D)} < 1$. □

Remark 2.8. *By* (2.8)*, for any* $\boldsymbol{m}\in\mathcal{F}$

$$\frac{\partial^{|\boldsymbol{m}|}}{\partial_{y_1}^{m_1}\partial_{y_2}^{m_2}\dots}u(\boldsymbol{y})|_{\boldsymbol{y}=0} = \sum_{\boldsymbol{k}\in\mathcal{F}} t_{\boldsymbol{k}}\frac{\partial^{|\boldsymbol{m}|}}{\partial_{y_1}^{m_1}\partial_{y_2}^{m_2}\dots}\boldsymbol{y}^{\boldsymbol{k}}|_{\boldsymbol{y}=0}\boldsymbol{y}^{\boldsymbol{m}} = \boldsymbol{m}!t_{\boldsymbol{m}},$$

where we formally exchanged the derivative with the summation over $\boldsymbol{k}$ *(this can be made rigorous). Thus*

$$t_{\boldsymbol{m}} = \frac{1}{\boldsymbol{m}!}\frac{\partial^{|\boldsymbol{m}|}}{\partial_{y_1}^{m_1}\partial_{y_2}^{m_2}\dots}u(\boldsymbol{y})|_{\boldsymbol{y}=0},$$

and (2.8) *can be interpreted as a Taylor expansion in infinitely many variables.*

2.3 Interpolation error

In this section we'll see that we can approximate u with an algebraic convergence rate. The rate will depend on how fast the sequence $(\|\psi_j\|_{L^\infty(D)})_{j\in\mathbb{N}}$ tends to 0. Roughly speaking, we will see if we have the bound $\|\psi_j\|_{L^\infty(D)} \le Cj^{-\gamma}$ for some $\gamma > 1$, then we can uniformly approximate the function $\boldsymbol{y}\mapsto u(\boldsymbol{y})$ at a converence rate proportional to γ. Hence, the larger γ, the stronger the decay of the sequence $(\|\psi_j\|_{L^\infty(D)})_{j\in\mathbb{N}}$ and the faster the convergence rate.

Lemma 2.9. *Let* $\boldsymbol{\rho} = (\rho_j)_{j\in\mathbb{N}}$ *be a sequence of numbers larger than one and let* $p\in(0,1]$ *be such that* $\sum_{j\in\mathbb{N}}\rho_j^{-p} < \infty$*. Then*

$$\sum_{\boldsymbol{k}\in\mathcal{F}} (\boldsymbol{\rho}^{-\boldsymbol{k}})^p < \infty.$$

Proof. It holds

$$\sum_{\boldsymbol{k}\in\mathcal{F}} (\boldsymbol{\rho}^{-\boldsymbol{k}})^p = \sum_{\boldsymbol{k}\in\mathcal{F}}\prod_{j\in\mathbb{N}}\rho_j^{-pk_j} = \prod_{j\in\mathbb{N}}\sum_{n\in\mathbb{N}_0}\rho_j^{-pn} = \prod_{j\in\mathbb{N}}\frac{1}{1-\rho_j^{-p}}.$$

With $\kappa := \max_{j\in\mathbb{N}} \rho_j^{-p} < 1$ we have $\frac{1}{1-\rho_j^{-p}} \le 1 + \frac{1}{1-\kappa}\rho_j^{-p}$. Since $\log(1+x) \le x$ for all $x \ge 0$,

$$\prod_{j\in\mathbb{N}} \frac{1}{1-\rho_j^{-p}} = \exp\left(\sum_{j\in\mathbb{N}} \log(1+\frac{1}{1-\kappa}\rho_j^{-p})\right) \le \exp\left(\sum_{j\in\mathbb{N}} \frac{1}{1-\kappa}\rho_j^{-p}\right) < \infty. \qquad \square$$

Lemma 2.10. *Let $q > 0$ and $(a_j)_{j\in\mathbb{N}} \in \ell^q(\mathbb{N})$. If $(a_j)_{j\in\mathbb{N}}$ is monotonically decreasing then $a_N \le N^{-\frac{1}{q}}(\sum_{j\in\mathbb{N}} a_j^q)^{\frac{1}{q}}$.*

Proof. Since $(a_j)_{j\in\mathbb{N}}$ is monotonically decreasing, for every $N \in \mathbb{N}$

$$a_N^q \le \frac{1}{N}\sum_{j=1}^{N} a_j^q \le \frac{1}{N}\sum_{j\in\mathbb{N}} a_j^q$$

and consequently

$$a_N \le N^{-1/q}\left(\sum_{j\in\mathbb{N}} a_j^q\right)^{1/q}. \qquad \square$$

Let us now define sets of multiindices as follows: given $\varepsilon > 0$ and a sequence $(\rho_j)_{j\in\mathbb{N}}$ we let

$$\mathcal{J}_\varepsilon := \{\boldsymbol{k} \in \mathcal{F} \,:\, \boldsymbol{\rho}^{-\boldsymbol{k}} > \varepsilon\}. \tag{2.10}$$

This strategy is called *thresholding*: We choose all multiindices $\boldsymbol{k}$ corresponding to the largest values of $\boldsymbol{\rho}^{-\boldsymbol{k}}$, with the threshold given by $\varepsilon > 0$. We now show that these multiindex sets provide suitable polynomial spaces to approximate u: the following theorem proves a convergence rate for the polynomial expansion and the polynomial interpolant. Since this is an infinite dimensional problem, the following result shows that the curse of dimensionality can be overcome.

Theorem 2.11. *Let $p \in (0,1)$ and assume that $\sum_{j\in\mathbb{N}} \|\psi_j\|_{L^\infty(D)}^p < 1$. Define*

$$\rho_j := \|\psi_j\|_{L^\infty(D)}^{p-1} > 1,$$

and let $\mathcal{J}_\varepsilon \subseteq \mathcal{F}$ be as in (2.10).

Then there exists $C > 0$ such that

(i) for all $\varepsilon > 0$

$$\sup_{\boldsymbol{y}\in[-1,1]^{\mathbb{N}}} \left\| u(\boldsymbol{y}) - \sum_{\boldsymbol{k}\in\mathcal{J}_\varepsilon} \boldsymbol{y}^{\boldsymbol{k}} t_{\boldsymbol{k}} \right\|_X \le C|\mathcal{J}_\varepsilon|^{-\frac{1}{p}+1}$$

(ii) if the interpolation nodes satisfy the bound (1.5) *on the Lebesgue constant, then for all $\varepsilon > 0$*

$$\sup_{\boldsymbol{y}\in[-1,1]^{\mathbb{N}}} \|u(\boldsymbol{y}) - I_{\mathcal{J}_\varepsilon}[u](\boldsymbol{y})\|_X \le C|\mathcal{J}_\varepsilon|^{-\frac{1}{p}+2+\tau}.$$

Proof. By (2.8)

$$\sup_{\boldsymbol{y}\in[-1,1]^{\mathbb{N}}}\left\|u(\boldsymbol{y})-\sum_{\boldsymbol{k}\in\mathcal{J}_\varepsilon}\boldsymbol{y}^{\boldsymbol{k}}t_{\boldsymbol{k}}\right\|_X\le\sup_{\boldsymbol{y}\in[-1,1]^{\mathbb{N}}}\sum_{\boldsymbol{k}\in\mathcal{F}\setminus\mathcal{J}_\varepsilon}|\boldsymbol{y}^{\boldsymbol{k}}|\,\|t_{\boldsymbol{k}}\|_X\le\sum_{\boldsymbol{k}\in\mathcal{F}\setminus\mathcal{J}_\varepsilon}\|t_{\boldsymbol{k}}\|_X\,,$$

since $|\boldsymbol{y}^{\boldsymbol{k}}|\le 1$ for all $\boldsymbol{y}\in[-1,1]^{\mathbb{N}}$.

Since $\rho_j^{-p/(1-p)}=\|\psi_j\|^p$ and $\sum_{j\in\mathbb{N}}\|\psi_j\|^p<\infty$, Lemma 2.9 yields with $q:=\frac{p}{1-p}$ that

$$\sum_{\boldsymbol{k}\in\mathcal{F}}(\boldsymbol{\rho}^{-\boldsymbol{k}})^q<\infty.$$

Moreover $\sum_{j\in\mathbb{N}}\|\psi_j\|\rho_j=\sum_{j\in\mathbb{N}}\|\psi_j\|^p<1$ and Thm. 2.7 gives

$$\sum_{\boldsymbol{k}\in\mathcal{F}}\|t_{\boldsymbol{k}}\|_X\boldsymbol{\rho}^{\boldsymbol{k}}=:C_0<\infty. \tag{2.11}$$

Now, by (2.11) and Lemma 2.10

$$\sum_{\boldsymbol{k}\in\mathcal{F}\setminus\mathcal{J}_\varepsilon}\|t_{\boldsymbol{k}}\|_X=\sum_{\boldsymbol{k}\in\mathcal{F}\setminus\mathcal{J}_\varepsilon}\|t_{\boldsymbol{k}}\|_X\,\boldsymbol{\rho}^{-\boldsymbol{k}}\boldsymbol{\rho}^{\boldsymbol{k}}\le C_0\sum_{\boldsymbol{k}\in\mathcal{F}\setminus\mathcal{J}_\varepsilon}\boldsymbol{\rho}^{-\boldsymbol{k}}\le C|\mathcal{J}_\varepsilon|^{-\frac{1}{q}}=C|\mathcal{J}_\varepsilon|^{-\frac{1}{p}+1}, \tag{2.12}$$

for some $C>0$ independent of $\varepsilon>0$.

Finally, using Lemma 1.4, and the fact that $I_{\mathcal{J}_\varepsilon}[\boldsymbol{x}^{\boldsymbol{k}}]=\boldsymbol{x}^{\boldsymbol{k}}$ whenever $\boldsymbol{k}\in\mathcal{J}_\varepsilon$ (shown in Lemma 1.3)

$$\begin{aligned}\sup_{\boldsymbol{y}\in[-1,1]^{\mathbb{N}}}\|u(\boldsymbol{y})-I_{\mathcal{J}_\varepsilon}[u](\boldsymbol{y})\|_X&=\sup_{\boldsymbol{y}\in[-1,1]^{\mathbb{N}}}\left\|\sum_{\boldsymbol{k}\in\mathcal{F}}t_{\boldsymbol{k}}(\boldsymbol{y}^{\boldsymbol{k}}-I_{\mathcal{J}_\varepsilon}[\boldsymbol{x}^{\boldsymbol{k}}](\boldsymbol{y}))\right\|_X\\&\le\sum_{\boldsymbol{k}\in\mathcal{F}\setminus\mathcal{J}_\varepsilon}\|t_{\boldsymbol{k}}\|_X\sup_{\boldsymbol{y}\in[-1,1]^{\mathbb{N}}}|\boldsymbol{y}^{\boldsymbol{k}}-I_{\mathcal{J}_\varepsilon}[\boldsymbol{x}^{\boldsymbol{k}}](\boldsymbol{y})|\\&\le(1+|\mathcal{J}_\varepsilon|^{1+\tau})\sum_{\boldsymbol{k}\in\mathcal{F}\setminus\mathcal{J}_\varepsilon}\|t_{\boldsymbol{k}}\|_X\\&\le C|\mathcal{J}_\varepsilon|^{-\frac{1}{p}+1+\tau+1},\end{aligned}$$

where we used again (2.12). □

Remark 2.12. *By a slightly more involved analysis one can show the convergence rate* $\frac{1}{p}-1$ *(instead of* $\frac{1}{p}-\tau-2$*) also for the interpolant* $I_{\mathcal{J}}$ *in Thm. 2.11 (ii).*

Remark 2.13. *The statement in Thm. 2.11 (i) is often referred to as "best* N*-term approximation": we approximate* u *by the truncating* $u(\boldsymbol{y})=\sum_{\boldsymbol{k}\in\mathcal{F}}t_{\boldsymbol{k}}\boldsymbol{y}^{\boldsymbol{k}}$ *after the "best"* N *terms in this expansion.*

Remark 2.14. *The statement in Thm. 2.11 (ii) gives a convergence rate for the interpolant in terms of cardinality of the set* $\mathcal{J}_\varepsilon$*. In case the chosen interpolation points are nested in the sense that* $\chi_{n,i}=\chi_{m,i}$ *whenever* $i\le n<m$*, then* $|\mathcal{J}_\varepsilon|$ *equals the number of required function evaluations of* u *to compute the interpolant* $I_{\mathcal{J}_\varepsilon}u$*. Hence this convergence rate is also valid in terms of the number of required evaluations of* u*.*

References

[1] A. Chkifa, A. Cohen, and C. Schwab. High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. *Found. Comput. Math.*, 14(4):601–633, 2014.

High-Dimensional Approximation and Applications in Uncertainty Quantification

VI. Adaptive High-Dimensional Approximation Methods

Prof. Dr. Robert Scheichl
r.scheichl@uni-heidelberg.de

Dr. Alexander Gilbert
a.gilbert@uni-heidelberg.de

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Institut für Angewandte Mathematik, Universität Heidelberg

Summer Semester 2020

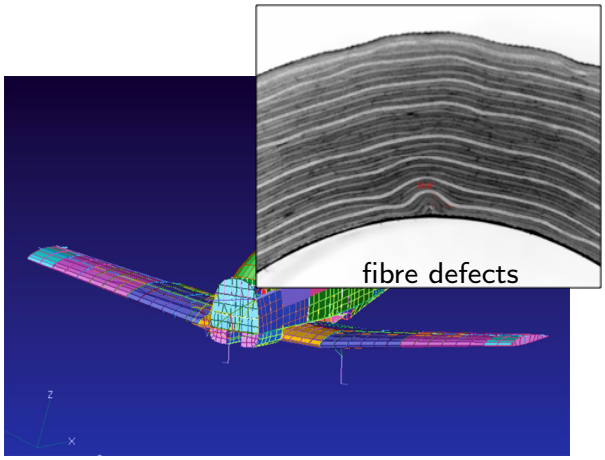
1. Motivation – Need for Adaptivity

Examples from real applications

- **Structural Mechanics** (e.g. aerospace composites, additive manufacture):

$$\nabla \cdot \left(\overline{\overline{C}}(x,\omega) : \frac{1}{2} \left[\nabla \mathbf{u} + \nabla \mathbf{u}^T \right] \right) + \mathbf{F} \;=\; 0 \quad \text{in} \quad D$$

subject to BCs



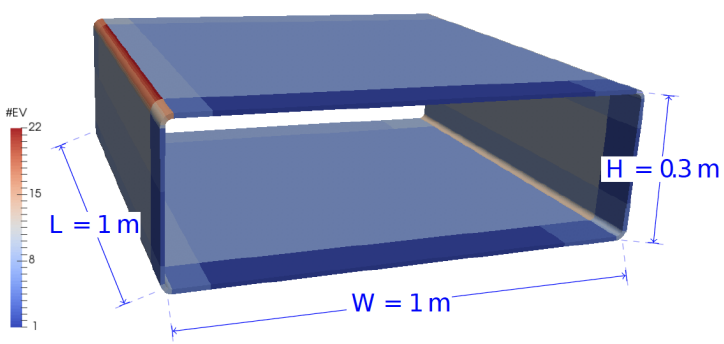
©MX3D, Netherlands

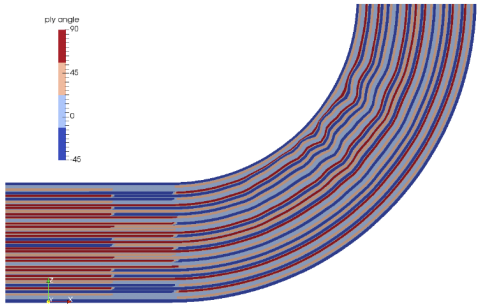
Examples from real applications

- **Structural Mechanics** (e.g. aerospace composites, additive manufacture):

$$\nabla \cdot \left(\overline{\overline{C}}(x,\omega) : \frac{1}{2} \left[\nabla \mathbf{u} + \nabla \mathbf{u}^T \right] \right) + \mathbf{F}(x,\omega) \;=\; 0 \quad \text{in} \quad D(\omega)$$

subject to BCs

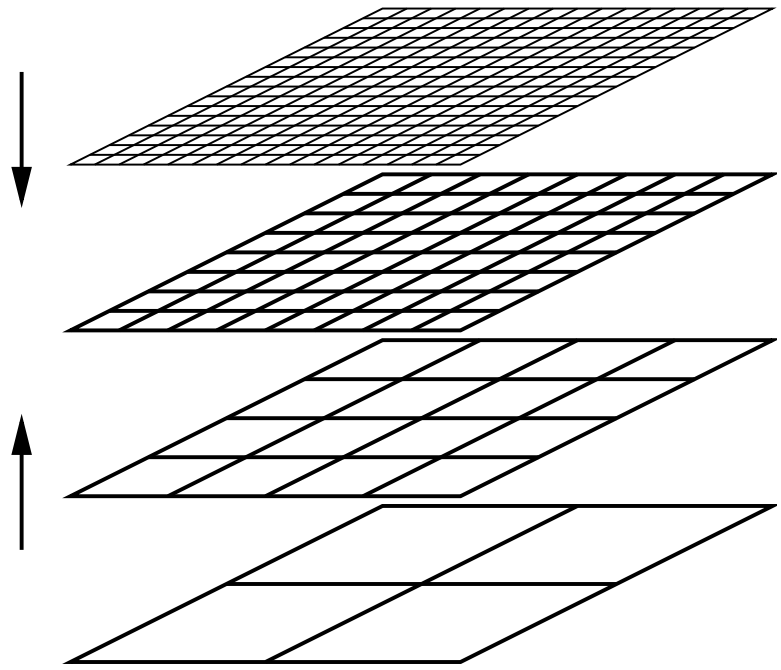




Dune [Butler, Dodwell, Reinarz, Sandhu, RS, Seelinger, '19]

Spatial Mesh Hierarchy (so far)

$\mathcal{T}_L$



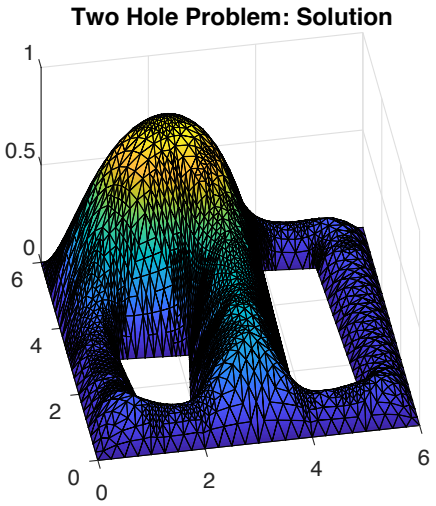
$\mathcal{T}_0$

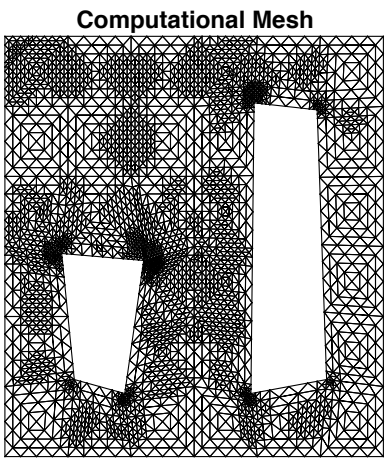
Uniform!

Model Problems

- Model elliptic problem with **uncertain source** or **uncertain geometry**:

$$\begin{aligned} -\Delta u &= f(x,\omega) && \text{in } D(\omega) \\ \text{subject to} \quad u &= 0 && \text{on } \partial D(\omega) \end{aligned}$$





$\Rightarrow$ **Spatial Adaptivity (sample-wise) !**

2. Multilevel Stochastic Collocation

Stochastic Collocation Method (Section IV.5) [Xiu, Hesthaven, '05]

Stochastic PDE example in parametric form in $D \times \Gamma \subset \mathbb{R}^{d \times s}$:

$$-\nabla \cdot \Big(a(x,y)\nabla u(x,y)\Big) = f(x,y), \quad (x,y) \in D \times \Gamma \text{ and } u|_{\partial D} \equiv 0 \qquad (2.1)$$

where $y = (y_1, \ldots, y_s) \in \Gamma = \Gamma_1 \times \cdots \times \Gamma_s$ with Γ_j **bounded** (Assumption)

- Use sampling points $\{y^{(i)}\}_{i=1,\ldots,N}$ in Γ and
- FE solutions $u_\ell(y^{(i)}) \in V_\ell \subset V$ (w.r.t. mesh $\mathcal{T}_\ell$) and $Q_\ell(y^{(i)}) = \psi(u_\ell(y^{(i)}))$
- to construct the (single-level) **interpolant** (here only for functionals)

$$Q_{N,\ell}^{(SL)}(y) = \mathcal{I}_N[Q_\ell](y) = \sum_{i=1}^{N} Q_\ell(y^{(i)})\phi_i(y) \qquad (2.2)$$

in the polynomial space $\mathcal{P}_N = \text{span}\{\phi_i\}_{i=1}^N \subset L^2_\rho(\Gamma)$ such that

$\mathcal{I}[Q_\ell](y^{(i)}) = Q_\ell(y^{(i)}), \quad$ for $\; i = 1, \ldots, N$ (interpolating condition)

- $\mathbb{E}[Q_\ell]$ approximated by integrating $Q_{N,\ell}^{(SL)}$ (repeated 1D integrals)

Adaptive Sparse Grid Stochastic Collocation

Problem:

- Curse of Dimensionality for **full tensor grid** (N **exponential** in s!)

Remedy:

- **Anisotropic** Smolyak sparse grids [Nobile, Tempone, Webster, 2008].
- Can choose collocation points $\{y^{(i)}\}_{i=1,\ldots,N}$ s.t.

$$\big|\mathbb{E}\big[Q_\ell - \mathcal{I}_N[Q_\ell]\big]\big| \leq C(s) N^{-\mu(s)} \qquad (2.3)$$

under suitable regularity conditions [Nobile, Tamelini, Tempone, '16], [Haji-Ali, Harbrecht, Peters, Siebenmorgen, 2018].

In general, $\mu = \mu(s)$ (i.e. **dimension dependent**) but under suitable p-sparsity assumptions $\mu = \frac{2}{p} - 1$ [**Zech**, 2018] (see Chapter V).

- Can compute $\{y^{(i)}\}_{i=1,\ldots,N}$ **adaptively** using a posteriori estimation of errors via surplus operators (see Section IV.6).
  [Gerstner, Griebel, 2003], [Schieche, Lang, 2014], [Guignard, Nobile, 2018]

Multilevel Stochastic Collocation [Teckentrup, Jantsch, Webster et al., '15]

$$Q_L^{(ML)} = \sum_{\ell=0}^{L} \mathcal{I}_{N_{L-\ell}}[Q_\ell - Q_{\ell-1}] = \sum_{\ell=0}^{L} \left(Q^{(SL)}_{N_{L-\ell},\ell} - Q^{(SL)}_{N_{L-\ell},\ell-1} \right) \tag{2.4}$$

where $Q_\ell = \psi(u_\ell)$ with u_ℓ computed on $\mathcal{T}_\ell$ with $h_\ell \sim 2^{-\ell}$ (uniform in y).

Theorem 2.1 (Multilevel Complexity Theorem [Teckentrup, '13])

Let us assume there are $\alpha, \beta, \gamma, \mu > 0$ such that

(M1) $|\mathbb{E}[Q_\ell - Q]| = \mathcal{O}(h_\ell^\alpha)$,

(M2) $\left|\mathbb{E}\left[Q_\ell - Q_{\ell-1} - \mathcal{I}_{N_{L-\ell}}[Q_\ell - Q_{\ell-1}]\right]\right| = \mathcal{O}(N_{L-\ell}^{-\mu} h_\ell^\beta)$,

(M3) Cost/sample on Level ℓ $= \mathcal{O}(h_\ell^{-\gamma})$.

Then, for any $\varepsilon < e^{-1}$, there are L and $\{N_\ell\}_{\ell=0}^L$ s.t. $\left|\mathbb{E}[Q - Q_L^{(ML)}]\right| \le \varepsilon$ and

$$\mathit{Cost}\left(Q_L^{(ML)}\right) = \mathcal{O}\left(\varepsilon^{-\frac{1}{\mu} - \max\left(0, \frac{\gamma\mu - \beta}{\alpha\mu}\right)}\right) \quad \textit{(+ possible log-factor)}$$

Usually $\beta \approx \alpha$ and $\frac{1}{\mu} < \frac{\gamma}{\alpha} \Rightarrow \mathrm{Cost}\left(Q_L^{(ML)}\right) \approx \mathcal{O}\left(\varepsilon^{-\frac{\gamma}{\alpha}}\right)$ (cost/sample on finest level !)

3. Sample-Adaptive Finite Element Spaces

Fully Adaptive Multilevel Stochastic Collocation

Combine sample-wise spatial adaptivity and adaptive sparse grids

Using a posteriori error estimators:

- compute $u_\ell(y^{(j)}) \in V_\ell(y^{(j)})$ s.t. $|Q(y^{(j)}) - Q_\ell(y^{(j)})| \leq \eta_{X_\ell}$ (prescribed)
  (e.g. using the dual weighted residual method (DWRM) [Becker, Rannacher, '01])
- and choose $\{y^{(j)}\}_{j=1,\dots,N}$ **adaptively** such that

$$\left|\mathbb{E}\left[Q_\ell - Q_{\ell-1} - \mathcal{I}_{N_{L-\ell}}[Q_\ell - Q_{\ell-1}]\right]\right| = C(s) \cdot \eta_{Y_{L-\ell}} \qquad \text{(optimised)}$$

Theorem 3.1 (Adaptive Complexity Theorem [Lang, **RS**, Silvester, 2020]**)**

Let $\eta_{X_\ell} = q^\ell \eta_{X_0}$, for some $q \in (0,1)$ and $\eta_{X_0} > 0$, and suppose $\exists\, t, \mu > 0$ s.t.

$$\begin{aligned} \big(\textit{Cost/sample}\big)_\ell &= \mathcal{O}\left(\eta_{X_\ell}^{-t}\right) \\ \eta_{Y_{L-\ell}} &= \mathcal{O}\left(N_{L-\ell}^{-\mu} \eta_{X_{\ell-1}}\right) \end{aligned}$$

often $\frac{1}{\mu} < t < \frac{\gamma}{\alpha}$!

Then there exist L, $\{\eta_{Y_\ell}\}_{\ell=0}^L$ (explicit) to obtain $\left|\mathbb{E}[Q - Q_L^{(ML)}]\right| \leq \varepsilon$ with

$$\textit{Cost}\left(Q_L^{(ML)}\right) = \mathcal{O}\left(\varepsilon^{-\max\left(\frac{1}{\mu}, t\right)}\right)$$

(+ possible log-factor)

Comments

- Many different ways to estimate errors and to adapt spatial/stochastic grids.
- With optimal linear solver (such as multigrid), we can typically achieve (A1) with $t = d/2$ (for functionals).
- Adaptive spatial schemes with guaranteed convergence exist [Dörfler, '96]
- With suitable choice of collocation points, (A2) follows from (2.3).
- But rigorous proof of convergence for adaptive sparse grid stochastic collocation still lacking (see below).
- **Analogous results for interpolation.**

For the numerical experiments we use:

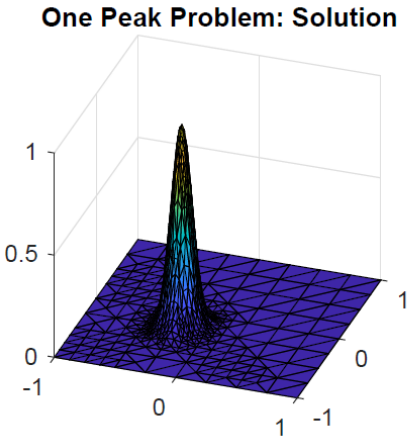
- Matlab package `p1afem`: adaptive piecewise linear FEs and the DWRM (`http://www.asc.tuwien.ac.at/~praetorius/matlab`)
- Matlab package *Sparse Grid Kit*: adaptive Smolyak algorithm (`https://www.epfl.ch/labs/csqi/`)

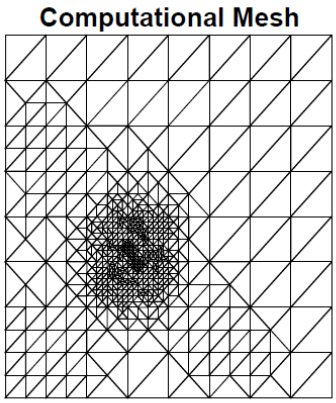
4. Numerical Experiments

Numerical Example (uncertain source): $d = 2$, $s = 2$

- Poisson equation $-\nabla^2 u = f$ on $D = (-1, 1)^2$ with $u|_{\partial D} = 0$
- **Random source location** $(y_1, y_2) \sim \mathcal{U}[-1/4, 1/4]^2$ with $f(x, y)$ s.t.

$$u(x, y) = \exp\left[50\left(\alpha(y_1)(x_1 - y_1)^2 + (x_2 - y_2)^2\right)\right] \quad \text{with} \quad \alpha(y_1) = 18y_1 + \tfrac{11}{2}$$





- Quantity of interest: $Q = \psi(u) = \int_D u^2 \, \mathrm{d}x$
- Full H^2-regularity, but strong local refinement near centre of source: $t \approx 1$

Numerical Example (uncertain source): $d = 2$, $s = 2$

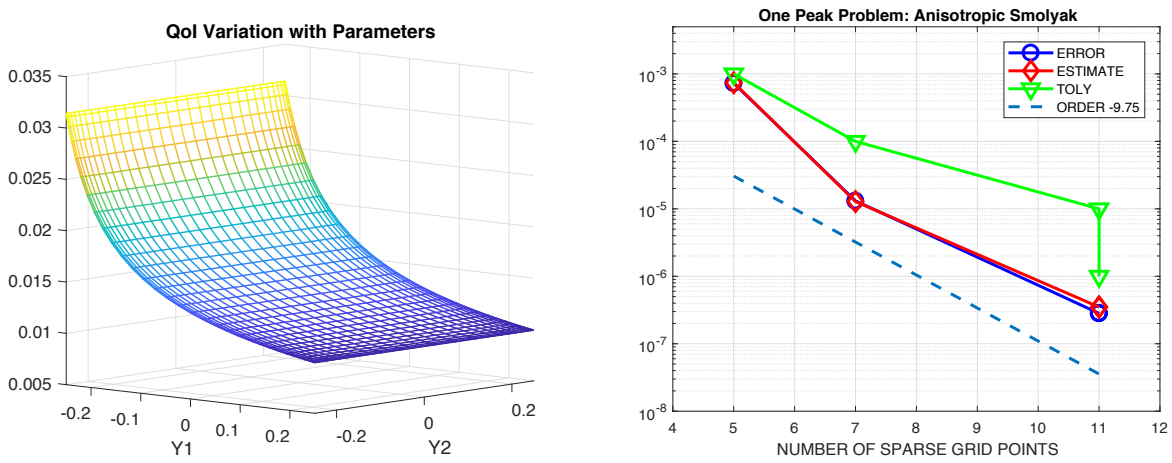


Figure: Very smooth parameter dependence (left); consequently, very fast convergence of anisotropic Smolyak algorithm (right). The estimated convergence order for $\mathbb{E}[\psi(u)]$ in terms of collocation points is -9.75.

The adaptive Smolyak rules are in fact one-dimensional rules in the y_1 direction and **11 collocation points** are suuficient for a tolerance of $\eta_Y = 10^{-6}$!

Numerical Example (uncertain source): $d = 2$, $s = 2$

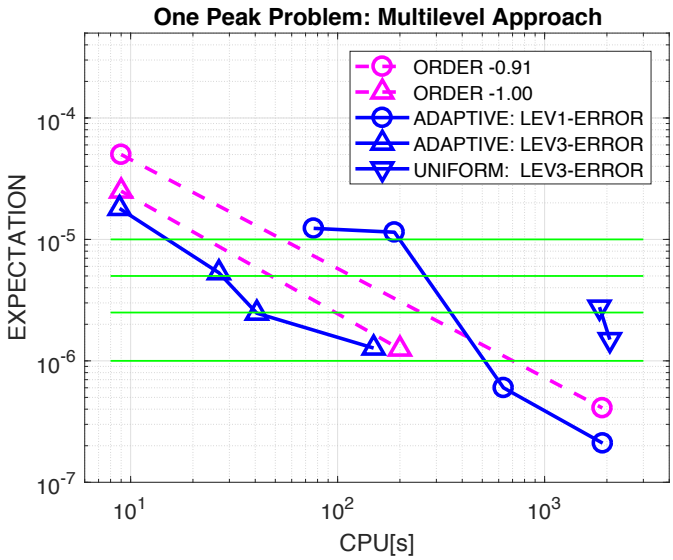
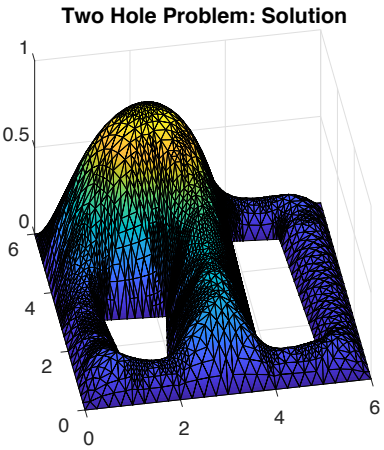


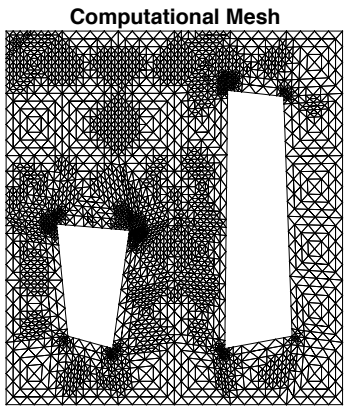
Figure: Errors in the expected values $\mathbb{E}[Q]$: Comparing 3-level adaptive MLSC, adaptive SLSC, 3-level (uniform) MLSC for $\epsilon = 10^{-5}, 5 \times 10^{-6}, 2.5 \times 10^{-6}, 10^{-6}$. The orders of convergence predicted by Theorem 3.1 for aMLSC and aSLSC are -1 and -0.91, resp.

To achieve an accuracy of $\epsilon = 2.8 \times 10^{-6}$ with three-level adaptive MLMC, requires $N_2 = 243$ and $N_0 > 500000$ and 7.8×10^4 seconds.

Numerical Example (uncertain domain): $d = 2$, $s = 16$

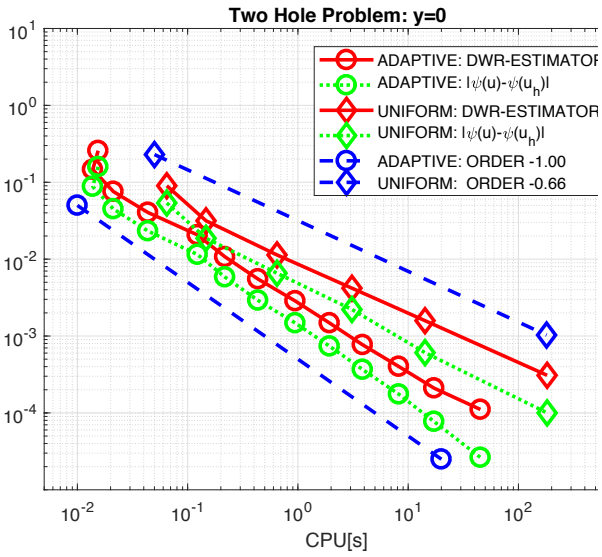
- Poisson eqn. $-\nabla^2 u = f$ with $u|_{\partial D} = 0$ and $f \equiv 1$ on **random domain** $D(y)$:
  (coordinates of corners of holes uniformly distributed, i.e. $s = 16$ parameters)



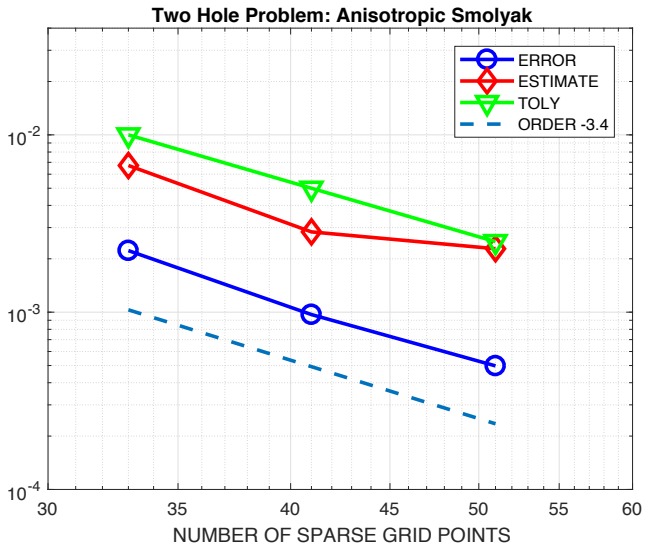


- Quantity of interest: $Q = \psi(u) = \int_D u^2 \, \mathrm{d}x$
- Regularity: u at least in $H^{\frac{13}{8}}(D)$ and no better than $H^{\frac{5}{3}}(D)$
  (can be reformulated as variable coefficient problem to apply analysis)
- Adaptive FEM: $t \approx 1$; Uniform FEM: $1.5 < t < 1.6$ (using MG)

Numerical Example (uncertain domain): $d = 2$, $s = 16$

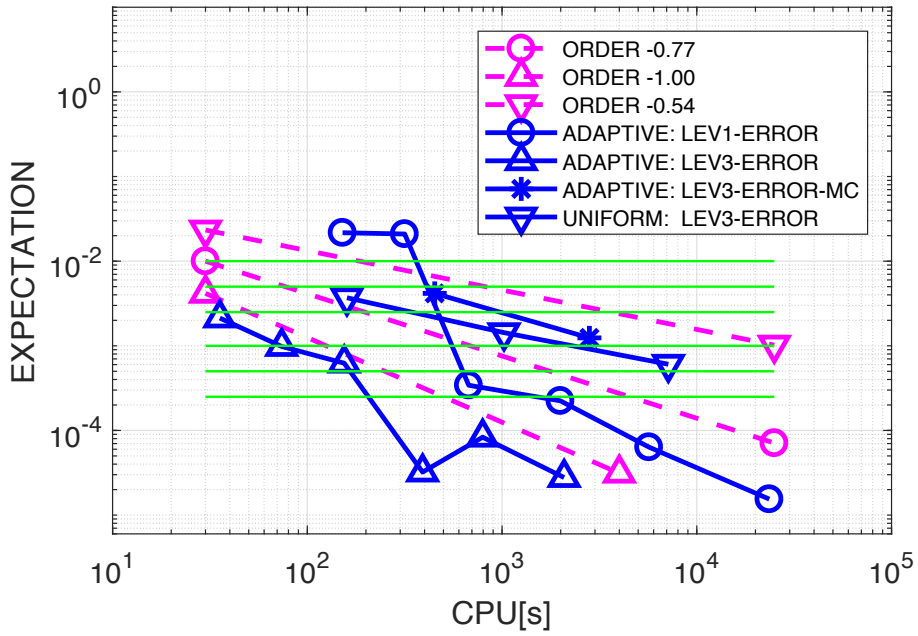


Spatial convergence rates $1/t$
(for rectangular holes)



Sparse grid convergence rate μ

Numerical Example (uncertain domain): $d = 2$, $s = 16$



Comparing AMLSC, ASLSC, MLSC & MLMC w. uniform grid refinement

(tolerances (green): $\varepsilon = 10^{-2}, 5\times10^{-2}, 2.5\times10^{-2}, 10^{-3}, 5\times10^{-4}, 2.5\times10^{-4}$)

5. Conclusions & Further Reading

Conclusions & Further Reading

- Often very smooth parameter dependence: **Huge gains** of **multilevel stochastic collocation** over **multilevel Monte Carlo**.
- For problems with **random local features**: **Huge gains** through **adaptive, sample-dependent FE spaces**.
- So far **no rigorous convergence theory** for adaptive stochastic collocation.
- **Alternative** to stochastic collocation: **Stochastic Galerkin**
  Use orthogonal basis and integrate against test function (as for FEs) also in stochastic variable (instead of collocation at interpolation points).
  - Ghanem & Spanos, *Stochastic FEs: A Spectral Approach*, Springer, NY, 1991
  - Schwab & Gittelson, Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs, *Acta Num*, **20**, 2011
  - Lord, Powell & Shardlow, An Introduction to Computational Stochastic PDEs, Cambridge University Press, 2014
- Due to **Galerkin orthogonality** property, **rigorous convergence** analysis for **adaptive algorithms** possible:
  - Eigel, Gittelson, Schwab & Zander, A convergent adaptive stochastic Galerkin FE method with quasi-optimal spatial meshes, *ESAIM M^2AN*, **49**, 2015
  - Bespalov, Praetorius, Rocchi & Ruggeri, Convergence of adaptive stochastic Galerkin FEM, *SIAM J Numer Anal*, **57**, 2019

Conclusions & Further Reading

- **But** leads to **big, coupled** linear system and due to tensor product structure, **no sample-wise adaptivity** at individual stochastic grid points possible.
- **Current research in our group:** local hp-adaptivity in stochastic space.
- **Other References:**
  - Gerstner & Griebel, Dimension-adaptive tensor-product quadrature, *Computing*, **71**, 2003
  - Xiu & Hesthaven, High-order collocation methods for differential equations with random inputs, *SIAM J Sci Comput*, **27**, 2005
  - Nobile, Tempone & Webster, A sparse grid stochastic collocation method for PDEs with random input data, *SIAM J Numer Anal*, **46**, 2008
  - Teckentrup, Jantsch, Webster & Gunzburger, A multilevel stoch. collocation method for PDE with random input data, *SIAM/ASA J Uncertainty Q*, **3**, 2015
  - Guignard & Nobile, A posteriori error estimation for the stochastic collocation FE method, *SIAM J Numer Anal*, **56**, 2018
  - **Zech**, Sparse-grid approximation of high-dimensional parametric PDEs, PhD Thesis, ETH Zürich, 2018
  - Lang, **RS** & Silvester, A fully adaptive multilevel stochastic collocation strategy for solving elliptic PDEs with random data, *J Comput Phys*, 2020

High-Dimensional Approximation and Applications in Uncertainty Quantification

VII. Low-Rank Tensor Formats and TT Cross Approximation

Prof. Dr. Robert Scheichl
r.scheichl@uni-heidelberg.de

Dr. Alexander Gilbert
a.gilbert@uni-heidelberg.de

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Institut für Angewandte Mathematik, Universität Heidelberg

Summer Semester 2020

1. High-Dimensional Approximation

High-dimensional problems in mechanics and mhysics

Many problems of computational science, probability and statistics require the approximation, integration or optimization of functions of many variables

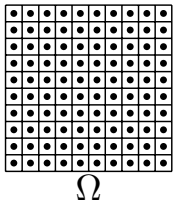
$$u(x_1, \ldots, x_d)$$

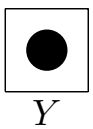
- Navier Stokes equation

  Find $u(x,t)$: $\quad \dfrac{\partial u}{\partial t} + u \cdot \nabla u - \nu \Delta u = f, \quad x \in \Omega \subset \mathbb{R}^d,\ 0 < t < T$

- Multiscale problems

  Find $u(x,y,t), \quad x \in \Omega,\ y \in Y$ where





- Schrödinger equation

  Find $\Psi(x_1, \ldots, x_d, t)$: $\quad i\hbar \dfrac{\partial \Psi}{\partial t} = -\dfrac{\hbar}{2\mu} \Delta \Psi + V \Psi$

High-dimensional problems in statistics and data science

- Unsupervised learning: Estimation of the probability distribution

$$F(x_1, \dots, x_d) = \mathbb{P}(X_1 \leq x_1, \dots, X_d \leq x_d),$$

  of random vector $X = (X_1, \dots, X_d)$ from samples of X or a function of X.

- Supervised learning: Approximation of a random variable Y by a function of a set of random variables $X = (X_1, \dots, X_d)$, using samples of (X, Y). The approximation is used as a predictive model.

- These are two typical tasks in Uncertainty Quantification, where Y is some output variable of a (numerical or experimental) model depending on a set of random parameters X.

The following high level, abstract introduction of high dimensional approximation and low-rank tensor formats is following

A. Nouy, Deep Tensor Networks, Mini-Course, Airbus Group, Paris, June 2019

https://anthony-nouy.github.io/tutorials.html

High-dimensional approximation

Goal. Approximate a function $u(x_1, \dots, x_d)$ by an element of a subset of functions X_n described by n parameters.

- X_n is called an approximation tool, model class or hypothesis set, e.g. splines, wavelets, polynomials (with or without adaptivity).

- For a function u from a normed space, the best approximation error

$$e_n(u) = \inf_{v \in X_n} \|u - v\|,$$

  quantifies what we can expect from X_n.

- We distinguish linear approximation, where X_n are linear spaces, from nonlinear approximation, where X_n are nonlinear spaces.

Fundamental questions.

- determine the complexity $n = n(\varepsilon, u)$ required for obtaining an error

$$e_n(u) \leq \varepsilon,$$

- provide practical approximation algorithms that achieve this precision with almost optimal complexity (using available information on the function).

The Curse of Dimensionality

For a function u from classical regularity classes (Sobolev or Besov spaces) and for standard approximation tools (polynomials, splines, wavelets), it is known that

$$n(\epsilon, u) \lesssim \epsilon^{-d/k} .$$

- We observe that $n(\epsilon, u)$ grows exponentially with the dimension d, which is the curse of dimensionality.
- Better performance observed for particular functions and approximation tools.
- But a priori, we can not expect a better performance from any (reasonable) approximation tool without further assumptions on the function.

To **break** the curse of dimensionality we have to

- make stronger assumptions on the structure of the function, beyond standard assumptions (see Chapters III-V on mixed smoothness classes),
- propose approximation tools (model classes) that capture these structures.

Some standard model classes

- Linear models
$$a_1 x_1 + \ldots + a_d x_d$$
- Polynomial or more general sparse tensor models
$$\sum_{\alpha \in \Lambda} a_\alpha x_1^{\alpha_1} \ldots x_d^{\alpha_d} \quad \text{or} \quad \sum_{\alpha \in \Lambda} a_\alpha \varphi^1_{\alpha_1}(x_1) \ldots \varphi^d_{\alpha_d}(x_d)$$
with $\Lambda \subset \mathbb{N}^d$ a set of multi-indices, either fixed (linear) or free (nonlinear).
- Additive or multiplicative models
$$u_1(x_1) + \ldots + u_d(x_d) \quad \text{or} \quad u_1(x_1) \ldots u_d(x_d)$$
or more generally
$$\sum_{\alpha \subset T} u_\alpha(x_\alpha) \quad \text{or} \quad \prod_{\alpha \in T} u_\alpha(x_\alpha)$$
where $T \subset 2^{\{1,\ldots,d\}}$ is again either fixed (linear approximation) or a free parameter (nonlinear approximation). An instance of a graphical model.

Composition of functions

$f(g(x))$ using standard model classes for both f and g.

- Linear transformations (ridge functions) $g = Wx$, $W \in \mathbb{R}^{m \times d}$, with an additive model for f:

$$f_1(w_1^T x) + \ldots + f_m(w_m^T x) \quad \text{(projection pursuit)}$$

- A special case is the sum of m perceptrons:

$$\sum_{i=1}^{m} a_i \sigma(w_i^T x + b_i),$$

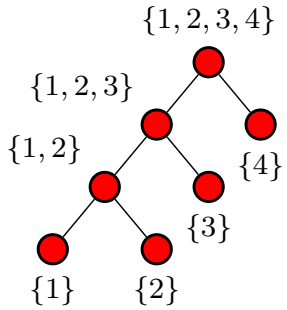
  i.e. a shallow neural network with one hidden layer of width m.

- Sparse transformations, e.g. $f(g_{1,2}(x_1, x_2), g_{3,4}(x_3, x_4), \ldots)$.
- More compositions $f \circ g_L \circ g_{L-1} \circ \ldots \circ g_1(x) \;\rightarrow$ deep neural networks

Deep Neural Networks

- Recurrent networks: sparse transformations with sparsity induced by a linear tree

$$f_{1,2,3,4}\left(f_{1,2,3}\left(f_{1,2}\left(f_1(x_1), f_2(x_2)\right), f_3(x_3)\right), f_4(x_4)\right)$$



These are highly nonlinear approximation tools, with a high approximation power.

Known to achieve optimal performance for standard regularity classes, but cannot expect better than classical tools without further assumptions on the function.

However, **even if the expected error $e_n(u)$ is small** for a certain function u,

- there is no known, certified algorithm for constructing an approximation achieving this error,
- and a best approximation (when it exists) may be highly unstable.

2. Multilinear Maps and Tensor Networks

Low-rank formats (more details below)

A multivariate function $u(x_1, \ldots, x_d)$ can be identified with an order-d tensor.

- The rank of an order-2 tensor $u \in V \otimes W$ (a matrix if $V = \mathbb{R}^m$, $W = \mathbb{R}^n$), denoted $\mathrm{rank}(u)$, is the minimal integer r such that

$$u = \sum_{k=1}^{r} v_k \otimes w_k\,, \quad \text{for some } \; v_k \in V,\; w_k \in W.$$

- Can be computed via singular value decomposition (SVD) and truncated SVD (after the largest r terms) provides a best rank-r approximation of u.
- Extension to order-d tensors:
  - Canonical rank one (multiplicative model): $v(x) = u_1(x_1) \ldots u_d(x_d)$
  - Canonical tensor format with rank less than r:

$$v(x) = \sum_{k=1}^{r} u_1^k(x_1) \ldots u_d^k(x_d)$$

- **But** canonical format and canonical rank not practically useful for $d > 2$.

Tensor formats and α-rank

A better notion for rank for high-order tensors is the following:
(again more details below)

- Let $\alpha \subset \{1, \dots, d\} := D$ and let x_α and x_{α^c} denote the complementary groups of variables. Then $u(x)$ can be identified with a bivariate function

$$\tilde{u}(x_\alpha, x_{\alpha^c})$$

and the rank of this bivariate function $\tilde{u}$ is called the α-rank of u, denoted $\mathrm{rank}_\alpha(v)$. It is the minimal integer r_α such that

$$u(x) = \sum_{k=1}^{r_\alpha} v_k^\alpha(x_\alpha) w_k^{\alpha^c}(x_{\alpha^c})$$

- Any collection $T \subset 2^D$ of subsets of D defines a tensor format

$$\mathcal{T}_r^T = \{v : \mathrm{rank}_\alpha(v) \le r_\alpha, \ \forall \alpha \in T\}.$$

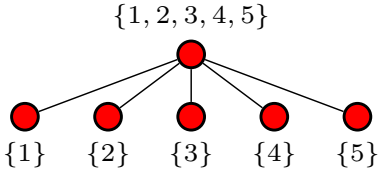
Tree-based tensor formats

- When T is a dimension partition tree over D with

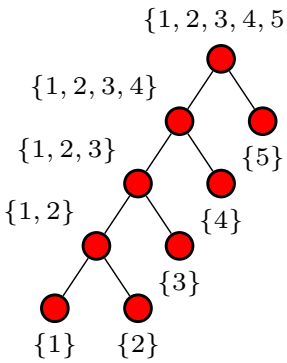
$$\textit{root} \ \ D \ \ \text{and} \ \ \textit{leaves} \ \ \mathcal{L}(T) = \{\{\nu\} : 1 \le \nu \le d\}$$

such that, for any $\alpha \in T$, the sons $S(\alpha)$ form a non-tivial partition of α, then $\mathcal{T}_r^T$ defines a tree-based tensor format.

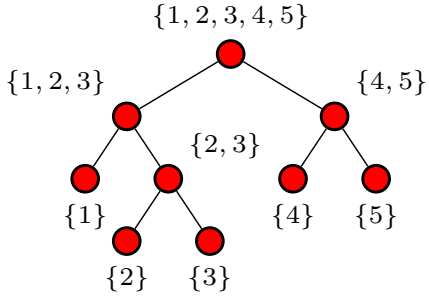
- **Prominent examples** of tree-based tensor formats:



Tucker



Tensor Train



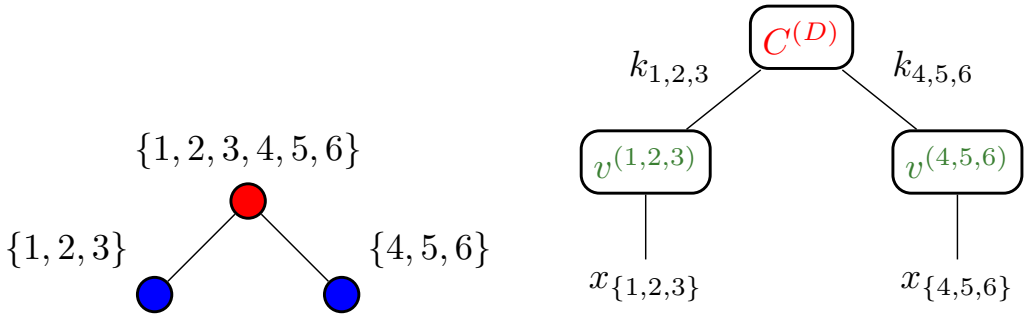
Hierarchical Tucker

Tree-based tensor formats

Elements of $\mathcal{T}_r^T$ admit an explicit representation. Let $u \in \mathcal{T}_r^T$ with T-rank $r = (r_\alpha)_{\alpha \in T}$. At the first level, v admits the representation

$$v(x) = \sum_{k_{\beta_1}=1}^{r_{\beta_1}} \cdots \sum_{k_{\beta_s}=1}^{r_{\beta_s}} C^{(D)}_{k_{\beta_1},\ldots,k_{\beta_s}} v^{(\beta_1)}_{k_{\beta_1}}(x_{\beta_1}) \ldots v^{(\beta_s)}_{k_{\beta_s}}(x_{\beta_s})$$

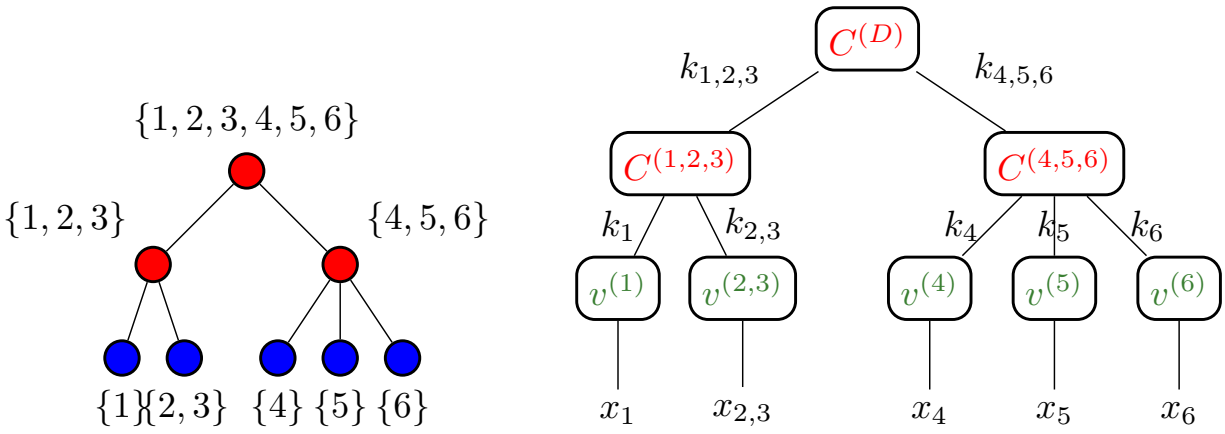
where $\{\beta_1, \ldots, \beta_s\} = S(D)$ are the children of the root node and $\{v^{(\beta)}_{k_\beta}\}_{1 \le k_\beta \le r_\beta}$ form a basis of $U^{\min}_\beta(v)$, the minimal subspace s.t. $\tilde{v}(x_\beta, x_{\beta^c}) \in U^{\min}_\beta(v) \otimes V_{\beta^c}$
(spanned by dominant r_β singular vectors)



Tree-based tensor formats

Then, for an interior node α of the tree, with children $S(\alpha) = \{\beta_1, \ldots, \beta_s\}$, the functions (or tensors) $v^{(\alpha)}_{k_\alpha}$ admit the representation

$$v^{(\alpha)}_{k_\alpha}(x_\alpha) = \sum_{k_{\beta_1}=1}^{r_{\beta_1}} \cdots \sum_{k_{\beta_s}=1}^{r_{\beta_s}} C^{(\alpha)}_{k_\alpha,k_{\beta_1},\ldots,k_{\beta_s}} v^{(\beta_1)}_{k_{\beta_1}}(x_{\beta_1}) \ldots v^{(\beta_s)}_{k_{\beta_s}}(x_{\beta_s}).$$

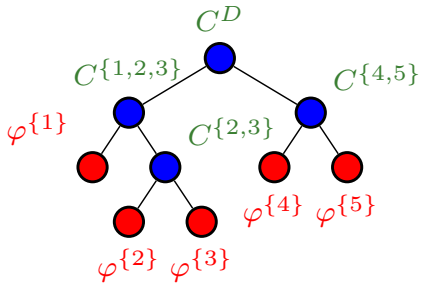


Tree-based tensors as compositions of multilinear maps

For each node α with children $\{\beta_1, \ldots, \beta_s\}$, the tensor C^α in $\mathbb{R}^{r_{\beta_1} \times \ldots \times r_{\beta_s} \times r_\alpha}$ can be identified with a multilinear map from $\mathbb{R}^{r_{\beta_1}} \times \ldots \times \mathbb{R}^{r_{\beta_s}}$ to $\mathbb{R}^{r_\alpha}$.

Also, given bases $\{\phi^\alpha_{i_\alpha}(x_\alpha)\}_{i_\alpha=1}^{n_\alpha}$ of functions for the spaces V_α for $\alpha \in \mathcal{L}(T)$, the leaf-tensors φ^ν can be identified with multilinear maps from $\mathbb{R}^{n_\alpha}$ to $\mathbb{R}^{r_\alpha}$.

Thus, the tree-based format can be written as a composition of multilinear maps. For example,



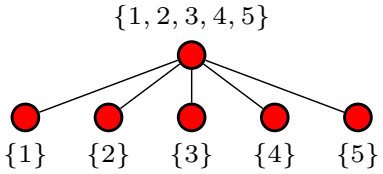
$$v(x) = C^D\left(C^{\{1,2,3\}}\left(\varphi^{\{1\}}(x_1), C^{\{2,3\}}\left(\varphi^{\{2\}}(x_2), \varphi^{\{3\}}(x_3)\right)\right), C^{\{4,5\}}\left(\varphi^{\{4\}}(x_4), \varphi^{\{5\}}(x_5)\right)\right)$$

Tree-based tensor formats correspond to deep neural networks with multilinear mappings and a sparse architecture given by a dimension partition tree.

Storage complexity for the tree-based format ($r_\alpha \leq r$, $n_\nu \leq n$)

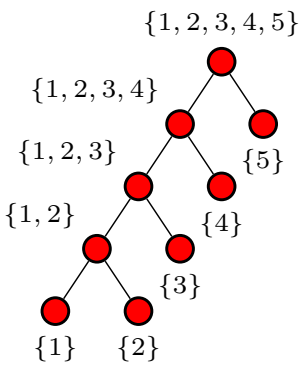
- For the Tucker format (one level), the storage complexity is

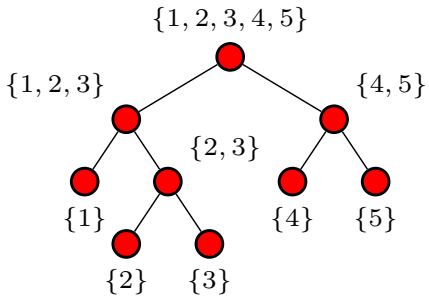
$$C(T, r) = O(dnr + r^d)$$



- For any binary tree such as a linear binary tree (Tensor Train Tucker format) or a balanced binary tree (Hierarchical Tucker format) the storage complexity is

$$C(T, r) = O(dnr + (d-2)r^3) \qquad \text{(linear in } d \text{ and } n\text{)}$$





Computing with tree-based tensor formats

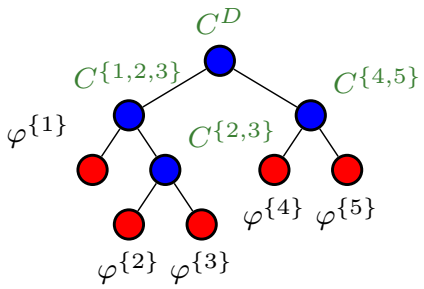
Many favorable properties from a computational point of view:

- Complexity for storage, evaluation, differentiation, integration,... linear in d and cubic in the rank
- "Not so nonlinear" approximation tool. Geometrical properties can be exploited for optimization and dynamical approximation.
- Topological properties ensure the well-posedness of optimization problems and existence of stable algorithms
- Notion of hierarchical singular value decomposition (HSVD; see below) and a way to obtain approximations u_r in $\mathcal{T}_r^T$ such that

$$\|u - u_r\| \leq \sqrt{2d-2} \inf_{v \in \mathcal{T}_r^T} \|u - v\|.$$

- Universal approximation tool, i.e. for any $u \in V$, we can find a sequence $\{u_r\}_{r \geq 1}$ with $u_r \in \mathcal{T}_r^T$ that converges to u.
- **Key question.** How fast does r grow with d and ε^{-1}? Depends on u!

Training a tree-based tensor network



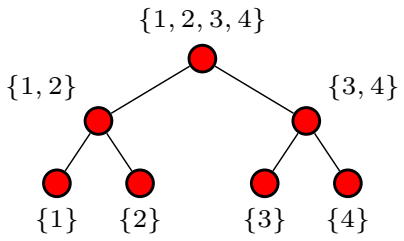
- Simple alternating algorithm for optimization in given tree-based format $\mathcal{T}_r^T$ that exploits the multilinearity of the parametrization.
- At each step, optimization over one parameter (training a linear model!).
- Efficient strategy for rank adaptation based on HSVD.
- Tree adaptation using a stochastic algorithm, able to explore the set of possible trees and recover hidden structures of functions.

A very powerful alternative to deep neural networks!

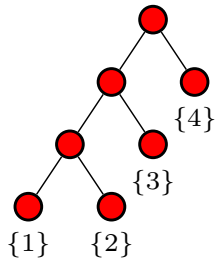
Influence of the tree

- For some functions, the choice of tree is not crucial. For example, an additive function
$$u_1(x_1) + \ldots + u_d(x_d)$$
has α-ranks equal to 2 whatever $\alpha \subset D$.

- But usually, different trees lead to different complexities of representations.



T^B (Balanced tree)



T^L (Linear tree)

  - If $\mathrm{rank}_{T^L}(u) \leq r$ then $\mathrm{rank}_{T^B}(u) \leq r^2$
  - If $\mathrm{rank}_{T^B}(u) \leq r$ then $\mathrm{rank}_{T^L}(u) \leq r^{\log_2(d)/2}$

Example: Canonical versus tree-based format

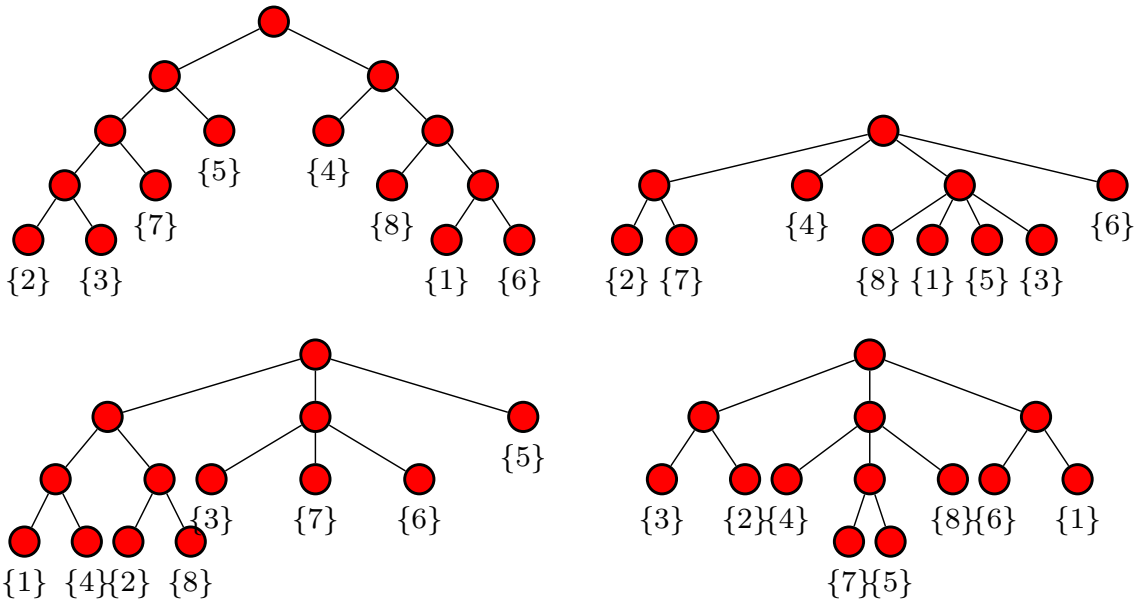
Consider the d-dimensional tensor space $V = \mathbb{R}^n \otimes \ldots \otimes \mathbb{R}^n$.

- From canonical format to binary tree-based format.
For any v in V and any $\alpha \subset D$, the α-rank is bounded by the canonical rank:
$$\mathrm{rank}_\alpha(v) \leq \mathrm{rank}(v).$$
Therefore,
$$\mathcal{R}_r \subset \mathcal{T}_r^T, \quad \text{for any binary tree } T,$$
so that an element in canonical format $\mathcal{R}_r$ with storage complexity $O(dnr)$ admits a representation in $\mathcal{T}_r^T$ with storage complexity $O(dnr + dr^3)$.

- From binary tree-based format to canonical format.
For a balanced or linear binary tree, the subset
$$S = \{v \in \mathcal{T}_r^T : \mathrm{rank}(v) < q^{d/2}\}, \quad q = \min\{n, r\},$$
is of Lebesgue measure 0. Thus, a typical element $v \in \mathcal{T}_r^T$ with storage complexity of order $dnr + dr^3$ admits a representation in canonical format with a storage complexity of order $dnq^{d/2}$.

How to choose a good tree (architecture of the network)?

A crucial but combinatorial problem...



... but stochastic algorithms for tree adaption exist [Grelier, Nouy, Chevreuil, 2018]

Choice of tree – historical note

- Tree-based tensor formats were first introduced in quantum physics and quantum chemistry for the approximation of the high-dimensional solutions of Schrödinger's equation, to deal with so-called quantum entanglement:
  - renormalisation group ideas and matrix product states – linear binary tree (TT) [Wilson, 1975], [White, 1992], [Fannes, Nachtergaele, Werner, 1992], [Perez-Garcia, Verstraete, Wolf, Cirac, 2006],
  - tensor network states – General Hierarchical Trees [Murg, Verstraete, Cirac, 2007], [Schollwöck, 2011], ...
  - multi-configurational time-dependent Hartree (MCTDH) – basic Tucker format [Meyer, Manthe, Cederbaum, 1990], ...
  - multilayer MCTDH – General Hierarchical Trees [Wang, Thoss, 2003], [Vendrell, Meyer, 2011]

  Often due to structure of system, but more often as a compromise for ease of treatment versus control of the ranks and the accuracy.

We will only focus on linear binary trees, the Tensor Train Tucker or TT format.

3. Tensors, Ranks and Singular Value Decompositions

Algebraic tensors

Given d index sets $I_\nu = \{1, \dots, N_\nu\}$, $1 \le \nu \le d$, we introduce the multi-index set

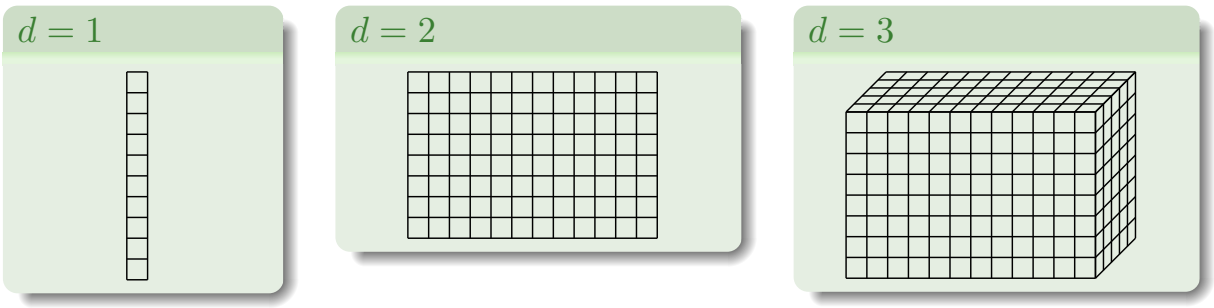
$$I = I_1 \times \ldots \times I_d.$$

An element v of the vector space $\mathbb{R}^I$ is a tensor of order d and is identified with a multidimensional array

$$(v_i)_{i \in I} = (v_{i_1,\ldots,i_d})_{i_1 \in I_1,\ldots,i_d \in I_d}$$

which represents the coefficients of v on the canonical basis of $\mathbb{R}^I$, also denoted

$$v(i) = v(i_1, \dots, i_d).$$



Algebraic tensors

Given d vectors $v^{(\nu)} \in \mathbb{R}^{I_\nu}$, $1 \le \nu \le d$, the tensor product of these vectors

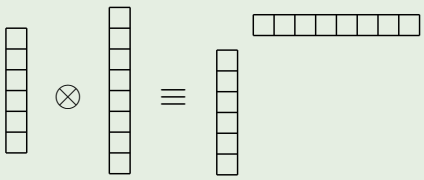
$$v := v^{(1)} \otimes \ldots \otimes v^{(d)}$$

is defined by

$$v(i) = v^{(1)}(i_1) \ldots v^{(d)}(i_d)$$

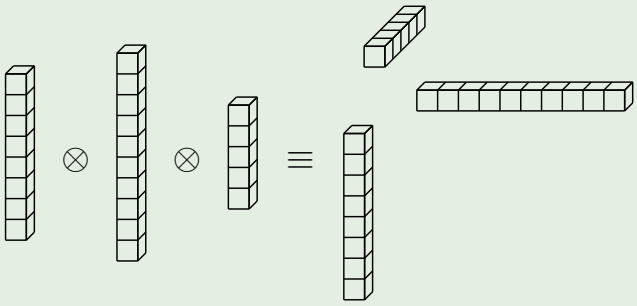
and is called an elementary tensor.

$d = 2$



Using matrix notations, $v \otimes w$ is identified with the matrix vw^T.

$d = 3$



Algebraic tensors

The tensor space $\mathbb{R}^I = \mathbb{R}^{I_1 \times \ldots \times I_d}$, also denoted $\mathbb{R}^{I_1} \otimes \ldots \otimes \mathbb{R}^{I_d}$, is defined by

$$\mathbb{R}^I = \mathbb{R}^{I_1} \otimes \ldots \otimes \mathbb{R}^{I_d} = \operatorname{span}\{v^{(1)} \otimes \ldots \otimes v^{(d)} : v^{(\nu)} \in \mathbb{R}^{I_\nu}, 1 \le \nu \le d\}$$

The canonical norm on $\mathbb{R}^I$, also called the Frobenius norm, is given by

$$\|v\| = \sqrt{\sum_{i \in I} v(i)^2} \tag{3.1}$$

and makes $\mathbb{R}^I$ a Hilbert space with inner product

$$(v^{(1)} \otimes \ldots \otimes v^{(d)}, w^{(1)} \otimes \ldots \otimes w^{(d)}) = (v^{(1)}, w^{(1)})_2 \ldots (v^{(d)}, w^{(d)})_2 \,.$$

It coincides with the natural norm on $\ell_2(I)$ and is the only norm associated with an inner product that has the property

$$\|v^{(1)} \otimes \ldots \otimes v^{(d)}\| = \|v^{(1)}\|_2 \ldots \|v^{(d)}\|_2.$$

Tensor product of functions

Let $\mathcal{X}_\nu \subset \mathbb{R}$, $1 \le \nu \le d$, and $V_\nu \subset \mathbb{R}^{\mathcal{X}_\nu}$ be a space of functions defined on $\mathcal{X}_\nu$.

The tensor product of functions $v^{(\nu)} \in V_\nu$, denoted

$$v = v^{(1)} \otimes \ldots \otimes v^{(d)},$$

is a multivariate function defined on $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$ and such that

$$v(x) = v(x_1, \ldots, x_d) = v^{(1)}(x_1) \ldots v^{(d)}(x_d), \quad \text{for} \quad x = (x_1, \ldots, x_d) \in \mathcal{X}.$$

For example, for $i \in \mathbb{N}_0^d$, the monomial $x^i = x_1^{i_1} \ldots x_d^{i_d}$ is an elementary tensor.

The algebraic tensor product of spaces V_ν is defined as

$$V_1 \otimes \ldots \otimes V_d = \operatorname{span}\{v^{(1)} \otimes \ldots \otimes v^{(d)} : v^{(\nu)} \in V_\nu, 1 \le \nu \le d\}$$

which is the space of multivariate functions v that can be written as a finite sum

$$v(x) = \sum_{k=1}^{n} v_k^{(1)}(x_1) \ldots v_k^{(d)}(x_d).$$

(Can be extended to arbitrary vector spaces V_ν, up to the definition of the tensor product $\otimes$.)

Infinite-dimensional tensor spaces

For infinite dimensional Hilbert spaces V_ν, a Hilbert tensor space with norm $\|\cdot\|$ is obtained by the completion of the algebraic tensor space

$$\overline{V}^{\|\cdot\|} = \overline{V_1 \otimes \ldots \otimes V_d}^{\|\cdot\|}.$$

If $(\cdot,\cdot)_\nu$ is the inner product in V_ν, a canonical inner product on V can be first defined for elementary tensors

$$(v^{(1)} \otimes \ldots \otimes v^{(d)}, w^{(1)} \otimes \ldots \otimes w^{(d)}) = (v^{(1)}, w^{(1)})_1 \ldots (v^{(d)}, w^{(d)})_d$$

and then extended by linearity to the whole space V. As usual, the associated norm $\|\cdot\|$ is called the canonical norm.

Example 3.1 (L^p spaces & Sobolev spaces)

(a) Let $1 \le p < \infty$. If $V_\nu = L^p_{\mu_\nu}(\mathcal{X}_\nu)$, then with $\mu = \mu_1 \otimes \ldots \otimes \mu_d$

$$L^p_{\mu_1}(\mathcal{X}_1) \otimes \ldots \otimes L^p_{\mu_d}(\mathcal{X}_d) \subset L^p_\mu(\mathcal{X}_1 \times \ldots \times \mathcal{X}_d) \quad \text{and}$$
$$\overline{L^p_{\mu_1}(\mathcal{X}_1) \otimes \ldots \otimes L^p_{\mu_d}(\mathcal{X}_d)}^{\|\cdot\|} = L^p_\mu(\mathcal{X}_1 \times \ldots \times \mathcal{X}_d)$$

where $\|\cdot\|$ is the natural norm on $L^p_\mu(\mathcal{X}_1 \times \ldots \times \mathcal{X}_d)$.

(b) The Sobolev spaces

$$H^k(\mathcal{X}) = \overline{H^k(\mathcal{X}_1) \otimes \ldots \otimes H^k(\mathcal{X}_d)}^{\|\cdot\|_{H^k}} \quad \text{and}$$
$$H^k_{\text{mix}}(\mathcal{X}) = \overline{H^k(\mathcal{X}_1) \otimes \ldots \otimes H^k(\mathcal{X}_d)}^{\|\cdot\|_{H^k_{\text{mix}}}}$$

of functions defined on $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$, equipped with the norms

$$\|u\|^2_{H^k} = \sum_{|\alpha| \le k} \|\partial^\alpha u\|^2_{L^2} \quad \text{and} \quad \|u\|^2_{H^k_{\text{mix}}} = \sum_{|\alpha|_\infty \le k} \|\partial^\alpha u\|^2_{L^2},$$

respectively, are **two different** tensor Hilbert spaces (see Section III.3).

Tensor product basis

If $\{\psi^{(\nu)}_i\}_{i \in I_\nu}$ is a basis of V_ν, then a basis of $V = V_1 \otimes \ldots \otimes V_d$ is given by

$$\left\{ \psi_i = \psi^{(1)}_{i_1} \otimes \ldots \otimes \psi^{(d)}_{i_d} : i \in I = I_1 \times \ldots \times I_d \right\}.$$

A tensor $v \in V$ admits a decomposition

$$v = \sum_{i \in I} a_i \psi_i = \sum_{i_1 \in I_1} \ldots \sum_{i_d \in I_d} a_{i_1,\ldots,i_d} \psi^{(1)}_{i_1} \otimes \ldots \otimes \psi^{(d)}_{i_d},$$

and v can be identified with the (algebraic) tensor of its coefficients $a \in \mathbb{R}^I$.

If $\{\psi^{(\nu)}_i\}_{i \in I_\nu}$ is an orthonormal basis of V_ν, then $\{\psi_i\}_{i \in I}$ is an orthonormal basis of $\overline{V}^{\|\cdot\|}$ and

$$\|v\|^2 = \sum_{i \in I} a_i^2 := \|a\|^2.$$

The map $\Psi : a \mapsto \sum_{i \in I} a_i \psi_i$ defines a linear isometry from $\mathbb{R}^I$ to V for finite dimensional spaces and from $\ell_2(I)$ to $\overline{V}^{\|\cdot\|}$ for infinite dimensional spaces.

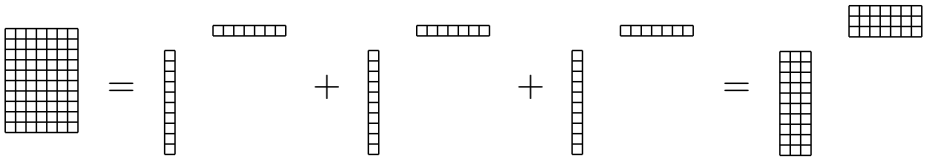
Tensor ranks (order-two tensors)

The rank of an order-two tensor $u \in V \otimes W$, denoted $\mathrm{rank}(u)$, is the minimal integer r s.t.

$$u = \sum_{k=1}^{r} v_k \otimes w_k, \quad \text{for some } \; v_k \in V, \; w_k \in W.$$

If $V = \mathbb{R}^n$ and $W = \mathbb{R}^m$ it can be identified with a matrix $u \in \mathbb{R}^{n \times m}$ and $\mathrm{rank}(u)$ coincides with the matrix rank, i.e.

$$u = \sum_{k=1}^{r} v_k w_k^T = V W^T, \quad \text{where} \quad V \in \mathbb{R}^{n \times r}, \quad W \in \mathbb{R}^{m \times r}.$$



The set of tensors in $V \otimes W$ with rank bounded by r, denoted

$$\mathcal{R}_r = \{v : \mathrm{rank}(v) \le r\},$$

is neither a linear space nor a convex set. However, best approximation in $\mathcal{R}_r$ is well posed and it has many favorable properties for numerical treatment.

Canonical rank of higher-order tensors & canonical format

For tensors $u \in V_1 \otimes \ldots \otimes V_d$ with $d \ge 3$, there are different notions of rank.

The canonical rank, which is the natural extension of the notion of rank for order-two tensors, is the minimal integer r such that

$$u(x_1, \ldots, x_d) = \sum_{k=1}^{r} v_k^{(1)}(x_1) \ldots v_k^{(d)}(x_d), \quad \text{for some } \; v_k^{(\nu)} \in V_\nu \, .$$

The subset of $V = V_1 \otimes \ldots \otimes V_d$ with canonical rank bounded by r is denoted

$$\mathcal{R}_r = \{v \in V : \mathrm{rank}(v) \le r\}. \tag{3.2}$$

If $\dim(V_\nu) \le n$, the storage complexity of tensors in $\mathcal{R}_r$ is

$$\mathrm{storage}(\mathcal{R}_r) = r \sum_{\nu=1}^{d} \dim(V_\nu) \le rdn \, .$$

Unfortunately, for $d \ge 3$, the set $\mathcal{R}_r$ loses many of the favourable properties.

Drawbacks of canonical format

- Determining the rank of a given tensor is a NP-hard problem.
- The set $\mathcal{R}_r$ is not an algebraic variety or a manifold.
- No notion of singular value decomposition.
- The map $v \mapsto \operatorname{rank}(v)$ is not lower semi-continuous and so $\mathcal{R}_r$ is not closed.

Example 3.2

Consider the order-3 tensor

$$v = a \otimes a \otimes b + a \otimes b \otimes a + b \otimes a \otimes a$$

where a and b are linearly independent vectors in $\mathbb{R}^m$. The rank of v is 3. The sequence of rank-2 tensors

$$v_n = n(a + \frac{1}{n}b) \otimes (a + \frac{1}{n}b) \otimes (a + \frac{1}{n}b) - na \otimes a \otimes a$$

converges to v as $n \to \infty$.

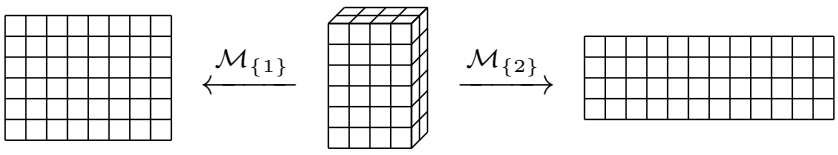
- As a consequence, for most problems, there is no robust method for approximation in canonical format $\mathcal{R}_r$.

α-rank (tensors)

For a non-empty subset α of $D = \{1, \dots, d\}$, a tensor $u \in V = V_1 \otimes \dots \otimes V_d$ can be identified with an order-two tensor

$$\mathcal{M}_\alpha(u) \in V_\alpha \otimes V_{\alpha^c}\,, \tag{3.3}$$

where $V_\alpha = \bigotimes_{\nu \in \alpha} V_\nu$, and $\alpha^c = D \setminus \alpha$. The operator $\mathcal{M}_\alpha = V \to V_\alpha \otimes V_{\alpha^c}$ is called the matricisation operator.



The α-rank of u, denoted $\operatorname{rank}_\alpha(u)$, is the rank of the order-two tensor $\mathcal{M}_\alpha(u)$,

$$\operatorname{rank}_\alpha(u) = \operatorname{rank}(\mathcal{M}_\alpha(u)),$$

which is the minimal integer r_α such that

$$\mathcal{M}_\alpha(u) = \sum_{k=1}^{r_\alpha} v_k^\alpha \otimes w_k^{\alpha^c}\,, \quad \text{for some } \; v_k^\alpha \in V_\alpha,\; w_k^{\alpha^c} \in V_{\alpha^c}.$$

Note that $\operatorname{rank}_\alpha(u) = \operatorname{rank}_{\alpha^c}(u)$.

α-rank (functions)

A multivariate function $u(x_1, \dots, x_d)$ with $\operatorname{rank}_\alpha(u) \le r_\alpha$ is such that

$$u(x) = \sum_{k=1}^{r_\alpha} v_k^\alpha(x_\alpha) w_k^{\alpha^c}(x_{\alpha^c})$$

for some functions $v_k^\alpha(x_\alpha)$ and $w_k^{\alpha^c}(x_{\alpha^c})$ of $x_\alpha = \{x_\nu\}_{\nu\in\alpha}$ and $x_{\alpha^c} = \{x_\nu\}_{\nu\in\alpha^c}$.

Example 3.3

(a) $u(x) = u^1(x_1)\dots u^d(x_d)$ can be written $u(x) = u^\alpha(x_\alpha)u^{\alpha^c}(x_{\alpha^c})$, with $u^\alpha(x_\alpha) = \prod_{\nu\in\alpha} u^\nu(x_\nu)$. Therefore, for any α, $\operatorname{rank}_\alpha(u) = 1$.

(b) $u(x) = u^1(x_1) + \dots + u^d(x_d)$ can be written $u(x) = u^\alpha(x_\alpha)\cdot 1 + 1\cdot u^{\alpha^c}(x_{\alpha^c})$, with $u^\alpha(x_\alpha) = \sum_{\nu\in\alpha} u^\nu(x_\nu)$. Therefore, $\operatorname{rank}_\alpha(u) \le 2$.

(c) $u(x) = \sum_{k=1}^r u_k^1(x_1)\dots u_k^d(x_d)$ can be written $\sum_{k=1}^r u_k^\alpha(x_\alpha)u_k^{\alpha^c}(x_{\alpha^c})$ with $u_k^\alpha(x_\alpha) = \prod_{\nu\in\alpha} u_k^\nu(x_\nu)$.

Hence, $\operatorname{rank}_\alpha(u) \le \operatorname{rank}(u) = r$, for any α, with equality if the functions $\{u_k^\alpha(x_\alpha) : 1 \le k \le r\}$ and $\{u_k^{\alpha^c}(x_{\alpha^c}) : 1 \le k \le r\}$ are linearly independent.

Singular value decomposition of order-two tensors

When V and W are Hilbert spaces, an algebraic tensor $u \in V \otimes W$ admits a singular value decomposition (SVD)

$$u = \sum_{k\ge 1} \sigma_k v_k \otimes w_k, \tag{3.4}$$

where $v_k \in V$ and $w_k \in W$ are orthonormal vectors (singular vectors) and $\sigma_k \in \mathbb{R}^+$ are the singular values. The rank of u coincides with the number of non-zero singular values:

$$\operatorname{rank}(u) = \#\{k : \sigma_k \neq 0\}.$$

For $V = \mathbb{R}^n$ and $W = \mathbb{R}^m$, u is identified with a matrix in $\mathbb{R}^{n\times m}$ and

$$u = \sum_{k=1}^{\operatorname{rank}(u)} \sigma_k v_k w_k^T = \mathbf{V}\mathbf{\Sigma}\mathbf{W}^T. \tag{3.5}$$

with orthogonal matrices $\mathbf{V} \in \mathbb{R}^{n\times n}, \mathbf{W} \in \mathbb{R}^{m\times m}$ and diagonal matrix $\mathbf{\Sigma} \in \mathbb{R}^{n\times m}$.

(A tensor $u \in \overline{V \otimes W}^{\|\cdot\|_\vee}$ still admits a SVD of the form (3.4). The rank is possibly infinite.)

Best approximation of order-two tensors via truncated SVD

With the singular values $\{\sigma_k\}_{k\geq 1}$ sorted by decreasing order, an element of best approximation of u in the set of tensors with rank bounded by r is provided by the truncated singular value decomposition

$$u_r = \sum_{k=1}^{r} \sigma_k v_k \otimes w_k, \tag{3.6}$$

with an error

$$\|u - u_r\|^2 = \min_{\mathrm{rank}(v)\leq r} \|u - v\|^2 = \sum_{k\geq r+1} \sigma_k^2. \tag{3.7}$$

An approximation u_r with relative precision ϵ, such that $\|u - u_r\| \leq \epsilon\|u\|$, can be obtained by choosing a rank r such that

$$\sum_{k\geq r+1} \sigma_k^2 \leq \epsilon^2 \sum_{k\geq 1} \sigma_k^2. \tag{3.8}$$

The complexity of computing the SVD is $O(n^3)$ if $\dim(V) = \dim(W) = n$. For u given in low-rank format $u = \sum_{k=1}^{r} a_k \otimes b_k$, with a rank $r < n$, the complexity reduces to $O(r^3 + 2rn^2)$.

Higher-order singular value decomposition

For a non-empty $\alpha \subset D = \{1, \ldots, d\}$, a tensor $u \in V_1 \otimes \ldots \otimes V_d$ can be identified with its matricisation

$$\mathcal{M}_\alpha(u) \in V_\alpha \otimes V_{\alpha^c},$$

an order-two tensor which admits a singular value decomposition

$$\mathcal{M}_\alpha(u) = \sum_{k\geq 1} \sigma_k^\alpha v_k^\alpha \otimes w_k^{\alpha^c}.$$

The set $\sigma^\alpha(u) := \{\sigma_k^\alpha\}_{k\geq 1}$ is called the set of α-singular values of u. The α-rank of u is the number of non-zero α-singular values

$$\mathrm{rank}_\alpha(u) = \#\{k : \sigma_k^\alpha \neq 0\}.$$

Truncated higher-order singular value decomposition

By sorting the α-singular values by decreasing order, an approximation u_r with α-rank r can be obtained by retaining the r largest singular values, i.e.

$$u_r \quad \text{such that} \quad \mathcal{M}_\alpha(u_r) = \sum_{k=1}^{r} \sigma_k^\alpha v_k^\alpha \otimes w_k^{\alpha^c},$$

which satisfies

$$\|u - u_r\|^2 = \min_{\operatorname{rank}_\alpha(v) \le r} \|u - v\|^2 = \sum_{k>r} (\sigma_k^\alpha)^2.$$

But, there are 2^{d-1} different binary partitions $\alpha \cup \alpha^c$ of D, each with corresponding SVD and a way to truncate a higher-order tensor!

For tree-based tensor formats

$$\mathcal{T}_r^T = \{v : \operatorname{rank}_\alpha(v) \le r_\alpha, \alpha \in T\},$$

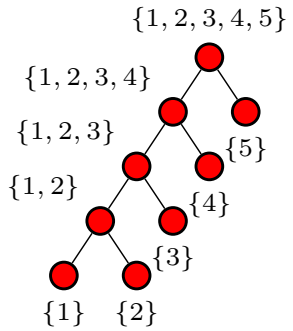
where T is a dimension partition tree over $D = \{1, \dots, d\}$, a higher order singular value decomposition (HOSVD) (also called hierarchical SVD) can also be defined from SVDs of matricisations $\mathcal{M}_\alpha(u)$ of u.

Will now consider the tensor train format.

4. Tensor Train Decomposition

Linear binary trees – Tensor Train (TT) Tucker format

Arguably the simplest tree-based tensor format $\mathcal{T}_r^T \subset \mathbb{R}^I$ with $I = I_1 \times \ldots \times I_d$ is the Tensor Train Tucker format based on a linear binary tree (short TT format):

$T =$ 

It is equivalent to matrix product states [White, 1992] in quantum physics, since each element of a tensor $\mathbf{A} \in \mathcal{T}_r^T$ can be factorised into a product of matrices:

$$A(i_1, \ldots, i_d) = G_1(i_1) G_2(i_2) \ldots G_d(i_d) \quad \text{with} \quad G_\nu(i_\nu) \in \mathbb{R}^{r_{\nu-1} \times r_\nu}, \tag{4.1}$$

where $r_0 = r_d = 1$ and $i = (i_1, \ldots, i_d) \in I$.

Each $\mathbf{G}_\nu$ is in fact a tensor of order 3 in $\mathbb{R}^{r_{\nu-1} \times n_\nu \times r_\nu}$ where $n_\nu = \dim(I_\nu)$, such that the decomposition in index form becomes

$$A(i_1, \ldots, i_d) = \sum_{k_1=1}^{r_1} \ldots \sum_{k_{d-1}=1}^{r_{d-1}} G_1(1, i_1, k_1) G_2(k_1, i_2, k_2) \ldots G_d(k_{d-1}, i_d, 1). \tag{4.2}$$

TT decomposition and α-ranks [Oseledets, SISC, 2011]

For any $1 \le \nu \le d-1$, let $\alpha = \{1, \ldots, \nu\} \subset T$. Since (4.2) implies

$$\mathbf{A} = \sum_{k_\nu=1}^{r_\nu} \mathbf{A}_{k_\nu}^{\alpha} \otimes \mathbf{A}_{k_\nu}^{\alpha^c} \quad \text{with} \quad A_{k_\nu}^{\alpha} \in \mathbb{R}^{I_1 \times \ldots \times I_\nu}, \; A_{k_\nu}^{\alpha^c} \in \mathbb{R}^{I_{\nu+1} \times \ldots \times I_d},$$

it follows that

$$r_\nu \;\ge\; \operatorname{rank}_\alpha(\mathbf{A}) \;= \operatorname{rank}(\mathcal{M}_\alpha(\mathbf{A})).$$

Moreover, these ranks are achievable, providing a constructive way to compute the TT-decomposition.

Theorem 4.1

Let $\mathbf{A} \in \mathbb{R}^I$ and suppose that for all $1 \le \nu \le d-1$ and $\alpha = \{1, \ldots, \nu\}$,

$$\operatorname{rank}_\alpha(\mathbf{A}) = r_\nu.$$

Then there exists a TT-decomposition (4.2) of $\mathbf{A}$ with TT-ranks less than or equal to r_ν.

Proof. Demonstrated on the iPad. □

Approximate TT decomposition and error bound

In practical computations, the matricisations $\mathcal{M}_\alpha(\mathbf{A})$ are rarely going to be of low rank (exactly). However, if $\{\sigma_k^\alpha\}_{k\geq 1}$ are the α-singular values of $\mathbf{A}$ (sorted by decreasing order) and, for $\alpha = \{1,\ldots,\nu\}$,

$$\varepsilon_\nu^2 := \sum_{k>r_\nu} (\sigma_k^\alpha)^2 \,,$$

Then (using HOSVD) there exist two matrices A_ν, E_ν, such that

$$\mathcal{M}_\alpha(\mathbf{A}) = A_\nu + E_\nu \quad \text{with} \quad \operatorname{rank}(A_\nu) = r_\nu \quad \text{and} \quad \|E_\nu\| = \varepsilon_\nu \,. \tag{4.3}$$

Theorem 4.2

Let $\mathbf{A} \in \mathbb{R}^I$. Then there exists a tensor $\mathbf{B}$ in TT-format (4.2) with TT-ranks r_ν and

$$\|\mathbf{A}-\mathbf{B}\| \;\leq\; \sqrt{\sum_{\nu=1}^{d-1} \varepsilon_\nu^2} \,. \tag{4.4}$$

Proof. Demonstrated on the iPad. □

TT-SVD algorithm

The proof of Theorems 4.1 and 4.2 leads to the following practial algorithm:

TT-SVD Algorithm

Input. d-dimensional tensor $\mathbf{A}$, tolerance ε.

Output. TT cores $G_1,\ldots,G_d$ of TT approximation $\mathbf{B}$, with r_ν such that (4.3) holds with $\varepsilon_\nu = \frac{\varepsilon}{\sqrt{d-1}}\|\mathbf{A}\|$, in order to guarantee $\|\mathbf{A}-\mathbf{B}\| \leq \varepsilon\|\mathbf{A}\|$.

1: Set $\mathbf{C} = \mathbf{A}$, $r_0 = 1$.
2: **for** $\nu = 1,\ldots,d-1$ **do**
3: $\quad C = \mathcal{M}_\alpha(\mathbf{C})$ where $\alpha = \{k_{\nu-1}, \nu\}$ and $\alpha^c = \{\nu+1,\ldots,d\}$.
4: $\quad$ Compute truncated SVD $C = V\Sigma W + E$ s.t. $\operatorname{rank}(V\Sigma W) = r_\nu$, $\|E\| \leq \varepsilon_\nu$.
5: $\quad \mathbf{G}_\nu := \text{tensor}(V) \in \mathbb{R}^{r_{\nu-1}\times n_\nu \times r_\nu}$.
6: $\quad \mathbf{C} = \text{tensor}(\Sigma W) \in \mathbb{R}^{r_\nu} \otimes \mathbb{R}^{I_{\nu+1}} \otimes \ldots \otimes \mathbb{R}^{I_d}$.
7: **end for**
8: $\mathbf{G}_d := \mathbf{C}$.

Here, $C = \mathcal{M}_\alpha(\mathbf{C})$ denotes matricisation w.r.t. a set α and its complement α^c (as above) and $\mathbf{C} = \text{tensor}(C)$ is the reverse operation from matricised form back to tensor form.

Quasi-best approximation

Corollary 4.3

Let T be a linear binary tree on $\{1,\dots,d\}$, $r=(r_\nu)_{\nu=1}^{d-1}\subset\mathbb{N}^{d-1}$ and let $\mathbf{A}\in\mathbb{R}^I$ be arbitrary. Then the best approximation $\mathbf{A}^{best}\in\mathcal{T}_r^T$ to $\mathbf{A}$ in the Frobenius norm exists and the TT-approximation $\mathbf{B}$ computed by the TT-SVD algorithm is quasi-optimal:

$$\|\mathbf{A}-\mathbf{B}\| \;\le\; \sqrt{d-1}\,\|\mathbf{A}-\mathbf{A}^{best}\|$$

Proof. Let $\varepsilon := \inf_{\mathbf{C}\in\mathcal{T}_r^T}\|\mathbf{A}-\mathbf{C}\|$. For the proof that this infimum is in fact attained in $\mathcal{T}_r^T$, see [Oseledets, 2011, Cor. 2.4]. Let $\mathbf{B}^{\min}\in\mathcal{T}_r^T$ be this minimum.

(It follows from the fact that for any tensor $\mathbf{B}\in\mathcal{T}_r^T$ there exists a one-to-one correspondence with its matricisations $\mathcal{M}_\alpha(\mathbf{B})$ and the set of matrices of rank less than or equal to r_ν is closed.)

Since $\|\mathbf{A}-\mathbf{B}^{\min}\|=\varepsilon$, each matricisation of $\mathbf{A}$ can be approximated s.t. (4.3) holds with $\varepsilon_\nu\le\varepsilon$ and the quasi-optimality result follows from Theorem 4.2. □

Complexity considerations, rounding & basic operations

- In the basic TT format (4.2), the storage complexity, i.e. the number of parameters, is $(d-2)nr^2+2nr$. However, by using an auxiliary Tucker decomposition of the core tensors $\mathbf{G}_\nu$ this can be reduced to
$$\mathcal{O}(dnr+(d-2)r^3).$$
- Many basic linear algebra operations with TT tensors yield results also in TT format, but with increased ranks. Therefore a crucial operation for TT tensors is that of rounding (also called truncation or recompression),
i.e. given a tensor $\mathbf{A}\in\mathcal{T}_r^T$, to estimate $r'_\nu\le r_\nu$ such that $\mathbf{A}$ can be approximated in $\mathcal{T}_{r'}^T$ maintaining the prescribed tolerance ε.
Rounding can be carried out with computational complexity
$$\mathcal{O}(dnr^2+dr^4).$$
- Basic operations with TT tensors, such as addition, multidimensional contraction (e.g. for integration), elementwise (Hadamard) product, scalar product or Frobenius norm, can also all be computed with a complexity of
$$\mathcal{O}(dnr^2+dr^4).$$

For details on those points see [Oseledets, 2011, Sect. 3-4].

5. TT Cross Approximation

Curse of Dimensionality for TT-SVD

- For large $d \gg 2$, the TT-SVD algorithm is of course not practical !
  It requires access to all $\prod_{\nu=1}^{d} n_\nu$ entries of the tensor.

- For simplicity, let $n_\nu = n \in \mathbb{N}\backslash\{1\}$ for all ν. Then #entries of $\mathbf{A} \in \mathbb{R}^I$ is n^d !
  (Case $n_\nu = 1$ can be ignored, since then the tensor reduces to a lower-dimensional tensor.)

  Curse of Dimensionality!

- Also the computational complexity of TT-SVD is $\mathcal{O}(n^{d+1} + r^2 n^d)$!
  (for a single SVD of an $M \times N$ matrix with $M \leq N$ the cost is $\mathcal{O}(M^2 N)$)

- We need a significantly more efficient approximate method for finding good low-rank approximations of matrices to break the curse of dimensionality:

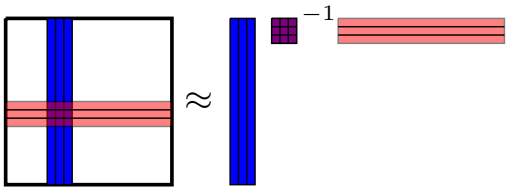
  Cross (or skeleton) approximation and the *maxvol* algorithm
  [Bebendorf, 2000], [Goreinov, Tyrtyshnikov, 2001]

Cross approximation methods for matrices ($d = 2$)

- Let $U \in \mathbb{R}^{m \times n}$. Recall SVD is best approximation $\|U - VW^T\|_F^2$ among all matrices $V \in \mathbb{R}^{m \times r}$, $W \in \mathbb{R}^{n \times r}$.
- Use interpolation instead: Choose V, W s.t.

$$U(\mathcal{I}, :) = V(\mathcal{I}, :)W^T, \qquad U(:, \mathcal{J}) = VW^T(:, \mathcal{J})$$

  for some index sets $\mathcal{I}, \mathcal{J} \subset \{1, \ldots, n\}$ with $|\mathcal{I}| = |\mathcal{J}| = r$.
- Equivalent to cross decomposition (i.e. truncated, pivoted LU factorisation):



$$U \approx \widetilde{U} := U(:, \mathcal{J})U^{-1}(\mathcal{I}, \mathcal{J})U(\mathcal{I}, :).$$

How to *find* index sets $\mathcal{I}, \mathcal{J}$?

Maximum Volume principle [Tyrtyshnikov, '00], [Goreinov, Tyrtyshnikov, '01]

(the modulus of the determinant of a square matrix is referred to as its *volume*)

- Best indices: If $|\det U(\mathcal{I}, \mathcal{J})| = \max_{\hat{\mathcal{I}}, \hat{\mathcal{J}}} \left|\det U(\hat{\mathcal{I}}, \hat{\mathcal{J}})\right|$ then

$$\|U - \widetilde{U}\|_C \leq (r+1) \min_{\mathrm{rank}(V)=r} \|U - V\|_2$$

  (where $\|A\|_C := \max_{i,j} |A_{ij}|$ is the Chebyshev norm and $\|A\|_2$ is the spectral norm).
- But: NP-hard to look through all submatrices.
- **However**, if $|\det U(\mathcal{I}, \mathcal{J})| = \eta \max_{\hat{\mathcal{I}}, \hat{\mathcal{J}}} \left|\det U(\hat{\mathcal{I}}, \hat{\mathcal{J}})\right|$, for some $\eta > 0$, then also

$$\|U - \widetilde{U}\|_C \leq \eta^{-1}(r+1) \min_{\mathrm{rank}(V)=r} \|U - V\|_2.$$

- If optimal $\mathcal{J}$ known a priori, maxvol algorithm [Goreinov et al, '10] provides optimal index set $\mathcal{I}$ in $n \times r$ submatrix in $\mathcal{O}(nr(r+p))$ complexity, where p is number of iterations in the maxvol algorithm. Similar to row pivoting in LU.
- In practice, $\mathcal{J}$ not known, but volume non-decreasing when iterating between row and column sets $\mathcal{I}$ and $\mathcal{J}$ [Tyrtyshnikov, '00] ...

Cross approximation via alternating iteration ($d = 2$)

[Bebendorf, '00], [Tyrtyshnikov, '00]

Practically realizable strategy (with $\mathcal{O}(2nr)$ samples & $\mathcal{O}(nr^2)$ flops):

Assume initial set $\mathcal{J} \subset \{1 \dots n\}$ is given (e.g. random) $\rightarrow$ $V = U(:, \mathcal{J})$

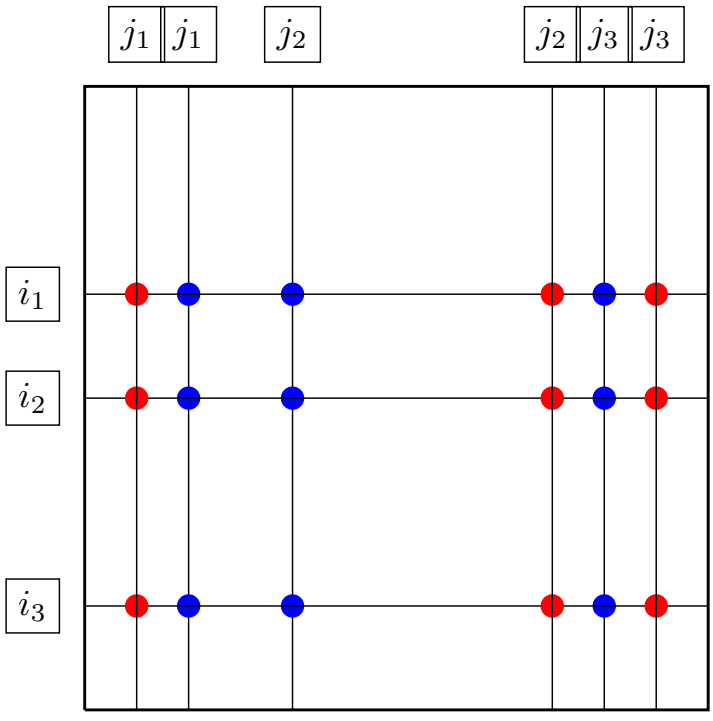
1. $\mathcal{I} = \texttt{pivots}\,(V) \quad \rightarrow \quad W = U^{-1}(\mathcal{I}, \mathcal{J})U(\mathcal{I}, :)$
2. $\mathcal{J} = \texttt{pivots}\,(W) \quad \rightarrow \quad V = U(:, \mathcal{J})$
3. repeat...

Get $U \approx VW^T$.

`pivots` feasible via pivoted LU [Bebendorf, '00] or maxvol [Tyrtyshnikov '00].

- For numerical stability better to use (rank-revealing) QR $\longrightarrow$ allows rank reduction
- To avoid underestimating rank, fix 'search' rank slightly larger and add random columns after QR-factorisation.
- Heuristic algorithm, but converges very fast in most important cases.

Cross approximation via alternating iteration ($d = 2$)



Extension to $d > 2$: The TT Cross algorithm

[Oseledets, Tyrtyshnikov, '10]

Given initial sets $\mathcal{J}_k$, $0 < k < d$, let $k = 1$ and iterate:
(for notational convenience set $\mathcal{I}_0 = \mathcal{J}_d = \emptyset$)

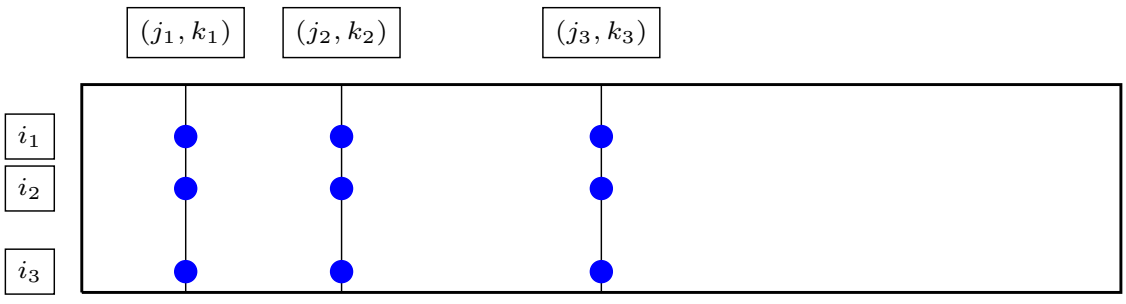
1. $\widetilde{G}_k(i_k) = U(\mathcal{I}_{k-1}, i_k, \mathcal{J}_k)$. {Update a block}
2. $\mathcal{I}_k = \texttt{pivots}_{\mathbf{row}}(\widetilde{G}_k)$ or $\mathcal{J}_{k-1} = \texttt{pivots}_{\mathbf{col}}(\widetilde{G}_k)$. {Update sets}
3. Move to the next block (set $k = k+1$ or $k = k-1$), switching direction if $k = d$ or $k = 1$ is reached.

Using different *matrizations* of the tensors in each step.

$\mathcal{O}(dnr^2)$ samples & $\mathcal{O}(dnr^3)$ flops per iteration **linear** in d and n

- Creates left- and right-nested sequences of index sets $\{\mathcal{I}_k\}$ and $\{\mathcal{J}_k\}$, resp.
- Explores the *entire range* in each variable in $\mathcal{O}(r^2)$ "fibres", defined by the index sets $\{\mathcal{I}_k\}$ and $\{\mathcal{J}_k\}$.
- Theory of quasi-best approximation for matrix cross can be extended to TT cross, e.g. [Savostyanov, 2014]

TT Cross – An Efficient Computation of a TT Decomposition



6. ALS-Cross: TT Surrogates for Parametric PDEs

Recall: Stochastic Collocation Method (Section IV.5 or VI.2)

Stochastic PDE example in parametric form in $D \times \Gamma \subset \mathbb{R}^{d_x \times d}$:

$$-\nabla \cdot \Big(a(x,y) \nabla u(x,y) \Big) = f(x,y), \quad (x,y) \in D \times \Gamma \text{ and } u|_{\partial D} \equiv 0 \tag{6.1}$$

where $y = (y_1, \ldots, y_d) \in \Gamma = \Gamma_1 \times \cdots \times \Gamma_d$ with Γ_j **bounded** (Assumption)

- Use sampling points $\{y^{(i)}\}_{i=1,\ldots,N}$ in Γ and
- FE solutions $u_h(x, y^{(i)}) \in V_h \subset V$ (w.r.t. mesh $\mathcal{T}_h$)
- to construct the (stochastic collocation) **interpolant**

$$u_{N,h}^{(\mathrm{SC})}(x,y) = \mathcal{I}_N[u_h](x,y) = \sum_{i=1}^{N} u_h(x, y^{(i)}) \phi_i(y) \tag{6.2}$$

in the polynomial space $\mathcal{P}_N = \mathrm{span}\{\phi_i\}_{i=1}^N \subset L^2_\rho(\Gamma)$ such that
(basis made up of products of basis functions for univariate polynomials)

$$u_{N,h}^{(\mathrm{SC})}(x, y^{(i)}) = u_h(x, y^{(i)}), \quad \text{for } i = 1, \ldots, N \qquad \text{(interpolating condition)}$$

- Functionals $Q_\ell = \psi(u_\ell)$ and integrals $\mathbb{E}[Q_\ell]$ approximated by applying the functional to the interpolant $u_{N,h}^{(\mathrm{SC})}$ and integrating result (repeated 1D integrals)

Block-diagonal structure of collocation system (first for $d = 1$)

- For each sampling point $y^{(i)}$ solve

$$\int_D a(x, y^{(i)}) \nabla \psi_j(x) \cdot \nabla u_h(x, y^{(i)}) \, dx = \int_D \psi_j(x) f(x, y^{(i)})) dx$$

Important: *independent* equations over x for different $y^{(i)}$:

$$\begin{bmatrix} A^{(1)} & & & \\ & A^{(2)} & & \\ & & \ddots & \\ & & & A^{(n)} \end{bmatrix} \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(n)} \end{bmatrix} = \begin{bmatrix} f^{(1)} \\ f^{(2)} \\ \vdots \\ f^{(n)} \end{bmatrix}$$

but every block $A^{(i)}$ is a FE system and not diagonal.

Strategy:

- Approximate and solve system in TT format without ruining block structure.
- Approximate $a(x, y)$, A, f and u in TT format.
- Apply variant of alternating least squares (ALS) to linear system in TT format [Holtz, Rohwedder, Schneider, '12]

Cross interpolation for stochastic collocation system via ALS

[Dolgov, RS, SIAM JUQ, 2019]

- Main idea: Apply cross algorithm directly to the entire PDE solution (rather than scalar entries).
- Rewrite interpolation as projection:

$$E_{\mathcal{J}}^{\top} \cdot I \cdot \mathrm{vec}(VW^T) = E_{\mathcal{J}}^{\top} \cdot I \cdot \mathrm{vec}(U),$$

  where $E_{\mathcal{J}}$ is a rectangular *submatrix of identity* on the index set $\mathcal{J}$.

- Replace I by the stiffness matrix A:

$$E_{\mathcal{J}}^{\top} \cdot A \cdot \mathrm{vec}(VW^T) = E_{\mathcal{J}}^{\top} \mathrm{vec}(F)$$

  . . . and $A \cdot \mathrm{vec}(U)$ by the right hand side $\mathrm{vec}(F)$.

- Then find V,W implicitly via alternating least squares (ALS)

Represent PDE coefficient and matrix in TT format

- *Coefficient* is low-rank (approximable):
  (clear for uniform coefficients, but also for lognormal)

$$a(x,y) \approx \sum_{\beta=1}^{R} g_\beta(x) h_\beta(y).$$

- Hence A is low-*Kronecker*-rank:

$$A = \sum_{\beta=1}^{R} A_\beta \otimes D_\beta$$

  and the matrix-vector product becomes

$$\mathbf{A} \cdot \mathrm{vec}(VW^\top) = \mathrm{vec}\left[\sum_\beta (A_\beta V)\,(D_\beta W)^\top\right]$$

- A_β is a FEM matrix, but $D_\beta = \mathrm{diag}(d_\beta)$ is diagonal.

Interpolatory representation

- VW^T is not unique $\rightarrow$ ensure $W(\mathcal{J}) = I$ (by choosing $U^{-1}(\mathcal{I},\mathcal{J})U(\mathcal{I},:)$)

  ... as a by-product $V = U(:,\mathcal{J})$ (as above in the TT Cross).

- *Distribute* the products

$$E_{\mathcal{J}}^\top \cdot A \cdot \mathrm{vec}(VW^T) = \sum_{\beta=1}^{R} A_\beta V \otimes [D_\beta(\mathcal{J},:)W] = \sum_{\beta=1}^{R} A_\beta U(:,\mathcal{J}) \otimes \mathrm{diag}(d_\beta(\mathcal{J}))$$

- Now as in the matrix cross algorithm:
  - start with initial parameter set $\mathcal{J}$ and solve r decoupled FE problems;
  - project the linear system onto the new subspace spanned by $V \times I$;
  - solve the resulting (block-diagonal) Galerkin system for W and apply *maxvol* to find a new pivot set $\mathcal{J}$;
  - repeat until convergence...

Iteration for $d = 1$, i.e. one parameter (in a nutshell)

Stage 1 (Space):

$$\{j_1, \ldots, j_r\} = \texttt{pivots}\,(W)$$

$$\begin{bmatrix} A_{j_1} & & \\ & A_{j_2} & \\ & & A_{j_r} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_r \end{bmatrix} = \begin{bmatrix} f_{j_1} \\ f_{j_2} \\ f_{j_r} \end{bmatrix}$$

Solve r independent deterministic problems.

- Similar to MC or stochastic collocation (can use specialised PDE solvers)

Stage 2 (Parameter): (A_j is not diagonal $\rightarrow$ ALS-projection)

1. Make V orthogonal $\rightarrow$ projection matrix $\mathcal{V} = V \otimes I$.
2. Solve $\left(\mathcal{V}^\top \mathbf{A} \mathcal{V}\right) w = \mathcal{V}^\top \mathbf{f}$ $\quad\rightarrow\quad$ reduces to n systems of size $r \times r$!
3. Use maxvol to find new pivot set $\{j_1, \ldots, j_r\} = \texttt{pivots}\,(W)$
4. Repeat ...

Similar to the Reduced Basis Method (in 2D), but with more sophisticated search via maxvol algorithm **and ...**

Tensor Train for stochastic (or parametric) PDEs

Idea **extensible** to many dimensions: $-\nabla a(x, y_1, \ldots, y_d)\nabla u = f$

$$\mathbf{A} = \sum_{(\beta_0,\ldots,\beta_d)=1}^{(r_0,\ldots,r_d)} A^{(0)}_{\beta_0} \otimes \mathrm{diag}(d^{(1)}_{\beta_0,\beta_1}) \otimes \cdots \otimes \mathrm{diag}(d^{(d)}_{\beta_{d-1}})$$

even for $d \gg 1$! Find solution $\mathbf{u}$ in TT format by alternating iteration:

$$\underbrace{\mathbf{u}_1 \cdots \mathbf{u}_{k-1}}_{U_{<k}} \cdot \mathbf{u}_k \cdot \underbrace{\mathbf{u}_{k+1} \cdots \mathbf{u}_d}_{U_{>k}}$$

The algorithm in a nutshell:

1. $\mathcal{J}_k = \texttt{pivots}_{\mathbf{col}}\,(U_{>k})$.
2. $\mathcal{U}_{<k} = U_{<k} \otimes I$.
3. Generate and solve $\left[\mathcal{U}_{<k}{}^\top \mathbf{A}_{\mathcal{J}_k} \mathcal{U}_{<k}\right] \mathbf{u}_k = \mathcal{U}_{<k}^\top \mathbf{f}_{\mathcal{J}_k}$.
   (for $k = 0$: r_0 independent PDE solves; for $k > 0$: nr dense $r \times r$ systems)
4. $\mathcal{I}_k = \texttt{pivots}_{\mathbf{row}}\,(U_{<k+1})$.
5. Set $k = k + 1$ or $k = k - 1$ and repeat...

Hybrid ALS-Cross algorithm

[Dolgov, RS, SIAM JUQ, 2019]

The main step is solving reduced systems with block-diagonal matrices:

$$\underbrace{\left[\mathcal{U}_{<k}{}^{\top}\mathbf{A}_{\mathcal{J}_k}\mathcal{U}_{<k}\right]}_{\sum_\beta \tilde{A}_\beta \otimes \mathrm{diag}(\tilde{d}_\beta)}\mathbf{u}_k = \mathcal{U}_{<k}^{\top}\mathbf{f}_{\mathcal{J}_k}$$

Most important properties:

- **Spatial unknowns:** $\mathcal{O}(r)$ decoupled deterministic PDEs (preconditioning...)
- **Parameters:** $\mathcal{O}(dnr)$ dense $r \times r$ systems solved directly at $\mathcal{O}(dnr^4)$ cost.

Given the $(k_1, \ldots, k_d)$-element of the TT approximation of the FE solution

$$u_h(x^{(j)}, y_1^{(k_1)}, \ldots, y_d^{(k_d)}) \approx \mathbf{u}_0(j)\mathbf{u}_1(k_1)\ldots\mathbf{u}_d(k_d), \quad j = 1, \ldots, M_h\,,$$

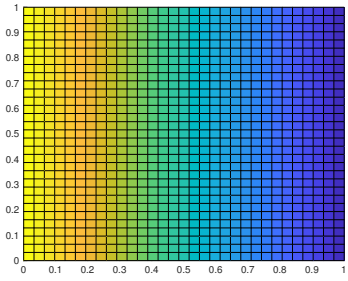
the interpolant at an arbitrary point $x \in D$, $y \in \Gamma$ is evaluated as

$$u_h^{(\mathrm{SC})}(x,y) = \left[\sum_{j=1}^{M_h}\mathbf{u}_0(j)\psi_j(x)\right]\left[\sum_{k_1=1}^{n_1}\mathbf{u}_1(k_1)\mathcal{L}_{k_1}(y_1)\right]\ldots\left[\sum_{k_d=1}^{n_d}\mathbf{u}_d(k_d)\mathcal{L}_{k_d}(y_d)\right]$$

at $\mathcal{O}(r + dnr^2)$ cost ! (since the ψ_j are local and each TT core $\mathbf{u}_j$ has dimension $r \times r$)

Numerical experiment

$$-\nabla a(x,y)\nabla u = 0 \quad \text{in} \quad (0,1)^2$$
$$u|_{x_1=0} = 1, \qquad u|_{x_1=1} = 0,$$
$$\frac{\partial u}{\partial n}|_{x_2=0} = \frac{\partial u}{\partial n}|_{x_2=1} = 0.$$



- Karhunen-Loève like expansion [Eigel, Pfeffer, Schneider '16]

$$a(x,y) = 10 + \sum_{k=1}^{d} y_k a_k(x)$$

with $y_k \sim U[-1,1]$ and $\|a_k\|_\infty = \mathcal{O}(k^{-\frac{\nu+1}{2}})$ (in the experiment below $\nu = 3$).

- Discretisation with bilinear FEs on uniform mesh.
- Truncation dimension d and mesh size h chosen such that bias error is less than requested error tolerance.
- Quantity of Interest: 10 first moments of average over a subdomain.

Benchmarking ALS-Cross vs. (ML)QMC and Sparse Grids

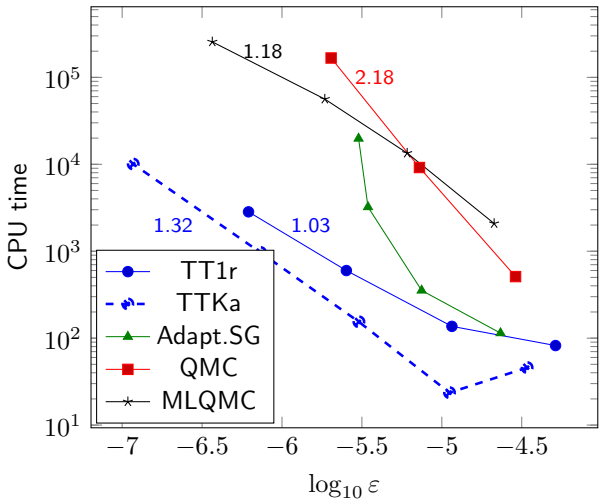
Quasi MC with lattice vector (Kuo)
`lattice-39102-1024-1048576.3600.txt`

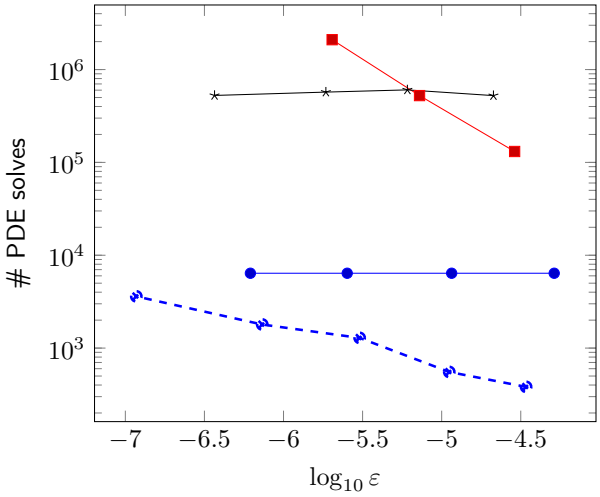
Multilevel QMC with same vector

Adaptive Sparse Grids toolbox (Klimke)
`www.ians.uni-stuttgart.de/spinterp/`

ALS-Cross TT algorithm

- TT1r: 1 iteration with random initial guess and TT ranks 800
- TTKa: K iterations with initial guess the coefficient and lower TT ranks





7. Conclusions & Further Reading

Conclusions & Further Reading

- Low-rank tensor approximation is a powerful, general-purpose high-dimensional approximation tool.
- TT cross approximation provides a very efficient algorithm to compute a low-rank approximation of a high dimensional tensor.
- Very competitive to deep neural networks **!**
- So far **no rigorous convergence theory** for the TT cross iteration.
- **Extension** of the idea to stochastic collocation systems arising from stochastic PDEs $\rightarrow$ efficient **surrogates** of the PDE solution.
  - More details in Dolgov & **RS**, A Hybrid Alternating Least Squares – TT-Cross Algorithm for Parametric PDEs, *SIAM/ASA J Uncertain Q*, **7**, 2019
  - See also Ballani & Grasedyck, Hierarchical Tensor Approximation of Output Quantities of parameter-dependent PDEs, *SIAM/ASA J Uncertain Q*, **3**, 2015
- **Current research in our group:** TT approximation of distributions, in particular PDE-constrained Bayesian inverse problems and normalizing flows
  - Dolgov, Anaya, Fox & **RS**, Approximation and Sampling of Multivariate Probability Distributions in the TT Decomposition, *Stats & Comput*, **30**, 2020
  - Rohrbach, Dolgov, Grasedyck & **RS**, Rank Bounds for Approximating Gaussian Densities in the Tensor-Train Format, *arXiv Preprint*, arXiv:2001.08187, 2020

References for this Chapter

- Bebendorf, Approximation of boundary element matrices, *Numer Math*, **86**, 2000
- Goreinov, Oseledets, Savostyanov, Tyrtyshnikov & Zamarashkin, How to find a good submatrix, ICM Research Report 08-10, Hong Kong Baptist Univ, 2010
- Goreinov & Tyrtyshnikov, The maximum-volume concept in approximation by low-rank matrices, *Contemporary Math*, **208**, 2001
- Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*, Springer, Berlin, 2012
- Holtz, Rohwedder & Schneider, The alternating linear scheme for tensor optimisation in the tensor train format, *SIAM J Sci Comput*, **34**, 2012
- Nouy, Low-rank methods for high-dimensional approximation and model order reduction, in Benner, Cohen, Ohlberger & Willcox (eds.), Model Reduction and Approximation: Theory and Algorithms, SIAM, Philadelphia, 2016
- Oseledets, Tensor-Train Decomposition, *SIAM J Sci Comput*, **33**, 2011
- Oseledets & Tyrtyshnikov, TT-cross approximation for multidimensional arrays, *Linear Alg Appl*, **432**, 2010
- Savostyanov, Quasioptimality of maximum-volume cross interpolation of tensors, *Linear Alg Appl*, **458**, 2014
- Tyrtyshnikov, Incomplete cross approximation in the mosaic-skeleton method, *Computing*, **64**, 2000

High-Dimensional Approximation and Applications in Uncertainty Quantification

Appendix: Background material

[Based on the notes of Prof. Dr Oliver Ernst (TU Chemnitz)]

Prof. Dr Robert Scheichl
r.scheichl@uni-heidelberg.de

Dr Alexander Gilbert
a.gilbert@uni-heidelberg.de

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Institut für Angewandte Mathematik, Universität Heidelberg

Summer Semester 2020

A. Probability Theory
- Random Variables
- Random Vectors
- Limit Theorems
- Statistical Estimation

B. Elliptic Boundary Value Problems
- Weak Formulation
- Finite Element Approximation
- Finite Element Convergence

A. Probability Theory

Probability Theory

Probability measure

We denote an abstract probability space by $(\Omega, \mathfrak{A}, \mathbb{P})$, in which

- Ω is an abstract set of elementary events,
- $\mathfrak{A}$ is a σ-algebra of subsets of Ω containing the measurable events and
- $\mathbb{P}$ is a probability measure on $\mathfrak{A}$.

Definition A.1

A measure $\mathbb{P}$ on a measurable space $(\Omega, \mathfrak{A})$ is called a probability measure if $\mathbb{P}(\Omega) = 1$.

Definition A.2

An event $A \in \mathfrak{A}$ is said to occur almost surely with respect to the measure $\mathbb{P}$ ($\mathbb{P}$-a.s.) if $\mathbb{P}(A) = 1$.

Probability Theory

Borel-Cantelli lemma

Proposition A.3 (Boole's inequality)

For events $\{A_n\}_{n\in\mathbb{N}}$ *there holds*

$$\mathbb{P}\left(\cup_{n=1}^{\infty} A_n\right) \leq \sum_{n=1}^{\infty} \mathbb{P}(A_n).$$

Definition A.4

The set of all $\omega \in \Omega$ such that $\omega \in A_n$ for infinitely many values of n, i.e., ω occurs infinitely often (i.o.), is defined as

$$\{A_n, \text{ i.o. }\} := \limsup_{n\in\mathbb{N}} A_n := \cap_{k=1}^{\infty} \cup_{n=k}^{\infty} A_n.$$

Theorem A.5 (Borel-Cantelli Lemma)

If $\sum_{n=1}^{\infty} \mathbb{P}(A_n) < \infty$, *then* $\mathbb{P}\{A_n, i.o.\} = 0$. *For independent events* $\{A_n\}_{n\in\mathbb{N}}$ *such that* $\sum_{n=1}^{\infty} \mathbb{P}(A_n) = \infty$ *there holds* $\mathbb{P}\{A_n, \text{i.o.}\} = 1$.

Probability Theory

Random variables

Definition A.6

Let $(\Omega, \mathfrak{A}, \mathbb{P})$ be a probability space and $(E, \mathfrak{E})$ a measurable space. A measurable function $X : \Omega \to E$ is called an (E-valued) random variable. Individual values $X(\omega)$ for $\omega \in \Omega$ are called realisations of the random variable.

Remark: If E is a topological space then the σ-algebra generated by the open subsets of E is called the Borel σ-algebra $\mathfrak{B}(E)$.

Definition A.7

Let X be an E-valued random variable where $(E, \mathfrak{E})$ is a measurable space and $(\Omega, \mathfrak{A}, \mathbb{P})$ is the underlying probability space. The probability distribution $\mathbb{P}_X$ of X (also called the law of X) is the probability measure on $(E, \mathfrak{E})$ defined by $\mathbb{P}_X(A) := \mathbb{P}(X^{-1}(A))$ for pre-images $X^{-1}(A) := \{\omega \in \Omega : X(\omega) \in A)\}$ of sets $A \in \mathfrak{E}$.

Remark: This construction is sometimes called the push-forward measure defined by $(\Omega, \mathfrak{A}, \mathbb{P})$, $(E, \mathfrak{E})$ and X.

Probability Theory

Expectation, moments

Definition A.8

The expectation of a Banach space-valued random variable X is defined as the integral

$$\mathbb{E}[X] := \int_\Omega X(\omega)\,\mathrm{d}\mathbb{P}(\omega).$$

Definition A.9

The k-th moment ($k \in \mathbb{N}$) of a real-valued random variable X is $\mathbb{E}\left[X^k\right]$.
The first moment $\mu := \mathbb{E}[X]$ is also called the mean or mean value.
The central moments $\mathbb{E}\left[(X-\mu)^k\right]$ measure the deviation of X from its mean.
The second central moment

$$\mathbf{Var}\,X := \mathbb{E}\left[(X-\mu)^2\right] = \mathbb{E}\left[X^2\right] - \mu^2$$

of a random variable X is called its variance.

Remark: The quantity $\sigma := \sqrt{\mathbf{Var}\,X}$ is called the standard deviation of X.

Probability Theory

Computation of moments

Moments of a random variable are sometimes more easily computed by integrating over the image variable.

Consider a real-valued random variable X from $(\Omega, \mathfrak{A})$ to $(\Gamma, \mathfrak{B}(\Gamma))$ where $\Gamma \subset \mathbb{R}$. For $B \in \mathfrak{B}(\Gamma)$, set $A := X^{-1}(B)$. Then by the definition of the probability distribution $\mathbb{P}_X$

$$\int_\Omega \mathbb{1}_A(\omega)\,\mathrm{d}\mathbb{P}(\omega) = \mathbb{P}(A) = \mathbb{P}_X(B) = \int_\Gamma \mathbb{1}_B(x)\,\mathrm{d}\mathbb{P}_X(x).$$

For measurable functions $f : \Gamma \to \mathbb{R}$ we have

$$\int_\Omega f(X(\omega))\,\mathrm{d}\mathbb{P}(\omega) = \int_\Gamma f(x)\,\mathrm{d}\mathbb{P}_X(x)$$

and, in particular,

$$\mathbb{E}[X] = \int_\Omega X(\omega)\,\mathrm{d}\mathbb{P}(\omega) = \int_\Gamma x\,\mathrm{d}\mathbb{P}_X(x).$$

Probability Theory

Probability density functions

Definition A.10

Let $\mathbb{P}$ be a probability measure on $(\Gamma, \mathfrak{B}(\Gamma))$ for some $\Gamma \subset \mathbb{R}$. If there exists a function $p : \Gamma \to [0, \infty)$ such that $\mathbb{P}(B) = \int_B p(x)\,\mathrm{d}x$ for any $B \in \mathfrak{B}(\Gamma)$ we say that $\mathbb{P}$ has a density p with respect to Lebesgue measure and we call p its probability density function (pdf). If X is a Γ-valued random variable on $(\Omega, \mathfrak{A}, \mathbb{P})$, the pdf p_X of X (if it exists) is the pdf of the probability distribution $\mathbb{P}_X$.

For real-valued random variables X from $(\Omega, \mathfrak{A}, \mathbb{P})$ to $(\Gamma, \mathfrak{B}(\Gamma))$ we then have[1]

$$\mathbb{E}[X] = \int_\Omega X(\omega)\,\mathrm{d}\mathbb{P}(\omega) = \int_\Gamma x\,\mathrm{d}\mathbb{P}_X(x) = \int_\Gamma x p(x)\,\mathrm{d}x. \tag{A.1}$$

Event probabilities are then easily calculated as

$$\mathbb{P}(X \in (a,b)) = \mathbb{P}(\{\omega \in \Omega : a < X(\omega) < b\}) = \mathbb{P}_X((a,b)) = \int_a^b p(x)\,\mathrm{d}x.$$

[1] (where we have omitted the subscript X)

Probability Theory

Uniform distribution

A random variable X is uniformly distributed on $D = [a,b] \subset \mathbb{R}$, $(a < b)$, denoted

$$X \sim \mathrm{Uni}(a,b),$$

if its pdf is

$$p(x) = \frac{1}{b-a}, \qquad x \in [a,b].$$

Using (A.1), we easily obtain

$$\mathbb{E}[X] = \int_a^b \frac{x}{b-a}\,\mathrm{d}x = \frac{a+b}{2}, \qquad \mathbb{E}\left[X^2\right] = \int_a^b \frac{x^2}{b-a}\,\mathrm{d}x = \frac{b^3-a^3}{3(b-a)},$$

so that $\mathbf{Var}\,X = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2 = \frac{(b-a)^2}{12}$.

Probability Theory

Gaussian distribution

A random variable X is said to follow the Gaussian or normal distribution on $\Gamma = \mathbb{R}$ if its pdf is given by

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right), \qquad x \in \mathbb{R},$$

with two real parameters $\mu \in \mathbb{R}$ and $\sigma > 0$, denoted $X \sim \mathrm{N}(\mu, \sigma^2)$.
As is easily verified,

$$\mathbb{E}[X] = \mu, \qquad \textbf{Var}\, X = \sigma^2.$$

The probability that X is within α of its mean is given by

$$\mathbb{P}(|X-\mu| \le \alpha) = \mathrm{erf}\left(\frac{\alpha}{\sqrt{2\sigma^2}}\right),$$

with the error function erf defined by

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, \mathrm{d}t.$$

Probability Theory

Gaussian distribution

The cumulative distribution function (cdf) of the standard normal distribution $\mathrm{N}(0,1)$ is denoted by

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}}\, \mathrm{d}t = \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{x}{\sqrt{2}}\right).$$

Any (finite) linear combination of (jointly) random variables is normally distributed.

Probability Theory

Change of variables formula

Lemma A.11 (Change of variables)

Suppose $Y : \Omega \to \mathbb{R}$ is a real-valued random variable and $f : (a, b) \to \mathbb{R}$ is continuously differentiable with inverse function f^{-1}. If p_Y is the pdf of Y, the pdf of the random variable $X : \Omega \to (a, b)$ defined via $X = f^{-1}(Y)$ is

$$p_X(x) = p_Y(f(x))\,|f'(x)| \qquad \text{for } a < x < b.$$

Probability Theory

Lognormal distribution

If $Y \sim \mathrm{N}(\mu, \sigma^2)$, then the random variable

$$X \coloneqq \exp(Y)$$

is said to follow a lognormal distribution. With $f(x) = \log x$, Lemma A.11 yields the pdf of X as

$$p_X(x) = \frac{1}{\sqrt{2\pi\sigma^2 x^2}} \exp\left(-\frac{[\log(x) - \mu]^2}{2\sigma^2}\right).$$

Moreover, there holds

$$\mathbb{E}[X] = \exp\left(\mu + \frac{\sigma^2}{2}\right), \qquad \textbf{Var}\, X = (e^{\sigma^2} - 1)e^{2\mu+\sigma^2}.$$

Probability Theory

Covariance

Definition A.12

The covariance between two real-valued random variables is defined as

$$\mathbf{Cov}(X, Y) = \mathbb{E}\left[(X - \mu_X)(Y - \mu_Y)\right],$$

where $\mu_X := \mathbb{E}[X]$ and $\mu_Y := \mathbb{E}[Y]$. In particular, $\mathbf{Cov}(X, X) = \mathbf{Var}\, X$.

Note: An equivalent expression is $\mathbf{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y]$.

Calculation of the covariance requires evaluating the integral

$$\mathbb{E}[XY] = \int_\Omega X(\omega)Y(\omega)\,\mathrm{d}\mathbb{P}(\omega) = \int_{X(\Omega)\times Y(\Omega)} xy\,\mathrm{d}\mathbb{P}_{X,Y}(x, y),$$

in which $\mathbb{P}_{X,Y}$ is the joint probability distribution of X and Y.
Sometimes it is useful to scale the covariance to lie in $[-1, 1]$. The resulting quantity is known as the correlation coefficient

$$\rho(X, Y) := \frac{\mathbf{Cov}(X, Y)}{\sigma_X \sigma_Y}.$$

Probability Theory

Joint probability distribution

Definition A.13

The joint probability distribution of two random variables X and Y is the distribution of the bivariate random variable $\mathbf{X} = (X, Y)$, i.e., for all $B \in \mathfrak{B}(X(\Omega) \times Y(\Omega))$

$$\mathbb{P}_{X,Y}(B) = \mathbb{P}(\{\omega \in \Omega : \mathbf{X}(\omega) \in B\}).$$

If it exists, the density $p_{X,Y}$ of $\mathbb{P}_{X,Y}$ is known as the joint pdf and

$$\mathbb{P}_{X,Y} = \int_B p_{X,Y}(x, y)\,\mathrm{d}x\,\mathrm{d}y.$$

Probability Theory

Uncorrelated random variables

Definition A.14

If $\mathbf{Cov}(X, Y) = 0$ the random variables X and Y are said to be uncorrelated. A family $\{X_\alpha\}_\alpha$ is said to be pairwise uncorrelated if X_α and X_β are uncorrelated for all $\alpha \neq \beta$.

Note: Uncorrelated random variables may still be strongly related. As an example,

$$X \sim \mathrm{N}(0, 1), \qquad \text{and} \qquad Y := \cos X$$

satisfy $\mu_X = 0$ and hence

$$\begin{aligned}\mathbf{Cov}(X, Y) = \mathbb{E}\left[X \cos X\right] &= \int_{\mathbb{R}} x \cos(x) \, \mathrm{d}\mathbb{P}_X(x) \\ &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} x \cos(x) \exp\left(\frac{-x^2}{2}\right) \mathrm{d}x = 0.\end{aligned}$$

A stronger notion is that of independent random variables.

Probability Theory

Sub σ-algebras, σ-algebras generated by random variables

Definition A.15

A σ-algebra $\mathfrak{B}$ is a sub σ-algebra of $\mathfrak{A}$ if $\mathfrak{B} \subset \mathfrak{A}$, i.e., if $A \in \mathfrak{B}$ implies $A \in \mathfrak{A}$.

Definition A.16

Let X be an E-valued random variable on $(\Omega, \mathfrak{A}, \mathbb{P})$ for a measurable space $(E, \mathfrak{E})$. The σ-algebra generated by X, denoted $\sigma(X)$, is defined as

$$\sigma(X) := \{X^{-1}(A) : A \in \mathfrak{E}\} \subset \mathfrak{A}.$$

Remark: $\sigma(X)$ is the smallest σ-algebra such that X is measurable. It may be considerably smaller than $\mathfrak{A}$.

Probability Theory

Independence of events, σ-algebras and random variables

Definition A.17

Two events $A, B \in \mathfrak{A}$ are independent if $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$.
Two σ-algebras $\mathfrak{A}_1$ and $\mathfrak{A}_2$ are independent if all pairs of events A_1 and A_2 with $A_1 \in \mathfrak{A}_1$ and $A_2 \in \mathfrak{A}_2$ are independent.

Definition A.18

Two random variables X, Y on a probability space $(\Omega, \mathfrak{A}, \mathbb{P})$ are said to be independent if the σ-algebras $\sigma(X)$ and $\sigma(Y)$ are independent.
A family $\{X_\alpha\}_\alpha$ of random variables is said to be pairwise independent if X_α and X_β are independent for all $\alpha \neq \beta$.

Independence of random variables X and Y can be conveniently determined using their joint distribution $\mathbb{P}_{X,Y}$: X and Y are independent if and only if $\mathbb{P}_{X,Y}$ equals the product measure $\mathbb{P}_X \times \mathbb{P}_Y$. If X and Y are real-valued with densities p_X and p_Y, they are independent if and only if their joint pdf is

$$p_{X,Y}(x, y) = p_X(x)p_Y(y).$$

Probability Theory

Independence implies uncorrelatedness

Lemma A.19

If X and Y are independent real-valued random variables and $\mathbb{E}[|X|], \mathbb{E}[|Y|] < \infty$, then X and Y are uncorrelated.

Note: The converse is generally false.

Theorem A.20 (Jensen's inequality)

If X is a real-valued random variable with $\mathbb{E}[|X|] < \infty$ and $\phi : \mathbb{R} \to \mathbb{R}$ a convex function, then

$$\phi(\mathbb{E}[X]) \leq \mathbb{E}[\phi(X)]. \tag{A.2}$$

Probability Theory

Bochner spaces

Definition A.21

Let $(\Omega, \mathfrak{A}, \mathbb{P})$ be a probability space and let W be a separable Banach space with norm $\|\cdot\|$. We denote by $L^p(\Omega; W)$, $1 \leq p < \infty$, the space of W-valued $\mathfrak{A}$-measurable random variables $X : \Omega \to W$ with $\mathbb{E}\left[\|X\|^p\right] < \infty$. The resulting space is a Banach space with the norm

$$\|X\|_{L^p(\Omega;W)} := \left(\int_\Omega \|X(\omega)\|^p \, \mathrm{d}\mathbb{P}(\omega)\right)^{1/p} = \mathbb{E}\left[\|X\|^p\right]^{1/p}.$$

Similarly, $L^\infty(\Omega; W)$ is the Banach space of W-valued random variables $X : \Omega \to W$ for which

$$\|X\|_{L^\infty(\Omega;W)} = \operatorname*{ess\,sup}_{\omega\in\Omega} \|X(\omega)\| < \infty.$$

Probability Theory

Bochner spaces, $p = 2$

The case $p = 2$ when W is a Hilbert space $W = H$ with inner product $(\cdot, \cdot)$ occurs frequently. In this case $L^2(\Omega; H)$ is a Hilbert space with inner product

$$(X, Y)_{L^2(\Omega;H)} := \mathbb{E}\left[(X, Y)\right] = \int_\Omega (X(\omega), Y(\omega)) \, \mathrm{d}\mathbb{P}(\omega).$$

Random variables in $L^2(\Omega; H)$ are called mean-square integrable random variables.

For random variables $X, Y \in L^2(\Omega; H)$ the Cauchy-Schwarz inequality takes on the form

$$|(X, Y)_{L^2(\Omega;H)}| \leq \|X\|_{L^2(\Omega;H)} \|Y\|_{L^2(\Omega;H)}$$

or

$$\mathbb{E}\left[(X, Y)\right] \leq \mathbb{E}\left[\|X\|^2\right]^{1/2} \mathbb{E}\left[\|Y\|^2\right]^{1/2}.$$

Probability Theory

Bochner spaces, $p = 2$, covariance

Definition A.22

Let H be a separable Hilbert space. A linear operator $C : H \to H$ is the covariance of two H-valued random variables X and Y if

$$(C\phi, \psi) = \mathbf{Cov}((\phi, X), (\psi, Y)) \qquad \forall \phi, \psi \in H.$$

X and Y are said to be uncorrelated if C is the zero operator. If $Y = X$ then C is called the covariance of X.

More generally, the covariance of two random variables X and Y with values in a separable Banach space W may be defined as a bilinear map $c : W' \times W' \to \mathbb{R}$ on the dual space W' of W such that

$$c(\phi, \psi) = \mathbf{Cov}(\langle \phi, X \rangle_{W' \times W}, \langle \psi, Y \rangle_{W' \times W}) \qquad \forall \phi, \psi \in W'.$$

Here $\langle \cdot, \cdot \rangle_{W' \times W}$ denotes the duality bracket between W' and W. The bilinear map c may be identified with a linear operator from $C : W' \to W''$ via the identity

$$\langle C\phi, \psi \rangle_{W'' \times W'} = c(\phi, \psi).$$

Probability Theory

Convergence of random variables

Definition A.23

Let W be a Banach space with norm $\|\cdot\|$ and $\{X_n\}_{n \in \mathbb{N}}$ be a sequence of W-valued random variables. We say X_n converges to $X \in W$

almost surely if $X_n(\omega) \to X(\omega)$ for almost all $\omega \in \Omega$, i.e., if

$$\mathbb{P}\left(\|X_n - X\| \to 0 \text{ for } n \to \infty\right) = 1.$$

in probability if $\mathbb{P}\left(\|X_n - X\| > \epsilon\right) \to 0$ for $n \to \infty$ for any $\epsilon > 0$.

in p-th mean or in $L^p(\Omega; W)$ if $\mathbb{E}\left[\|X_n - X\|^p\right] \to 0$ as $n \to \infty$. When $p = 2$ this is known as convergence in mean square.

in distribution if $\mathbb{E}\left[\phi(X_n)\right] \to \mathbb{E}\left[\phi(X)\right]$ as $n \to \infty$ for any bounded and continuous function $\phi : W \to \mathbb{R}$.

Probability Theory

Convergence of random variables

Theorem A.24

Let $X_k \to X$ in p-th mean and, for $r > 0$ and a constant $K = K(p)$, assume that

$$\|X_k - X\|_{L^p(\Omega;W)} := \mathbb{E}\left[\|X_k - X\|^p\right]^{1/p} \leq \frac{K(p)}{k^r}. \tag{A.3}$$

Then the following convergence properties apply:

(a) *$X_k \to X$ in probability and, for any $\epsilon > 0$,*

$$\mathbb{P}\left(\|X_k - X\| \geq k^{-r+\epsilon}\right) \leq \frac{K(p)^p}{k^{p\epsilon}}. \tag{A.4}$$

(b) *$\mathbb{E}\left[\phi(X_k)\right] \to \mathbb{E}\left[\phi(X)\right]$ for all Lipschitz continuous functions on W and, if L denotes a Lipschitz constant of ϕ,*

$$\left|\mathbb{E}\left[\phi(X_k)\right] - \mathbb{E}\left[\phi(X)\right]\right| \leq L\frac{K(p)}{k^r}.$$

(c) *If (A.3) holds for all p sufficiently large, then $X_k \to X$ a.s. Furthermore, for each $\epsilon > 0$ there exists a nonnegative random variable K such that $\|X_k(\omega) - X(\omega)\| \leq K(\omega)k^{-r+\epsilon}$ for almost all ω.*

Probability Theory

Random vectors

Random variables $\mathbf{X} = (X_1, \ldots, X_n)^\mathsf{T}$ from $(\Omega, \mathfrak{A}, \mathbb{P})$ to $(\Gamma, \mathfrak{B}(\Gamma)$ with $\Gamma \subset \mathbb{R}^n$ are known as random vectors or multivariate random variables (bivariate for $n = 2$).

Their expected value

$$\boldsymbol{\mu} = \mathbb{E}\left[\mathbf{X}\right] = \int_\Omega \mathbf{X}(\omega)\,\mathrm{d}\mathbb{P}(\omega) = \left[\mathbb{E}\left[X_1\right], \ldots, \mathbb{E}\left[X_n\right]\right]^\mathsf{T}$$

is a vector in $\mathbb{R}^n$. If $\mathbf{X}$ has a pdf p, then for $B \in \mathfrak{B}(\Gamma)$

$$\mathbb{P}(\mathbf{X} \in B) = \mathbb{P}(\{\omega \in \Omega : \mathbf{X}(\omega) \in B\}) = \mathbb{P}_\mathbf{X}(B) = \int_B p(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

The components $\{X_j\}_{j=1}^n$ of $\mathbf{X}$ are (pairwise) independent if and only if $\mathbb{P}_\mathbf{X}$ is the product measure $\mathbb{P}_{X_1} \times \cdots \times \mathbb{P}_{X_n}$. In terms of the pdf, this is equivalent to

$$p(\mathbf{x}) = p_{X_1}(x_1) \cdot p_{X_2}(x_2) \cdots p_{X_n}(x_n).$$

Probability Theory

Multivariate uniform

A random vector $\mathbf{X} : \Omega \to \Gamma$ with values in a set $\Gamma \subset \mathbb{R}^n$ with finite Lebesgue measure $|\Gamma|$ follows a multivariate uniform distribution on Γ, denoted by

$$\mathbf{X} \sim \mathrm{Uni}(\Gamma)$$

if it has the pdf

$$p(\mathbf{x}) \equiv \frac{1}{|\Gamma|}, \qquad \mathbf{x} \in \Gamma.$$

Probability Theory

Covariance matrix

Definition A.25

The covariance of two real-valued random vectors $\mathbf{X} = [X_1, \ldots, X_m]^\mathsf{T}$ and $\mathbf{Y} = [Y_1, \ldots, Y_n]^\mathsf{T}$ is given by the $m \times n$ matrix

$$\mathbf{Cov}(\mathbf{X}, \mathbf{Y}) = \mathbb{E}\left[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{Y} - \mathbb{E}[\mathbf{Y}])^\mathsf{T}\right].$$

$\mathbf{X}$ and $\mathbf{Y}$ are said to be uncorrelated if $\mathbf{Cov}(\mathbf{X}, \mathbf{Y}) = \mathbf{O}$ (the $m \times n$ zero matrix). The matrix $\mathbf{Cov}(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{n \times n}$ is called the covariance matrix of $\mathbf{X}$.

Proposition A.26

Let $\mathbf{X}$ be an $\mathbb{R}^n$-valued random variable with mean vector $\boldsymbol{\mu}$ and covariance matric $\mathbf{C}$. Then $\mathbf{C}$ is symmetric positive semi-definite and its trace is given by $\mathbb{E}\left[\|\mathbf{X} - \boldsymbol{\mu}\|_2^2\right]$.

Probability Theory

Multivariate normal distribution

A random vector with mean vector $\boldsymbol{\mu}$ and positive definite covariance matrix $\mathbf{C}$ is said to follow an n-variate Gaussian distribution if it has the pdf

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det \mathbf{C}}} \exp\left(\frac{-(\mathbf{x}-\boldsymbol{\mu})^\mathsf{T}\mathbf{C}^{-1}(\mathbf{x}-\boldsymbol{\mu})}{2}\right). \tag{A.5}$$

Definition A.27

An $\mathbb{R}^n$-valued random vector $\mathbf{X}$ follows a multivariate normal (or Gaussian) distribution, denoted

$$\mathbf{X} \sim \mathrm{N}(\boldsymbol{\mu}, \mathbf{C}),$$

where $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\mathbf{C} \in \mathbb{R}^{n\times n}$ is symmetric positive definite, if it has the pdf (A.5).

Note: The case that $\mathbf{C}$ is singular (pos. *semi*-definite) can be handled by characteristic functions.

Probability Theory

Multivariate normal distribution

If $\mathbf{X} \sim \mathrm{N}(\boldsymbol{\mu}, \mathbf{C})$ is a multivariate normal random vector, then for any $\mathbf{a} \in \mathbb{R}^n$ the linear combination

$$Y = \mathbf{a}^\top \mathbf{X} = \sum_{k=1}^{n} a_k X_k$$

follows the normal distribution $Y \sim \mathrm{N}(\mathbf{a}^\top\boldsymbol{\mu}, \mathbf{a}^\top\mathbf{C}\mathbf{a})$.

Probability Theory

i.i.d. random variables

Definition A.28

A sequence $\{X_j\}_{j\in\mathbb{N}}$ of random variables is said to be independent and identically distributed (i.i.d.) if they all follow the same probability distribution and, in addition, are pairwise independent.

The classical limit theorems of probability theory concern sums of i.i.d. random variables. For an i.i.d. sequence $\{X_j\}_{j\in\mathbb{N}}$, we introduce the notation

$$S_n := X_1 + \cdots + X_n, \qquad n \in \mathbb{N}.$$

Probability Theory

Weak Law of Large Numbers

Theorem A.29 (Chebyshev inequality)

A random variable X with finite mean μ and finite variance σ^2 satisfies

$$c^2\mathbb{P}(|X-\mu| \geq c) \leq \sigma^2.$$

Theorem A.30 (WLLN)

Let $\{X_k\}_{k\in\mathbb{N}}$ be a sequence of i.i.d. random variables on a given probability space $(\Omega, \mathfrak{A}, \mathbb{P})$ with mean μ and finite variance. Then

$$\frac{S_n}{n} \to \mu \quad \text{in probability, i.e.}$$

for ever fixed $\epsilon > 0$ there holds

$$\mathbb{P}\left(|S_n/n - \mu| > \epsilon\right) \to 0 \quad \text{as} \quad n \to \infty.$$

Probability Theory

Strong Law of Large Numbers

Theorem A.31 (SLLN)

Let $\{X_k\}_{k\in\mathbb{N}}$ be a sequence of i.i.d. real-valued random variables on a given probability space $(\Omega, \mathfrak{A}, \mathbb{P})$. Then S_n/n has a finite limit if and only if $\mathbb{E}[|X_1|] < \infty$, in which case

$$\frac{S_n}{n} \to \mathbb{E}[X_1] \qquad \text{a.s.}$$

If $\mathbb{E}[|X_1|] = \infty$, then $\limsup_{n\to\infty} |S_n|/n \to \infty$ a.s.

Probability Theory

Central Limit Theorem

Let the sequence $\{X_k\}_{k\in\mathbb{N}}$ of real-valued random variables be independent, but not necessarily identically distributed. In addition, let $\mathbb{E}[X_k] = 0$ and $\mathbb{E}\left[X_k^2\right] < \infty$ for all k.

Besides $S_n = \sum_{k=1}^n X_k$, introduce the quantities

$$\sigma_k^2 := \mathbf{Var}\, X_k,$$

$$\Sigma_n^2 := \sum_{j=1}^n \sigma_j^2 = \mathbf{Var}\, S_n.$$

The central limit theorem (CLT) is the statement that

$$\lim_{n\to\infty} \frac{S_n}{\Sigma_n} = \lim_{n\to\infty} \frac{S_n - \mathbb{E}[S_n]}{\sqrt{\mathbf{Var}\, S_n}} \sim \mathrm{N}(0,1) \qquad \text{in distribution.}$$

Probability Theory

Central Limit Theorem

Definition A.32 (Lyapunov condition)

The sequence $\{X_k\}_{k\in\mathbb{N}}$ satisfies the Lyapunov condition if $\mathbb{E}\left[|X_k|^3\right] < \infty$ for each k and

$$\lim_{n\to\infty} \frac{1}{\Sigma_n^2} \sum_{k=1}^{n} \mathbb{E}\left[|X_k|^3\right] = 0.$$

Theorem A.33 (Lyapunov CLT)

If $\{X_k\}_{k\in\mathbb{N}}$ satisfies the Lyapunov condition, then $S_n/\Sigma_n \to \mathrm{N}(0,1)$ in distribution.

Note: There exist several variants of the CLT with different assumptions.

Theorem A.34 (Simple CLT)

Let $\{X_k\}_{k\in\mathbb{N}}$ be a sequence of i.i.d. random variables, with $\mathbb{E}[X_k] = \mu$ and **Var** *$X_k = \sigma^2$ for all $k \in \mathbb{N}$. Then $\sqrt{n}(S_n/n - \mu) \to \mathrm{N}(0, \sigma^2)$ in distribution.*

Probability Theory

Berry-Esseen Theorem

Theorem A.35 (Berry, 1941; Esseen 1942)

Let $\{X_k\}_{k\in\mathbb{N}}$ be i.i.d. random variables such that, for all $k \in \mathbb{N}$,

$$\mu := \mathbb{E}\left[X_k\right], \quad \sigma^2 := \textbf{Var}\, X_k > 0, \quad \rho := \mathbb{E}\left[|X_k - \mu|^3\right] < \infty.$$

If F_n denotes the distribution function of $(S_n - n\mu)/(\sigma\sqrt{n})$ and Φ that of the standard normal distribution $\mathrm{N}(0,1)$, then, with a universal constant C,

$$\sup_{x\in\mathbb{R}} |\Phi(x) - F_n(x)| \le C \cdot \frac{\rho}{\sigma^3\sqrt{n}}.$$

Note: the constant C is known to satisfy $0.4097 \le C \le 0.7056$ [Shevtsova, 2007].

Statistical Estimation

- Estimation theory is concerned with determining an unknown quantity θ associated with the probability distribution of a random variable X given n i.i.d. samples $\{X_k\}_{k=1}^n$ of X.
- Typical examples of such quantities θ are moments of X's distribution such as the mean and the variance. Another common situation is the estimation of one or more parameters which determine the distribution of X.
- An estimator for a scalar quantity θ is a function

$$\phi : \mathbb{R}^n \to \mathbb{R}, \qquad \hat{\theta} = \phi(X_1, \ldots, X_n)$$

  mapping n i.i.d. realisations of X to the estimate $\hat{\theta}$ of θ.
- Note that, since each of the n random samples X_k are random variables, the same is true of

$$\hat{\theta} = \hat{\theta}(\omega) = \phi(X_1(\omega), \ldots, X_n(\omega)).$$

  Once the samples have been drawn/realised, the estimate $\hat{\theta}$ is a real number.

Statistical Estimation

Sample average, unbiased estimator

- The sample average

$$\hat{\mu}_n := \frac{X_1 + \cdots + X_n}{n}$$

  is an estimate for the mean $\mu = \mathbb{E}[X]$.
- Since the X_k are i.i.d., we conclude from the linearity of expectation that

$$\mathbb{E}[\hat{\mu}_n] = \frac{1}{n} \sum_{k=1}^{n} \mathbb{E}[X_k] = \frac{1}{n} \cdot n\mu = \mu.$$

- If $\mathbb{E}[|X|] < \infty$ the SLLN tells us that also $\hat{\mu}_n \to \mu = \mathbb{E}[X]$ a.s. as $n \to \infty$.
- Since $\mathbf{Var}\, \hat{\mu}_n = \frac{\sigma^2}{n}$, where $\sigma^2 = \mathbf{Var}\, X$, we note that the variance $\hat{\mu}_n$ decreases like $1/n$ with growing sample size.

Definition A.36

An estimator for which $\mathbb{E}[\hat{\theta}] = \theta$ is called unbiased.

Statistical Estimation

Sample variance

The sample variance

$$\hat{\sigma}_n^2 := \frac{1}{n-1} \sum_{k=1}^{n} (X_k - \hat{\mu}_n)^2$$

is an unbiased estimator for $\sigma^2 = \mathbf{Var}\, X$.

In addition, there holds $\hat{\sigma}_n^2 \to \sigma^2$ a.s. as $n \to \infty$.

References

[1] O. Kallenberg. *Foundations of Modern Probability* Springer, Berlin-Heidelberg, 1997.

[2] I. G. Shevtsova. Sharpening of the upper bound of the absolute constant in the Berry–Esseen inequality. *Theory Probab. Appl.*, 51, 549–553, 2007.

B. Elliptic Boundary Value Problems

Elliptic Boundary Value Problem

We consider the elliptic boundary value problem (BVP) of finding the solution of the partial differential equation with Dirichlet boundary condition

$$-\nabla\cdot(a\nabla u) = f \qquad \text{on } D, \tag{B.1a}$$

$$u = g \qquad \text{on } \partial D, \tag{B.1b}$$

given a bounded convex domain $D \subset \mathbb{R}^d$, $d = 1, 2, 3$ with sufficiently smooth boundary ∂D, a coefficient function $a : D \to \mathbb{R}^+$, a source term $f : D \to \mathbb{R}$ and boundary data in the form of a function $g : \partial D \to \mathbb{R}$.
The differential operator in (B.1a) is short for

$$\nabla\cdot(a\nabla u) = \sum_{j=1}^{d} \frac{\partial}{\partial x_j}\left(a(\mathbf{x})\frac{\partial u(\mathbf{x})}{\partial x_j}\right)$$

Equation (B.1a) is a model for diffusion phenomena occurring in , e.g., heat conduction, electrostatics, potential flow and elasticity. Generalisations of (B.1) involve the addition of lower-order terms, other boundary conditions, a matrix-valued coefficient function and dependence of a on u.

Elliptic Boundary Value Problem

Strong and weak solution

If $f \in C(\overline{D})$ and $a \in C^1(\overline{D})$, then a function $u \in C^2(D) \cap C^1(\overline{D})$ which satisfies (B.1) is called a classical solution or a strong solution of the boundary value problem.

There are (theoretical and practical) reasons for generalizing the classical solution concept. The key to this generalisation lies in reformulating (B.1) as a variational problem. Multiplying both sides of (B.1a) by an arbitrary function $\phi \in C_0^\infty(D)$, in this context known as a test function, and integrating by parts, we observe that any (classical) solution of (B.1) also satisfies the equation

$$a(u, \phi) = \ell(\phi) \qquad \text{for all } \phi \in C_0^\infty(D), \tag{B.2}$$

with the symmetric bilinear form $a(\cdot, \cdot)$ and linear functional $\ell(\cdot)$ given by

$$a(u, \phi) = \int_D a(\mathbf{x}) \nabla u(x) \cdot \nabla \phi(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \qquad \ell(\phi) = \int_D f(\mathbf{x}) \phi(\mathbf{x}) \, \mathrm{d}\mathbf{x}. \tag{B.3}$$

For (B.2) to make sense, it is sufficient that the integrals and derivatives are well-defined.

Elliptic Boundary Value Problem

Strong and weak solution

This is the case if u and ϕ are taken to lie in the Sobolev space

$$H^1(D) := \{v \in L^2(D) : \nabla v \in L^2(D)^2\},$$

which is a Hilbert space with respect to the inner product

$$(u, v)_{H^1(D)} = \int_D (\nabla u \cdot \nabla v + uv) \, \mathrm{d}\mathbf{x} = (\nabla u, \nabla v) + (u, v),$$

where we use $(\cdot, \cdot)$ to denote the inner product in $L^2(D)$. The associated norm on $H^1(D)$ is

$$\|u\|^2_{H^1(D)} = \int_D \left(|\nabla u|^2 + u^2\right) \, \mathrm{d}\mathbf{x}.$$

The gradients are in terms of weak derivatives in the sense of

$$\left(\frac{\partial u}{\partial x_j}, \phi\right) = -\left(u, \frac{\partial \phi}{\partial x_j}\right) \qquad \text{for all } \phi \in C_0^\infty(D).$$

Elliptic Boundary Value Problem

Strong and weak solution

Stating the boundary condition (B.1b) requires a well-defined notion of evaluating a function from $H^1(D)$ on the lower-dimensional manifold ∂D.

- Functions in $H^1(D)$ satisfying the BC with homogeneous boundary data $g \equiv 0$ are can be characterised as lying in the subspace $H_0^1(D) \subset H^1(D)$, which is defined as the closure of smooth functions with compact support with respect to $\|\cdot\|_{H^1}$:

$$H_0^1(D) := \overline{C_0^\infty(D)} \subset H^1(D).$$

- For inhomogeneous boundary data we define the space

$$W := H_g^1(D) := \{v \in H^1(D) : u_{|\partial D} = g\}.$$

  The evaluation on the boundary is understood in the following sense: for a sufficiently smooth boundary there exists a bounded trace operator $\gamma : H^1(D) \to L^2(\partial D)$ such that for all $u \in C^1(\overline{D})$ there holds $\gamma u = u_{|\partial D}$. Since $C^1(\overline{D})$ is dense in $H^1(D)$, we have $\gamma u = \lim_{n\to\infty} u_{|\partial D}$ for any approximating sequence $\{u_n\} \subset C^1(\overline{D})$ converging to u in $H^1(D)$.

Elliptic Boundary Value Problem

Strong and weak solution

Definition B.1

The trace space of $H^1(D)$ for a sufficiently smooth domain D is defined as

$$H^{1/2}(\partial D) := \gamma(H^1(D)) = \{\gamma u : u \in H^1(D)\}.$$

$H^{1/2}(\partial D)$ is a Hilbert space with norm

$$\|g\|_{H^{1/2}(\partial D)} := \inf\{\|u\|_{H^1(D)} : \gamma u = g, u \in H^1(D)\}.$$

Sine in general $H^{1/2}(\partial D) \subsetneq L^2(\partial D)$, boundary data g in (B.1b) must be chosen from $H^{1/2}(\partial D)$.

Lemma B.2

There exists $C_\gamma > 0$ such that, for all $g \in H^{1/2}(\partial D)$, we can find $u_g \in H^1(D)$ with $\gamma u_g = g$ and

$$\|u_g\|_{H^1(D)} \le C_\gamma \|g\|_{H^{1/2}(\partial D)}$$

Elliptic Boundary Value Problem

Strong and weak solution

We denote the spaces of trial and test functions by

$$W := H_g^1(D), \quad \text{and} \quad V := H_0^1(D).$$

Assumption 1

The coefficient function $a = a(\mathbf{x})$ in (B.1a) satisfies

$$0 < a_{\min} \le a(\mathbf{x}) \le a_{\max} < \infty \qquad \text{for almost all } \mathbf{x} \in D$$

for positive constants $a_{\min}$ and $a_{\max}$. In particular, $a \in L^\infty(D)$ and a is uniformly bounded away from zero.

By Assumption 1, the bilinear form $a(\cdot,\cdot)$ is bounded on $H^1(D)$, i.e.,

$$|a(u,v)| \le C\|u\|_{H^1(D)}\|v\|_{H^1(D)}, \qquad \text{for all } u,v \in H^1(D)$$

with a constant $C \le \|a\|_{L^\infty(D)}$.

Elliptic Boundary Value Problem

Strong and weak solution

Definition B.3

A weak solution of (B.1) is a function $u \in W$ such that

$$a(u,v) = \ell(v) \qquad \text{for all } v \in V, \tag{B.4}$$

with $a(\cdot,\cdot)$ and $\ell(\cdot)$ as defined in (B.3).

Elliptic Boundary Value Problem

Strong and weak solution

Definition B.4

A bilinear form $a : H \times H \to \mathbb{R}$ on a Hilbert space H is said to be coercive if there exists a constant $\alpha > 0$ such that

$$a(u,u) \geq \alpha \|u\|_H^2 \qquad \text{for all } u \in H.$$

Lemma B.5 (Lax–Milgram)

Let H be a real Hilbert space with norm $\|\cdot\|_H$ and let ℓ be a bounded linear functional on H. Let $a : H \times H \to \mathbb{R}$ be a bilinear form that is bounded and coercive. Then there exists a unique $u_\ell \in H$ such that $a(u_\ell, v) = \ell(v)$ for all $v \in H$, and the solution depends continuously on the data

$$\|u_\ell\|_H \leq \frac{1}{\alpha}\|\ell\|.$$

Elliptic Boundary Value Problem

Strong and weak solution

For functions in $H^1(D)$ we introduce the H^1 semi-norm

$$|u|_{H^1(D)} := \left(\int_D |\nabla u|^2 \, \mathrm{d}\mathbf{x} \right)^{1/2}.$$

as well as the energy norm associated with the coefficient function a as

$$|u|_a := a(u,u)^{1/2} = \left(\int_D a \nabla u \cdot \nabla u \, \mathrm{d}\mathbf{x} \right)^{1/2}.$$

Theorem B.6 (Poincaré–Friedrichs inequality)

For a bounded domain D there exists a constant $C = C_D > 0$ such that

$$\|u\|_{L^2(D)} \leq C_D |u|_{H^1(D)} \qquad \textit{for all } u \in H_0^1(D).$$

Elliptic Boundary Value Problem

Strong and weak solution

Lemma B.7

Under Assumption 1 the bilinear form $a : H^1(D) \times H_0^1(D) \to \mathbb{R}$ is bounded and the energy norm is equivalent to the H^1 semi-norm on $H^1(D)$.

Theorem B.8

Let Assumption 1 hold, $f \in L^2(D)$ and $g \in H^{1/2}(\partial D)$. Then (B.1) *has a unique weak solution $u \in W = H_g^1(D)$. Furthermore, the weak solution $u \in W$ satisfies*

$$|u|_{H^1(D)} \leq C \left(\|f\|_{L^2(D)} + \|g\|_{H^{1/2}(\partial D}) \right)$$

where $C = \max\{C_D/a_{\min}, C_\gamma(1 + a_{\max}/a_{\min})\}$.

Proof. Lax–Milgram Lemma. □

Finite Element Approximation

Galerkin discretisation

Given: linear variational problem of finding $u \in V$, V a Hilbert space with norm $\|\cdot\|$, such that

$$a(u, v) = \ell(v) \qquad \text{for all } v \in V \tag{B.5}$$

with a bilinear form $a(\cdot, \cdot)$ and linear form $\ell(\cdot)$ on V which satisfy the assumptions of the Lax-Milgram lemma.

Galerkin method for finding approximate solutions of (B.5) proceeds by restricting the problem to a finite-dimensional subspace $V_n \subset V$: denote by $u_n \in V_n$ the solution of

$$a(u_n, v_n) = \ell(v_n) \qquad \text{for all } v_n \in V_n. \tag{B.6}$$

Note: The Galerkin approximation u_n of u with respect to the space V_n is uniquely determined since the conditions of the Lax-Milgram Lemma are satisfied for Problem (B.6) by inclusion.

Finite Element Approximation

Céa's lemma

Galerkin orthogonality

The Galerkin solution $u_n \in V_n$ satisfies

$$a(u - u_n, v_n) = 0, \qquad \text{for all } v_n \in V_n.$$

The simple structure of a linear variational problem allows its reduction to a problem of best approximation.

Lemma B.9 (Céa)

If the assumptions of the Lax-Milgram lemma apply to Problem (B.5) *with solution $u \in V$, then the Galerkin approximation u_n, i.e., the solution of* (B.6), *satisfies*

$$\|u - u_n\| \leq \frac{C}{\alpha} \inf_{v_n \in V_n} \|u - v_n\|. \tag{B.7}$$

Finite Element Approximation

Céa's lemma, symmetric case

- If the bilinear form $a(\cdot, \cdot)$ is, in addition, symmetric (Hermitian) then, because of coercivity, it defines an inner product on V.
- Galerkin orthogonality then implies u_n is the a-orthogonal projection of u onto V_n and therefore the best approximation to u from V_n with respect to the associated (energy) norm.
- In the energy norm (B.7) is therefore satisfied with $C = \alpha = 1$.
- Coercivity and boundedness also imply that the energy norm is equivalent to $\|\cdot\|$, i.e.,

  $$\sqrt{\alpha}\|v\| \leq |v|_a \leq \sqrt{C}\|v\| \qquad \text{for all } v \in V,$$

  which leads to the improved estimate over (B.7)

$$\|u - u_n\| \leq \sqrt{\frac{C}{\alpha}} \inf_{v \in V_n} \|u - v\|.$$

Finite Element Approximation

Application to elliptic BVP

We have seen that, for the elliptic BVP (B.1), we have the equivalences

$$\|\cdot\|_{H^1(D)} \asymp |\cdot|_{H^1(D)} \asymp |\cdot|_a.$$

Corollary B.10

Under Assumption 1, the Galerkin approximation u_n fo the solution of the elliptic boundary value problem (B.1)*, with respect to any subspace V_n of $V = H_0^1(D)$, satisfies*

$$|u - u_n|_a = \inf_{v \in V_n} |u - v|_a,$$

$$|u - u_n|_{H^1(D)} \le \sqrt{\frac{a_{\min}}{a_{\max}}}\, |u - v|_{H^1(D)} \qquad \text{for all } v \in V_n.$$

Finite Element Approximation

Galerkin system

Given a basis $\{v_1, \dots, v_n\}$ of V_n and the solution $u_n = \sum_{j=1}^n \xi_j v_j$, then the Galerkin variational equation (B.6) is equivalent to

$$\sum_{j=1}^n \xi_j \; a(v_j, v_i) = \ell(v_i), \qquad i = 1, \dots, n,$$

which, when rewritten as a linear system of equation, becomes the Galerkin system

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{B.8}$$

with Galerkin matrix $[\mathbf{A}]_{i,j} = a(v_j, v_i)$, unknown vector $[\mathbf{x}]_i = \xi_i$ and right-hand side vector $[\mathbf{b}]_i = \ell(v_i)$.

- If $a(\cdot,\cdot)$ is symmetric, then so is $\mathbf{A}$.
- If $a(\cdot,\cdot)$ is coercive, then $\mathbf{A}$ is (uniformly) positive definite.

Finite Element Approximation

The finite element method

- Different Galerkin methods result from different choices of subspaces.
- Wavelets.
- Trigonometric functions, global polynomials (spectral methods).
- Radial basis functions.
- The finite element method employs finite dimensional subspaces of the variational spaces (trial and test spaces) consisting of piecewise polynomials with respect to a partition of D.
- We shall assume in the following that D is a polygon (polyhedron), but the finite element method can also be applied to domains with curved boundaries.
- For the remainder of this section we consider the case where $D \subset \mathbb{R}^2$, i.e., $d = 2$. The concepts can easily be extended to different d.

Finite Element Approximation

Triangulations

Assumptions on the partition of the domain D, denoted by $\mathscr{T}_h$ with elements K:

- (Z_1) $\overline{D} = \cup_{K \in \mathscr{T}_h} K$.
- (Z_2) Each $K \in \mathscr{T}_h$ is a closed set with nonempty interor $\mathring{K}$.
- (Z_3) For two distinct $K_1, K_2 \in \mathscr{T}_h$ there holds $\mathring{K}_1 \cap \mathring{K}_2 = \emptyset$.
- (Z_4) Each $K \in \mathscr{T}_h$ has a Lipschitz-continuous boundary ∂K.

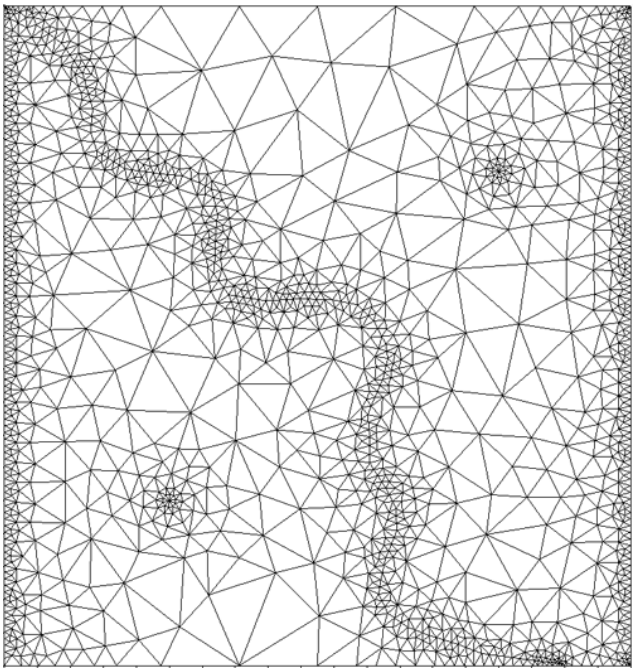
The partition is usually assigned a discretisation parameter $h > 0$ given by

$$h := \max_{K \in \mathscr{T}_h} \operatorname{diam} K,$$

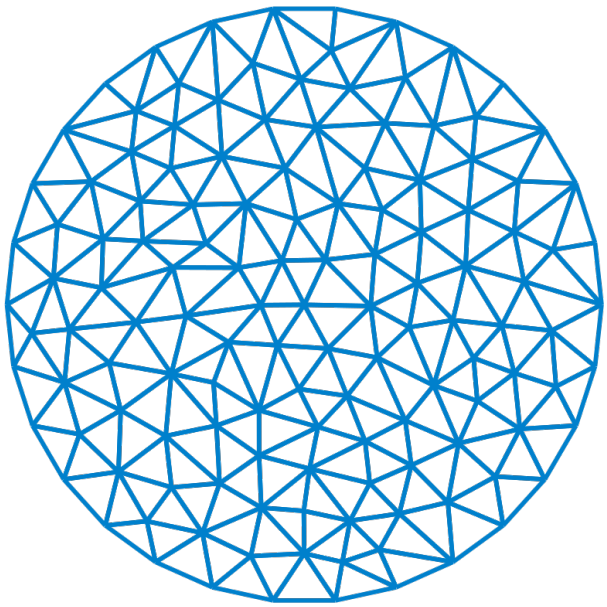
which is a measure of how fine the partition is.

Finite Element Approximation

Triangulations



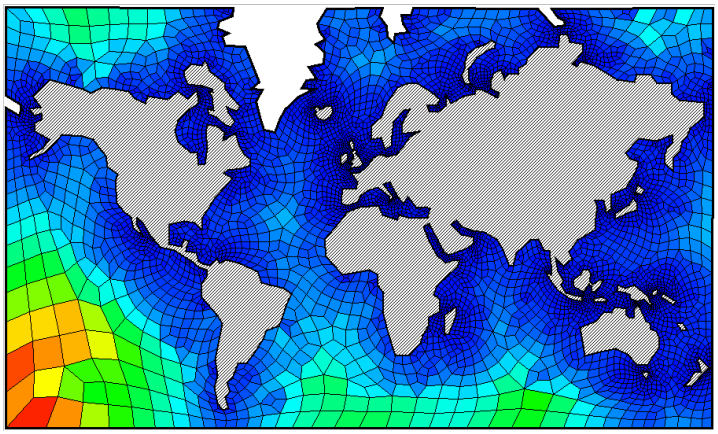
Triangular mesh on a square domain.



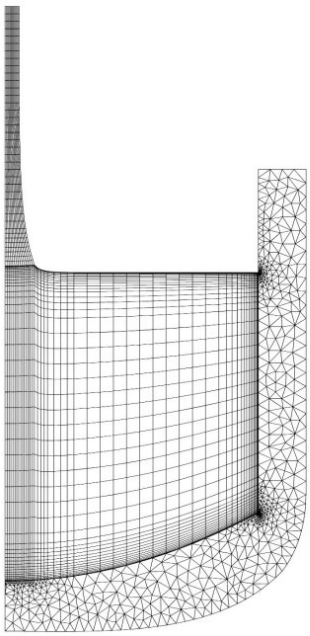
Triangular mesh on a polygonal approximation of a circle.

Finite Element Approximation

Triangulations



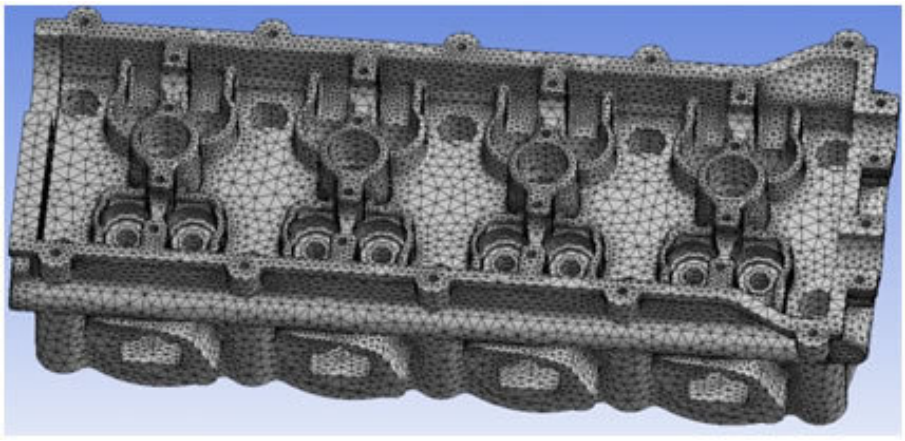
Quadrilateral mesh on a rectangular (exterior) domain.



Mesh consisting of triangles and quadrilaterals.

Finite Element Approximation

Triangulations



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

Finite Element Approximation

H^1-conforming finite element spaces

A conforming Galerkin approximation is one which employs finite-dimensional spaces V_n such that $V_n \subset V$.

Let V_h denote a space of piecewise continuous functions $v : \overline{D} \to \mathbb{R}$ with respect to an admissible triangulation $\mathscr{T}_h$ of D, i.e., such that each restriction $v|_K$ to any $K \in \mathscr{T}_h$ is continuous on K.

Theorem B.11

With the notation defined above, there holds $V_h \subset H^1(D)$ if, and only if,

$$V_h \subset C(\overline{D}) \qquad \text{and} \qquad \{v|_K : v \in V_h\} \subset H^1(K).$$

In this case $\{v \in V_h : v = 0 \text{ on } \partial D\} \subset H^1_0(D)$.

Finite Element Approximation

Finite elements

According to [Ciarlet, 1978], a finite element is a triple (K, P_K, Ψ_K) such that

(1) K is a nonempty set

(2) P_K is a finite-dimensional space of functions defined on K and

(3) Ψ_K is a set of linearly independent linear functionals ψ on P_K with the property that, for any $p \in P_K$,

$$\psi(p) = 0 \text{ for all } \psi \in \Psi_K \qquad \Rightarrow \qquad p = 0.$$

We shall consider a single finite element, the so-called linear triangle, where

(1) $K \in \mathbb{R}^2$ is a triangle with (non-collinear) vertices $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$,

(2) P_K is the space of all affine functions on K and

(3) Ψ_K consists of the three functionals

$$\Psi_K = \{\psi_j : P_K \to \mathbb{R}, \psi_j(p) = p(\mathbf{x}_j), j = 1, 2, 3\}.$$

Finite Element Approximation

Trianglular finite elements

- To construct a (global) finite element space V_h based on linear triangle elements consider a triangulation $\mathscr{T}_h$ of D consisting of (closed) triangles K which satisfy properties (**Z1**)–(**Z4**).
- The functions in V_h will also lie in $H^1(D)$ if they are continuous on $\overline{D}$, which, for piecewise linear (polynomial) functions, is equivalent to their being continuous across triangle boundaries.
- We thus obtain the space

$$V_h := \{v \in C(\overline{D}) : v|_K \in \mathscr{P}_1 \text{ for all } K \in \mathscr{T}_h\},$$

  where $\mathscr{P}_k$ denotes the space of (multivariate) polynomials of (complete) degree k.
- Define the subspace $V_{h,0}$ of V_h by

$$V_{h,0} := \{v \in V_h : v|_{\partial D} = 0\} \subset H_0^1(D).$$

Finite Element Approximation

Degrees of freedom, nodal basis

- A continuous piecewise linear function in V_h is completely determined by its values at all triangle vertices.
- Such a (finite) set of parameters which uniquely determine a finite element function is called a set of degrees of freedom (DOF).
- In $V_{h,0}$ these are the values at all nodes which do not lie on ∂D; denote their number by n.
- A particularly convenient basis $\{\phi_1, \ldots, \phi_n\}$ of $V_{h,0}$ is the so-called nodal basis characterised by

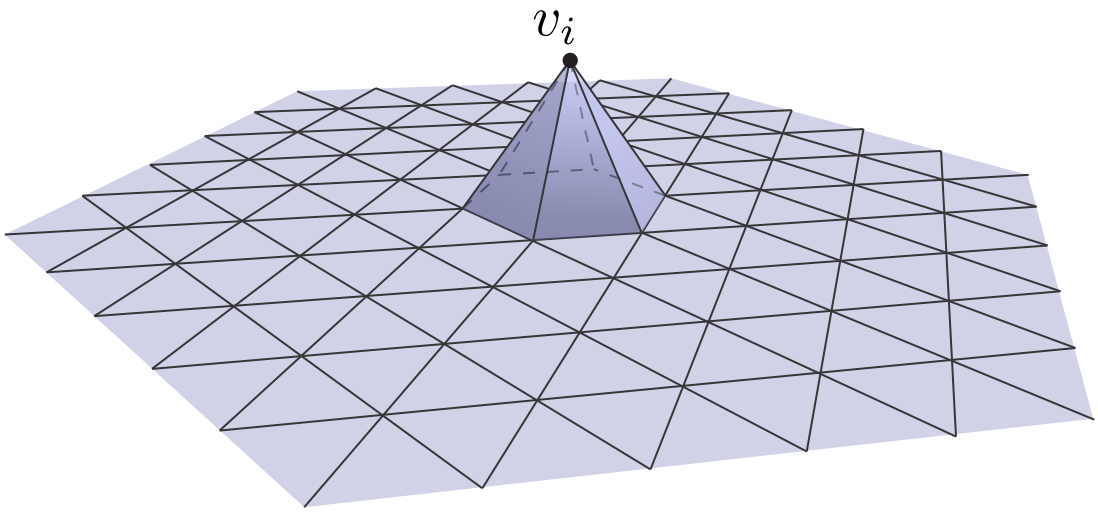
$$\phi_j(\mathbf{x}_i) = \delta_{i,j} \qquad i, j = 1, \ldots, n.$$

- If $\mathcal{N}_h = \{x_1, \ldots, x_n\}$ denotes the set of vertices $x_j \notin \partial D$, then

$$\operatorname{supp} \phi_j = \bigcup_{\substack{K \in \mathcal{T}_h \\ \mathbf{x}_j \in K}} K.$$

Finite Element Approximation

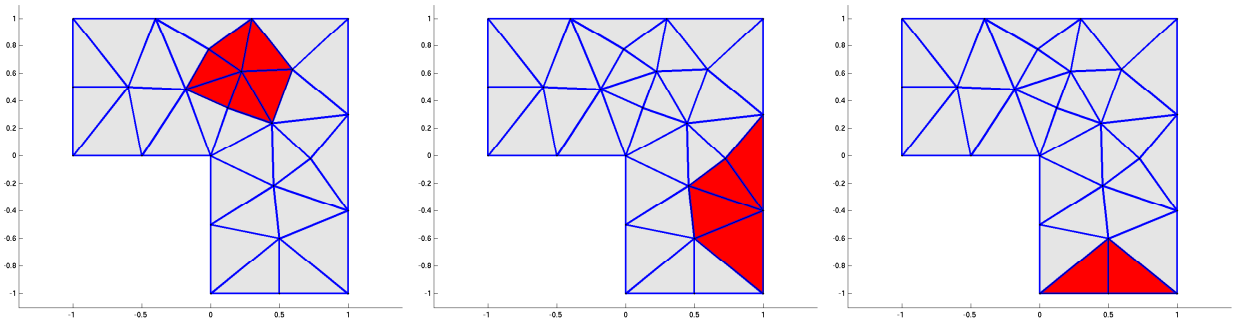
Nodal basis for linear triangles



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

Galerkin matrix, linear triangles

Implications for Galerkin system (B.8):

$$[\mathbf{b}]_i = \ell(\phi_i) = \int_D f\phi_i \,\mathrm{d}\mathbf{x} = \int_{\operatorname{supp}\phi_i} f\phi_i \,\mathrm{d}\mathbf{x},$$

$$\begin{aligned}[\mathbf{A}]_{i,j} = a(\phi_j,\phi_i) &= \int_D a(\mathbf{x})\phi_i(\mathbf{x})\cdot\nabla\phi_j(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= \int_{\operatorname{supp}\phi_i\cap\operatorname{supp}\phi_j} a(\mathbf{x})\nabla\phi_i(\mathbf{x})\cdot\nabla\phi_j(\mathbf{x})\,\mathrm{d}\mathbf{x}.\end{aligned}$$

In particular, the Galerkin matrix $\mathbf{A}$ is sparse.

Finite Element Approximation

Finite element assembly

Common procedure in assembling the Galerkin system:

(1) Ignore boundary condition initially, i.e., consider all of V_h with nodal basis

$$\{\phi_1, \phi_2, \ldots, \phi_n, \phi_{n+1}, \ldots, \phi_{\tilde{n}}\},$$

$\tilde{n} - n$ the number of vertices on the boundary ∂D.
Yields matrix $\tilde{\mathbf{A}} \in \mathbb{R}^{\tilde{n}\times\tilde{n}}$, vector $\tilde{\mathbf{b}} \in \mathbb{R}^{\tilde{n}}$.

(2) Then eliminate the DOF associated with boundary vertices.
Yields matrix $\mathbf{A}$, vector $\mathbf{b}$.

Note:

- Initial approach for step (1): compute $\tilde{\mathbf{A}}, \tilde{\mathbf{b}}$, entry by entry, i.e., basis function by basis function
- But: shape and connectivity of supports typically very different.
- Simpler: compute $\mathbf{A}$, $\mathbf{b}$ element by element.

Finite Element Approximation

Finite element assembly

$K \in \mathscr{T}_h$: then for $i, j = 1, 2 \ldots, \tilde{n}$:

$$a(\phi_j, \phi_i) = \int_D a\nabla\phi_j \cdot \nabla\phi_i \,\mathrm{d}\mathbf{x} = \sum_{K\in\mathscr{T}_h} \int_K a\nabla\phi_j \cdot \nabla\phi_i \,\mathrm{d}\mathbf{x} =: \sum_{K\in\mathscr{T}_h} a_K(\phi_j, \phi_i),$$

$$\ell(\phi_i) = \int_D f\phi_i \,\mathrm{d}\mathbf{x} = \sum_{K\in\mathscr{T}_h} \int_K f\phi_i \,\mathrm{d}\mathbf{x} =: \sum_{K\in\mathscr{T}_h} \ell_K(\phi_i).$$

Setting

$$[\tilde{\mathbf{A}}_K]_{i,j} := a_K(\phi_j, \phi_i) \qquad i, j = 1, 2, \ldots, \tilde{n},$$
$$[\tilde{\mathbf{b}}_K]_i := \ell_K(\phi_i, \qquad i = 1, 2, \ldots, \tilde{n},$$

we obtain

$$\tilde{\mathbf{A}} = \sum_{K\in\mathscr{T}_h} \tilde{\mathbf{A}}_K, \qquad \tilde{\mathbf{b}} = \sum_{K\in\mathscr{T}_h} \tilde{\mathbf{b}}_K.$$

Finite Element Approximation

Finite element assembly: element table

Since each element belongs to the support of exactly three basis functions, only (at most) nine entries of $\tilde{\mathbf{A}}_K$ and three entries of $\mathbf{b}_K$ are nonzero.
Which entries these are can be determined by maintaining an element table:

$$[G(i,j)]_{i=1,2,3;j=1,\ldots,n_K}:$$

Element	K_1	K_2	$\ldots$	K_{n_K}
first vertex	$i_1^{(1)}$	$i_1^{(2)}$	$\ldots$	$i_1^{(n_K)}$
second vertex	$i_2^{(1)}$	$i_2^{(2)}$	$\ldots$	$i_2^{(n_K)}$
third vertex	$i_3^{(1)}$	$i_3^{(2)}$	$\ldots$	$i_3^{(n_K)}$

Here n_K denotes the number of triangles in $\mathcal{T}_h$.

Besides the global vertex numbering

$$x_1, x_2, \ldots, x_{\tilde{n}},$$

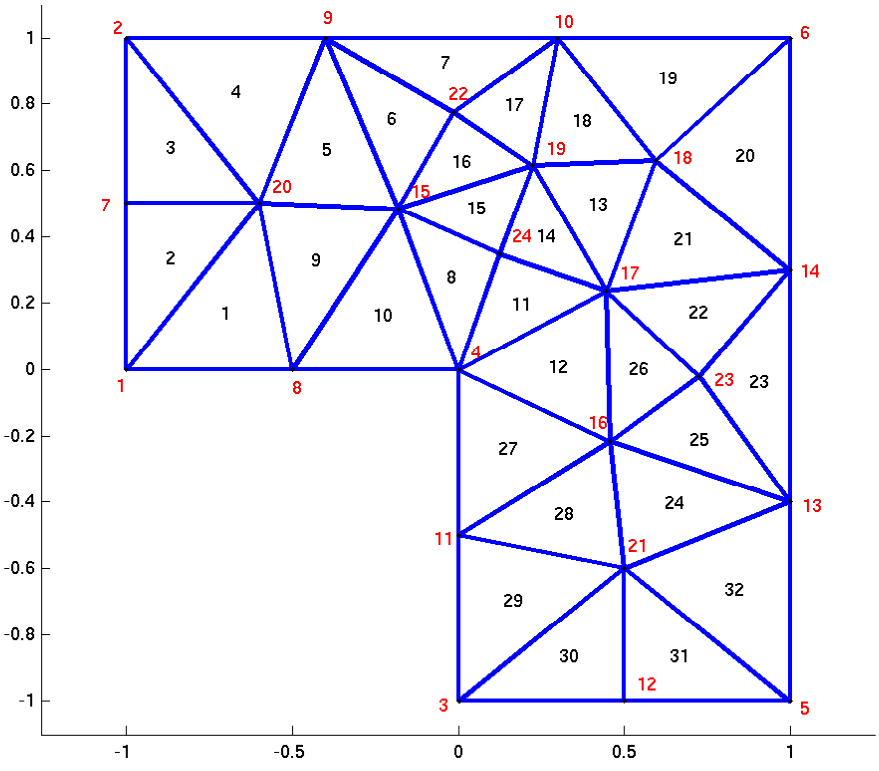
the element table introduces a second, local vertex numbering

$$x_1^{(K)},\ x_2^{(K)},\ x_3^{(K)}$$

of the vertices (DOFs) associated with K. G is the local to global mapping of the DOFs.

Finite Element Approximation

Finite element assembly



Global numbering of vertices (red) and elements (black) in a triangulation of an L-shaped domain.

Finite Element Approximation

Finite element assembly

With this notation the nonzero submatrix $\mathbf{A}_K$ of $\tilde{\mathbf{A}}_K$ and nonzero subvector $\mathbf{b}_K$ of $\tilde{\mathbf{b}}_K$ are given by

$$\mathbf{A}_K := \begin{bmatrix} a_K(\phi_1^{(K)}, \phi_1^{(K)}) & a_K(\phi_2^{(K)}, \phi_1^{(K)}) & a_K(\phi_3^{(K)}, \phi_1^{(K)}) \\ a_K(\phi_1^{(K)}, \phi_2^{(K)}) & a_K(\phi_2^{(K)}, \phi_2^{(K)}) & a_K(\phi_3^{(K)}, \phi_2^{(K)}) \\ a_K(\phi_1^{(K)}, \phi_3^{(K)}) & a_K(\phi_2^{(K)}, \phi_3^{(K)}) & a_K(\phi_3^{(K)}, \phi_3^{(K)}) \end{bmatrix}, \quad \mathbf{b}_K := \begin{bmatrix} \ell_K(\phi_1^{(K)}) \\ \ell_K(\phi_2^{(K)}) \\ \ell_K(\phi_3^{(K)}) \end{bmatrix}.$$

If K has number k in the enumeration of the elements, then the association of the local numbering $\{\phi_i^{(K)}\}_{i=1,2,3}$ of the three basis functions whose support contains K with the global numbering $\{\phi_j\}_{j=1}^{\tilde{n}}$ of all basis functions is given by

$$\phi_i^{(K)} = \phi_j, \qquad j = G(i,k), \quad i = 1,2,3.$$

$\mathbf{A}_K$ and $\mathbf{b}_K$ are sometimes called the element stiffness matrix and element load vector.

Finite Element Approximation

Finite element assembly

We summarise phase (1) of the finite element assembly process in the following algorithm[2]

Algorithm 1 Phase (1) of finite element assembly.

1: Initialise $\tilde{\mathbf{A}} := \mathbf{O}$, $\tilde{\mathbf{b}} := \mathbf{0}$.
2: **for** $K \in \mathscr{T}_h$ **do**
3: Compute $\mathbf{A}_K$ and $\mathbf{b}_K$
4: $k \leftarrow$ [index of element K]
5: $i_1 \leftarrow G(1,k)$, $i_2 \leftarrow G(2,k)$, $i_3 \leftarrow G(3,k)$
6: $\tilde{\mathbf{A}}([i_1 i_2 i_3], [i_1 i_2 i_3]) \leftarrow \tilde{\mathbf{A}}([i_1 i_2 i_3], [i_1 i_2 i_3]) + \mathbf{A}_K$
7: $\tilde{\mathbf{b}}([i_1 i_2 i_3]) \leftarrow \tilde{\mathbf{b}}([i_1 i_2 i_3]) + \mathbf{b}_K$
8: **end for**

[2] We use the following MATLAB-inspired notation:

$$\mathbf{A}([i_1 i_2 i_3], [i_1 i_2 i_3]) = \begin{bmatrix} a_{i_1,i_1} & a_{i_1,i_2} & a_{i_1,i_3} \\ a_{i_2,i_1} & a_{i_2,i_2} & a_{i_2,i_3} \\ a_{i_3,i_1} & a_{i_3,i_2} & a_{i_3,i_3} \end{bmatrix}, \quad \mathbf{b}([i_1 i_2 i_3]) = \begin{bmatrix} b_{i_1} \\ b_{i_2} \\ b_{i_3} \end{bmatrix}.$$

Finite Element Approximation

Reference element

Both the numerical integration as well as the error analysis benefit from a change of variables to a reference element $\hat{K} \subset \mathbb{R}^2$. Each element $K \in \mathscr{T}_h$ then has a parametrisation $K = \mu_K(\hat{K})$, where

$$\mu_K : \hat{K} \to K, \qquad \hat{K} \ni \boldsymbol{\xi} \mapsto \mathbf{x} \in K, \quad \mathbf{x} = \mu_K(\boldsymbol{\xi}) = B_K \boldsymbol{\xi} + \mathbf{b}_K.$$

Most common for triangular elements: unit simplex

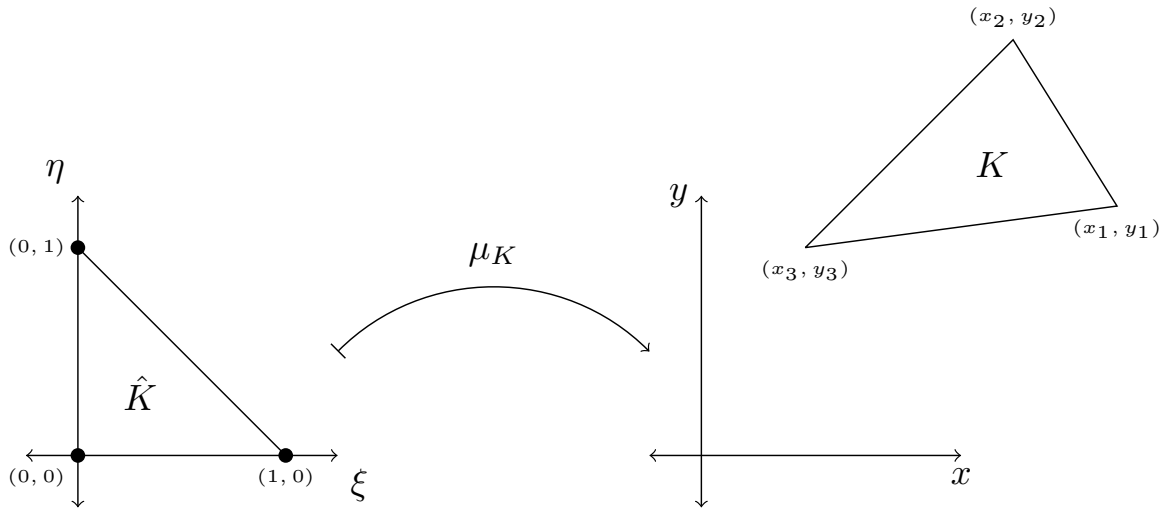
$$\hat{K} = \{(\xi, \eta) \in \mathbb{R}^2 : 0 \le \xi \le 1, 0 \le \eta \le 1 - \xi\}.$$

For each triangle $K \in \mathscr{T}_h$ the affine mapping μ_K is determined by prescribing, e.g.,

$$\begin{aligned} (1,0) &\mapsto (x_1, y_1), \\ (0,1) &\mapsto (x_2, y_2), \\ (0,0) &\mapsto (x_3, y_3), \quad \text{i.e.} \end{aligned}$$

Finite Element Approximation

Reference element



$$\begin{bmatrix} x \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} x_1 - x_3 & x_2 - x_3 \\ y_1 - y_3 & y_2 - y_3 \end{bmatrix}}_{B_K} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \underbrace{\begin{bmatrix} x_3 \\ y_3 \end{bmatrix}}_{\mathbf{b}_K}$$

Finite Element Approximation

Reference element

Local (nodal) basis on $\hat{K}$: (dual basis of DOF)

$$\hat{\phi}_1(\xi,\eta) = \xi, \quad \hat{\phi}_2(\xi,\eta) = \eta, \quad \hat{\phi}_3(\xi,\eta) = 1-\xi-\eta, \qquad (\xi,\eta) \in \hat{K}.$$

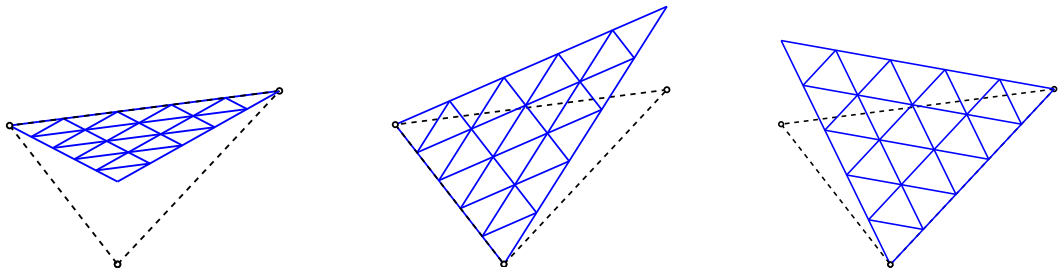
The correspondence

$$\hat{\phi} \mapsto \phi := \hat{\phi} \circ \mu_K^{-1}, \quad \text{d.h.} \quad \phi(\mathbf{x}) := \hat{\phi}(\boldsymbol{\xi}(\mathbf{x})) = \hat{\phi}(\mu_K^{-1}(\mathbf{x}))$$

assigns to $\hat{\phi}$ on $\hat{K}$ a unique function ϕ on K.

Local basis functions on K:

$$\phi_j = \hat{\phi}_j \circ \mu_K^{-1} : K \to \mathbb{R}, \qquad j = 1,2,3.$$



Finite Element Approximation

Reference element, change of variables

The chain rule[3] applied to $\phi(\mathbf{x}) = \hat{\phi}(\boldsymbol{\xi}(\mathbf{x}))$ gives

$$\nabla\phi = \begin{bmatrix}\phi_x \\ \phi_y\end{bmatrix} = \begin{bmatrix}\hat{\phi}_\xi\xi_x + \hat{\phi}_\eta\eta_x \\ \hat{\phi}_\xi\xi_y + \hat{\phi}_\eta\eta_y\end{bmatrix} = \begin{bmatrix}\xi_x & \eta_x \\ \xi_y & \eta_y\end{bmatrix}\begin{bmatrix}\hat{\phi}_\xi \\ \hat{\phi}_\eta\end{bmatrix} = (D\mu_K^{-1})^\top \hat{\nabla}\hat{\phi}.$$

Since $\quad \mathbf{x} = \mu_K(\boldsymbol{\xi}) = B_K\boldsymbol{\xi} + \mathbf{b}_K, \qquad$ i.e. $D\mu_K \equiv B_K,$

$\boldsymbol{\xi} = \mu_K^{-1}(\mathbf{x}) = B_K^{-1}(\mathbf{x} - \mathbf{b}_K), \quad$ i.e. $D\mu_K^{-1} \equiv B_K^{-1}$

we obtain

$$\nabla\phi = B_K^{-\top}\hat{\nabla}\hat{\phi}.$$

[3] $\hat{\nabla}$ indicates differentiation with respect to the variables ξ and η.

Finite Element Approximation

Reference element, element integrals

This finally gives the element integrals ($\phi_i = \phi_i^{(K)}$, $i = 1, 2, 3$)

$$\begin{aligned} a_K(\phi_j, \phi_i) &= \int_K a(\mathbf{x})\, \nabla\phi_j(\mathbf{x}) \cdot \nabla\phi_i(\mathbf{x})\, \mathrm{d}\mathbf{x} \\ &= \int_{\hat{K}} a(\mathbf{x}(\boldsymbol{\xi}))\ \left(B_K^{-\top} \hat{\nabla}\hat{\phi}_j(\boldsymbol{\xi})\right) \cdot \left(B_K^{-\top} \hat{\nabla}\hat{\phi}_i(\boldsymbol{\xi})\right) |\det B_K|\, \mathrm{d}\boldsymbol{\xi}. \end{aligned} \tag{B.9}$$

The determinant is given by (note K is a triangle)

$$|\det B_K| = 2|K|,$$

$$B_K^{-\top} = \frac{1}{2|K|} \begin{bmatrix} y_2 - y_3 & x_3 - x_2 \\ y_3 - y_1 & x_1 - x_3 \end{bmatrix},$$

$$\begin{bmatrix} \hat{\nabla}\hat{\phi}_1 & \hat{\nabla}\hat{\phi}_2 & \hat{\nabla}\hat{\phi}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}.$$

Finite Element Approximation

Eliminate constrained boundary DOF

To impose the Dirichlet boundary condition we require that the Galerkin approximation $u_h \in V_h$ satisfy

$$u_h(\mathbf{x}_j) = g(\mathbf{x}_j) \quad \text{at all boundary vertices } \{\mathbf{x}_j\}_{j=n+1}^{\tilde{n}}. \tag{B.10}$$

- We partition the coefficient vector $\mathbf{u} \in \mathbb{R}^{\tilde{n}}$ into a first block $\mathbf{u}_I \in \mathbb{R}^n$ containing the coefficients associated with the interior vertices $\{\mathbf{x}_j\}_{j=1}^n$ and a second block $\mathbf{u}_B \in \mathbb{R}^{\tilde{n}-n}$ containing the constrained coefficients associated with boundary vertices.
- For the assembled matrix $\tilde{\mathbf{A}}$ and vector $\tilde{\mathbf{b}}$ this induces the partitionings

$$\tilde{\mathbf{A}} = \begin{bmatrix} \tilde{\mathbf{A}}_{II} & \tilde{\mathbf{A}}_{IB} \\ \tilde{\mathbf{A}}_{BI} & \tilde{\mathbf{A}}_{BB} \end{bmatrix}, \qquad \tilde{\mathbf{b}} = \begin{bmatrix} \tilde{\mathbf{b}}_I \\ \tilde{\mathbf{b}}_B \end{bmatrix}.$$

- The constraint (B.10) now reads $\mathbf{u}_B = \mathbf{g}$, where $\mathbf{g} \in \mathbb{R}^{\tilde{n}-n}$ contains the boundary data $\{g(\mathbf{x}_j)\}_{j=n+1}^{\tilde{n}}$.

Finite Element Approximation

Eliminate constrained boundary DOF

This constraint is characterised by there being no coupling of the boundary DOF to either interior DOF or among themselves, resulting in the modified linear system of equations

$$\begin{bmatrix} \tilde{\mathbf{A}}_{II} & \tilde{\mathbf{A}}_{IB} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u}_I \\ \mathbf{u}_B \end{bmatrix} = \begin{bmatrix} \mathbf{b}_I \\ \mathbf{g} \end{bmatrix},$$

which gives the reduced system

$$\mathbf{A}\mathbf{u}_I = \mathbf{b}, \qquad \mathbf{A} = \tilde{\mathbf{A}}_{II}, \quad \mathbf{b} = \mathbf{b}_I - \tilde{\mathbf{A}}_{IB}\mathbf{g}$$

for the interior DOF.

Note that this procedure is a discrete variant of the reformulation of the BVP with inhomogeneous Dirichlet boundary conditions to an equivalent one with homogeneous Dirichlet boundary conditions.

Finite Element Convergence

Summary

- Céa's lemma characterises the Galerkin error as one of best appproximation from the FE subspace V_h.
- An upper bound for this error is the distance of the true solution from its interpolant from the FE subspace. This is the uniquely determined function from V_h which possesses the same global DOF as the exact solution.
- The asymptotic behavior of the interpolant is then analyzed on a sequence of meshes $\{\mathscr{T}_{h_n}\}_{n\in\mathbb{N}}$ with $\lim_{n\to\infty} h_n = 0$.
- For the interpolation error to become small, the mesh sequence has to be shape-regular: if ρ_K denotes the radius of the inscribed circle in K and $h_K = \operatorname{diam} K$, then a sequence of meshes is shape-regular provided the ratio

  $$\frac{\rho_K}{h_K}, \qquad K \in \mathscr{T}_h$$

  is bounded below uniformly for all $\{\mathscr{T}_{h_n}\}$.
- A priori convergence bounds are obtained by relating the smoothness of the exact solution to the convergence rate h^α of the interpolation error as $h \to 0$.

Finite Element Convergence

Extra regularity

Interpolation estimates for u that is only in $H^1(D)$ do not yield a useful rate h^α with an $\alpha > 0$. As such one looks for solutions that possesses higher regularity.

Definition B.12

For $r \in \mathbb{N}$ and $D \subset \mathbb{R}^d$ bounded, we denote by $H^r(D)$ the Sobolev space

$$H^r(D) := \{v \in L^2(D) : D^{\boldsymbol{\alpha}} u \in L^2(D) \text{ for all } \boldsymbol{\alpha} \in \mathbb{N}_0^d, |\boldsymbol{\alpha}| \le r\}.$$

$H^r(D)$ is a Hilbert space with the inner product

$$(u, v)_{H^r(D)} = \sum_{|\boldsymbol{\alpha}| \le r} \int_D (D^{\boldsymbol{\alpha}} u)(D^{\boldsymbol{\alpha}} v)\, \mathrm{d}\mathbf{x},$$

and the induced norm given by

$$\|u\|^2_{H^r(D)} = (u, u)_{H^r(D)} = \sum_{|\boldsymbol{\alpha}| \le r} \|D^{\boldsymbol{\alpha}}\|^2_{L^2(D)}.$$

Note: the vector $\boldsymbol{\alpha} \in \mathbb{N}_0^d$ is called a *multiindex*, and $|\boldsymbol{\alpha}| := \sum_{j=1}^d \alpha_j$.

Finite Element Convergence

Interpolation error of linear FE for H^2-regular functions

- Let V_h denote the space of piecewise linear functions subject to a shape-regular, admissible triangulation $\mathscr{T}_h$ of D.
- Denote by $I_h : C(\overline{D}) \to V_h$ the (global) interpolation operator assigning to each continuous function v the interpolant $v_h \in V_h$ determined by the condition that v_h agrees with v at all vertices of $\mathscr{T}_h$.
- Then the error of best approximation of $u \in C(\overline{D})$ is bounded by the interpolation error

$$\inf_{v \in V_h} |u - v|_{H^1(D)} \le |u - I_h u|_{H^1(D)}.$$

- If the solution u of (B.4) has additional regularity $u \in H^2(D)$, then the Sobolev imbedding theorem assures that u agrees a.e. with a function in $C(\overline{D})$, so that pointwise evaluation of u and thus the interpolant is well-defined.
- In this case a scaling argument can be used to show

$$|u - I_h u|_{H^1(D)} \le C\, h\, |u|_{H^2(D)}$$

with a constant C independent of h and u.

Finite Element Convergence

Model problem

Assumption 2 (H^2/elliptic regularity)

There exists a constant $C_2 > 0$ such that, for every $f \in L^2(D)$, the solution of (B.4) *belongs to $H^2(D)$ and satisfies*

$$|u|_{H^2(D)} \leq C_2 \|f\|_{L^2(D)}.$$

Theorem B.13

Under Assumptions 1 and 2, the solution u of (B.4) *with $f \in L^2(D)$ and the piecewise linear finite element approximation u_h on a sequence of shape-regular meshes satisfy*

$$|u - u_h|_a \leq C\sqrt{a_{\max}}|u|_{H^2(D)}\, h \leq CC_2\sqrt{a_{\max}}\|f\|_{L^2(D)}\, h, \tag{B.11}$$

with a constant C independent of h.

Corollary B.14

Under the assumptions of Theorem B.13 there holds

$$|u - u_h|_{H^1(D)} \leq C\sqrt{\frac{a_{\max}}{a_{\min}}}|u|_{H^2(D)}\, h \leq CC_2\sqrt{\frac{a_{\max}}{a_{\min}}}\|f\|_{L^2(D)}\, h.$$

References

[1] D. Braess. *Finite Elements.* Springer, Berlin-Heidelberg, 2003.

[2] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North Holland Publishing Company, New York-Amsterdam-Oxford, 1978.

[3] P. Bastian. *Scientific Computing with Partial Differential Equations.* Lecture Notes, Universität Heidelberg, 2017.