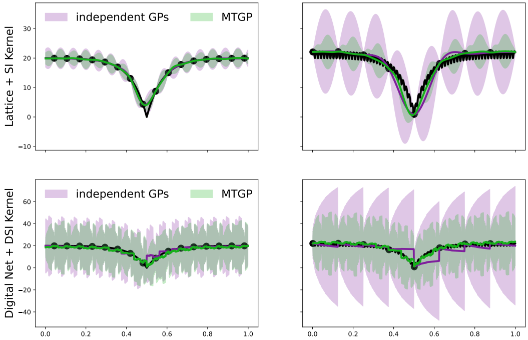
Quasi-Monte Carlo and Fast Multi-Task Gaussian Process Regression

Aleksei G Sorokin[1,2], Fred J Hickernell[1], Pieterjan M Robbe[2]
IIT, Department of Applied Math[1]. Sandia National Lab[2].

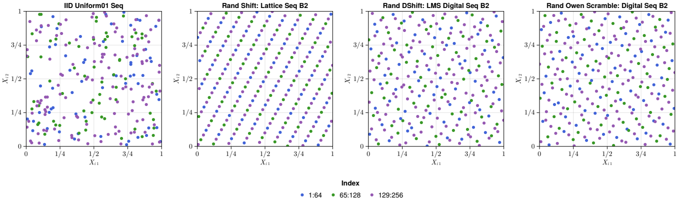


Caltech Lunch Group Seminar, 4/18/2025

Quasi-Monte Carlo Methods

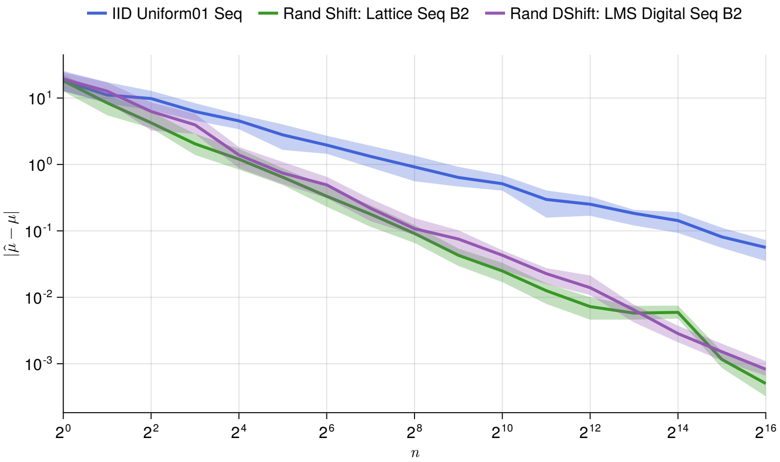
$$\mu = \mathbb{E}[g(\boldsymbol{T})] = \mathbb{E}[f(\boldsymbol{X})] \quad \approx \quad \frac{1}{N}\sum_{i=0}^{N-1} f(\boldsymbol{x}_i) = \widehat{\mu}, \qquad \boldsymbol{X} \sim \mathcal{U}[0,1]^d$$

- True mean μ is a function of true integrand g and true measure $\boldsymbol{T}$
- Transformation $\boldsymbol{T} \sim \boldsymbol{\psi}(\boldsymbol{X})$ makes $f = g \circ \boldsymbol{\psi}$,
  e.g., $\boldsymbol{T} \sim \mathcal{N}(\boldsymbol{m}, \Sigma = \mathsf{A}\mathsf{A}^T)$ then $\boldsymbol{\psi}(\boldsymbol{X}) = \mathsf{A}\boldsymbol{\Phi}^{-1}(\boldsymbol{X}) + \boldsymbol{m}$, standard normal CDF $\boldsymbol{\Phi}$
- Sampling nodes $\boldsymbol{x}_0, \boldsymbol{x}_1, \dots \in [0,1]^d$ chosen to be
  - IID $\to$ Monte Carlo (MC) $\to$ error $\mathcal{O}(N^{-1/2})$
  - Low discrepancy (LD) $\to$ Quasi-Monte Carlo (QMC) $\to$ error $\approx \mathcal{O}(N^{-1})$ or better [Dick and Pillichshammer, 2010, Dick et al., 2013]



Monte Carlo vs Quasi-Monte Carlo

MC error like $\mathcal{O}(N^{-1/2})$. QMC error like $\mathcal{O}(N^{-1+\delta})$ for $\delta > 0$ arbitrarily small



$d = 7$ dimensional function from [Keister, 1996]

Monte Carlo Stopping Criteria

How to choose N so that $|\mu - \widehat{\mu}| < \varepsilon$ for some user specified error tolerance $\varepsilon > 0$?
e.g., want to approximate the value of a financial option to within 1 penny
Approximations should hold with sufficiently high probability or with guarantees

Central Limit Theorem (CLT) Stopping Criteria for IID-Monte Carlo

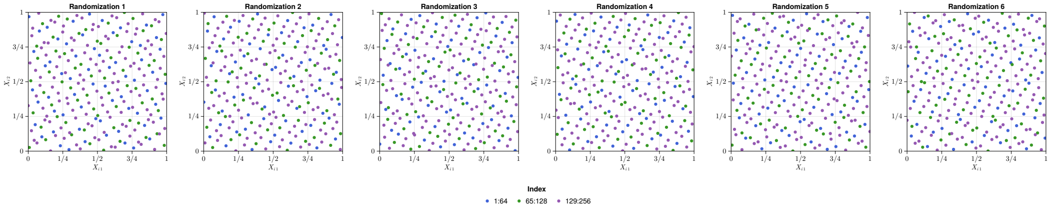
$$N \geqslant [2.58\sigma/\varepsilon]^2 \quad \longrightarrow \quad |\mu - \widehat{\mu}| < 2.58\sigma/\sqrt{N} \leqslant \varepsilon \text{ with probability } 99\%$$

- Only a heuristic algorithm as CLT is asymptotic in N
- May use initial sample to approximate σ
- [Hickernell et al., 2013] gives a guaranteed version of this two-step method for finite N using Berry-Esseen inequalities and assuming a bounded Kurtosis

Quasi-Monte Carlo Stopping Criteria

See [Owen, 2024] for a recent review of error estimation for QMC

1. Student-T intervals for R randomizations of a LD point set [L'Ecuyer et al., 2023]
2. Quickly track decay of coefficients for functions in cones [Hickernell et al., 2017]
   - Fourier coefficients using LD lattices [Jiménez Rugama and Hickernell, 2014]
   - Walsh coefficients using LD digital nets [Hickernell and Jiménez Rugama, 2014]
3. Fast Bayesian cubature for functions in cones [Rathinavel, 2019]
   - Gram matrices diagonalizable by FFT [Rathinavel and Hickernell, 2019]
   - Gram matrices diagonalizable by FWHT[1] [Rathinavel and Hickernell, 2022]



[1] Fast Walsh Hadamard Transform (FWHT) [Fino and Algazi, 1976]

Gaussian Process Regression (GPR)

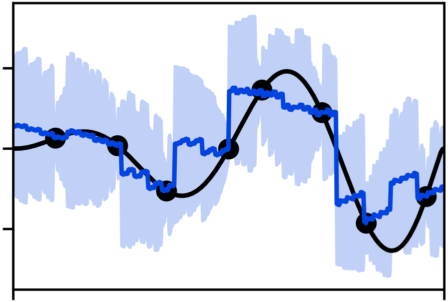
$$f \sim \mathrm{GP}(0, K)$$

Posterior mean and covariance

$$\mathbb{E}[f(\boldsymbol{x}) | \mathsf{X}, \boldsymbol{f}] = \boldsymbol{K}_{\mathsf{X}}(\boldsymbol{x}) \mathsf{K}^{-1} \boldsymbol{f}$$
$$\mathrm{Cov}[f(\boldsymbol{x}), f(\boldsymbol{x}') | \mathsf{X}, \boldsymbol{f}] = K(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{K}_{\mathsf{X}}(\boldsymbol{x})^T \mathsf{K}^{-1} \boldsymbol{K}_{\mathsf{X}}(\boldsymbol{x}')$$

- SPD $K : [0,1]^d \times [0,1]^d \to \mathbb{R}$
- Sampling locations $\mathsf{X} = \{\boldsymbol{x}_i\}_{i=0}^{N-1}$
- Sample values $\boldsymbol{f} = \{f(\boldsymbol{x}_i)\}_{i=0}^{N-1}$
- Kernel vector $\boldsymbol{K}_{\mathsf{X}}(\boldsymbol{x}) = \{K(\boldsymbol{x}, \boldsymbol{x}_i)\}_{i=0}^{N-1}$
- Gram matrix $\mathsf{K} = \{K(\boldsymbol{x}_i, \boldsymbol{x}_{i'})\}_{i,i'=0}^{N-1}$

Standard GPR cost is $\mathcal{O}(N^3)$



Bayesian Cubature

$$f \sim \mathrm{GP}(0,K)$$

$\mu = \mathbb{E}_{\boldsymbol{X}}[f(\boldsymbol{X})]$, $\boldsymbol{X} \sim \mathcal{U}[0,1]^d$ is Gaussian with posterior cubature mean and variance

$$\widehat{\mu} := \mathbb{E}_f[\mu|\mathsf{X},\boldsymbol{f}] = \widetilde{\boldsymbol{K}}_{\mathsf{X}}^T \mathsf{K}^{-1}\boldsymbol{f}$$

$$\widehat{\sigma}^2 := \mathrm{Var}_f[\mu|\mathsf{X},\boldsymbol{f}] = \widetilde{K} - \widetilde{\boldsymbol{K}}_{\mathsf{X}}^T \mathsf{K}^{-1}\widetilde{\boldsymbol{K}}_{\mathsf{X}}$$

$\widetilde{\boldsymbol{K}}_{\mathsf{X}} = \mathbb{E}_{\boldsymbol{X}}[\boldsymbol{K}_{\mathsf{X}}(\boldsymbol{X})]$ and $\widetilde{K} = \mathbb{E}_{(\boldsymbol{X},\boldsymbol{X}')}[K(\boldsymbol{X},\boldsymbol{X}')]$ for independent $\boldsymbol{X},\boldsymbol{X}' \sim \mathcal{U}[0,1]^d$

$\therefore\ |\mu - \widehat{\mu}| < 2.58\widehat{\sigma}$ with probability 99%

If $\widetilde{\boldsymbol{K}}_{\mathsf{X}}^T \mathsf{K}^{-1} = \mathbf{1}/N$ then $\widehat{\mu}$ is the sample average as in (Q)MC

Fast GPs Pairing LD Points with Special Kernels

1. **LD Lattices + Shift Invariant (SI) Kernels**
   - Give circulant Gram matrices $\mathsf{K} = \{K(\boldsymbol{x}_i, \boldsymbol{x}_{i'})\}_{i,i'=0}^{N-1}$

   $\therefore$ Eigendecomp $\mathsf{K} = \mathsf{V}\Lambda\overline{\mathsf{V}}$ where $\overline{\mathsf{V}}$ is the DFT matrix $\to$ FFT in $\mathcal{O}(N \log N)$
   - [Rathinavel and Hickernell, 2019]
2. **LD Digital Nets + Digitally Shift Invariant (DSI) Kernels**
   - Give Recursive Symmetric Block Toeplitz (RSBT) Gram matrices K

   $\therefore$ Eigendecomp $\mathsf{K} = \mathsf{V}\Lambda\overline{\mathsf{V}}$ where $\overline{\mathsf{V}}$ is the Hadamard matrix $\to$ FWHT in $\mathcal{O}(N \log N)$
   - [Rathinavel and Hickernell, 2022]

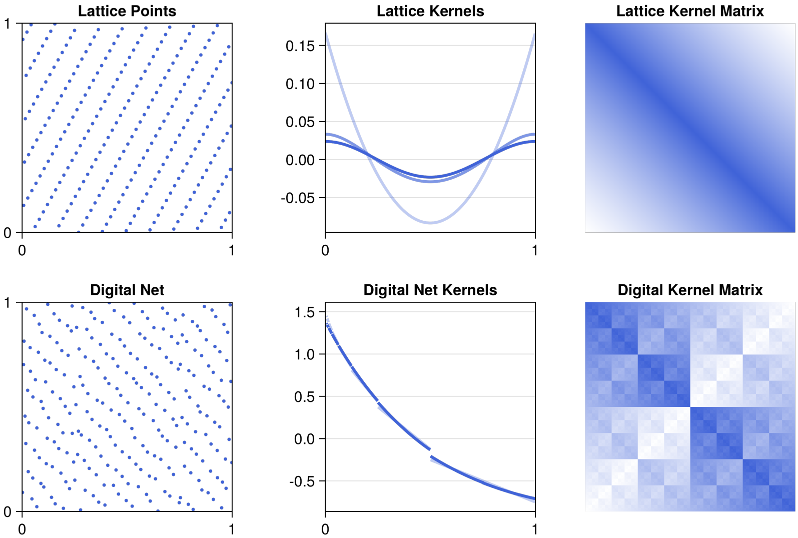
Let $\boldsymbol{v}_1 = \mathbf{1}/\sqrt{N}$ and $\boldsymbol{k}_1$ be the first columns of $\overline{\boldsymbol{V}}$ and K respectively:

$$\boldsymbol{\lambda} := \Lambda\mathbf{1} = \sqrt{N}\Lambda\overline{\boldsymbol{v}}_1 = \sqrt{N}\overline{\mathsf{V}}\mathsf{V}\Lambda\overline{\boldsymbol{v}}_1 = \sqrt{N}\overline{\mathsf{V}}\boldsymbol{k}_1$$

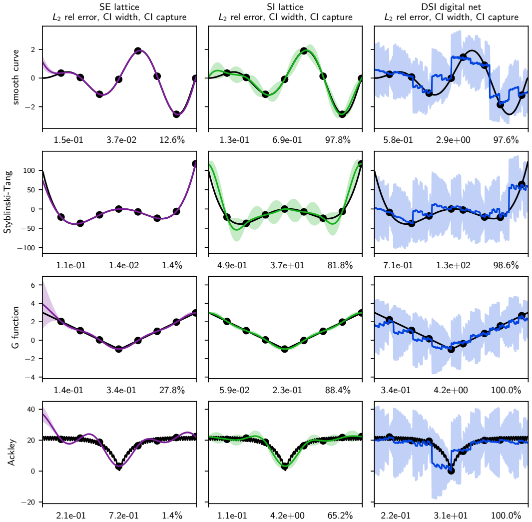
- $\mathsf{K}\boldsymbol{a}$, $\mathsf{K}^{-1}\boldsymbol{a}$, and $|\mathsf{K}|$ can all be computed in $\mathcal{O}(N \log N)$ computations
- Only requires evaluating and storing the first column of K

Originally developed in the context of fast Bayesian Cubature [Rathinavel, 2019]

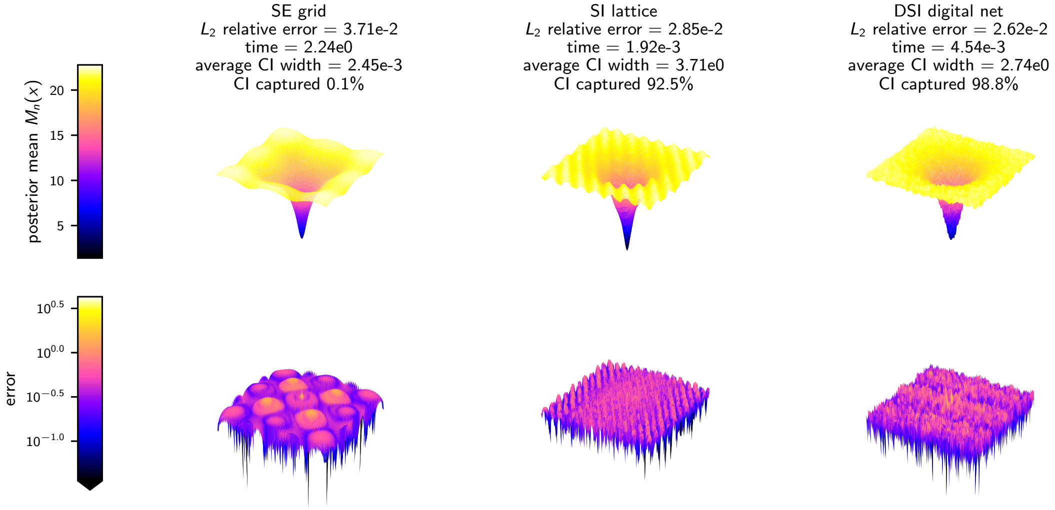
Lattices + SI K = Circulant K and Digital Nets + DSI K = RSBT K



Examples in One Dimension [Surjanovic and Bingham]



Ackley Function in Two Dimensions [Surjanovic and Bingham]

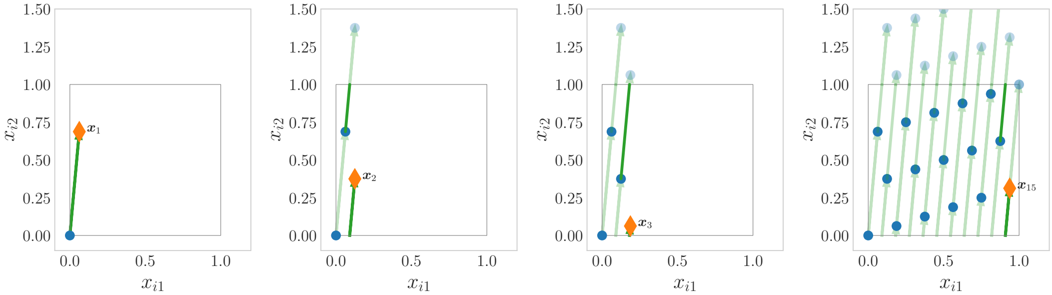


Unrandomized Lattice

Generating vector $\boldsymbol{g} \in \{1,\dots,N-1\}^d$ gives the **unrandomized lattice**

$$\boldsymbol{x}_i^{\text{lat}} = i\boldsymbol{g}/N \mod 1, \quad i = 0,\dots,N-1$$

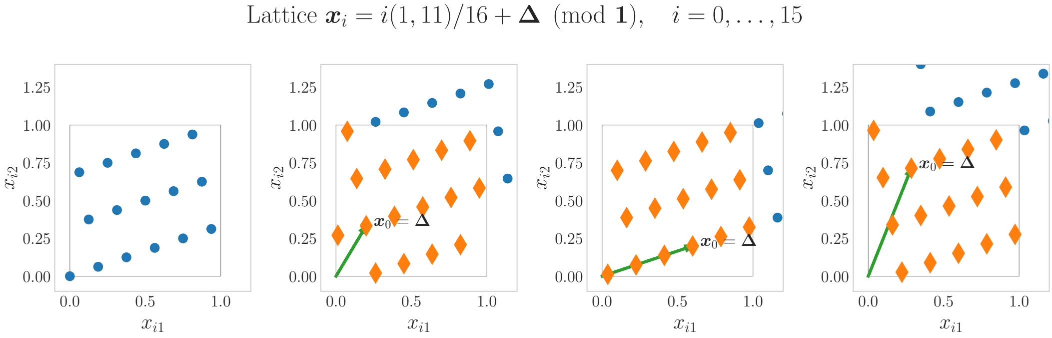
Lattice $\boldsymbol{x}_i^{\text{lat}} = i(1,11)/16 \pmod{\mathbf{1}}, \quad i = 0,\dots,15$



Shifted Lattice

Generating vector $\boldsymbol{g} \in \{1,\dots,N-1\}^d$ and shift $\boldsymbol{\Delta} \in [0,1]^d$ gives the **shifted lattice**

$$\boldsymbol{x}_i = (\boldsymbol{x}_i^{\text{lat}} + \boldsymbol{\Delta}) \mod 1 = (i\boldsymbol{g}/N + \boldsymbol{\Delta}) \mod 1, \quad i = 0,\dots,N-1$$



Shifted Lattices + SI Kenels = Circulant Matrices

Shift invariant (SI) kernels K may be written in terms of some $\widehat{K} : [0,1]^d \to \mathbb{R}$ s.t.

$$K(\boldsymbol{x},\boldsymbol{x}') = \widehat{K}(\boldsymbol{x} \ominus \boldsymbol{x}'), \qquad \boldsymbol{x} \ominus \boldsymbol{x}' = (\boldsymbol{x} - \boldsymbol{x}') \mod 1$$

For $\{\boldsymbol{x}_i\}_{i=0}^{N-1}$ a **shifted lattice**,

$$\begin{aligned}
\boldsymbol{x}_i \ominus \boldsymbol{x}_{i'} &= \left(\frac{i\boldsymbol{g}}{N} \ominus \boldsymbol{\Delta}\right) \ominus \left(\frac{i'\boldsymbol{g}}{N} \ominus \boldsymbol{\Delta}\right) \\
&= \left\{\frac{(i-i') \mod N}{N}\boldsymbol{g}\right\} = \boldsymbol{x}_0 \ominus \boldsymbol{x}_{(i'-i) \mod N} \\
\therefore\ \mathsf{K}_{i,i'} &= \mathsf{K}_{0,[(i'-i) \mod N]}
\end{aligned}$$

so K is circulant

Examples of Shift Invariant (SI) Kernels

For $f:[0,1]\to\mathbb{R}$, if $f^{(\alpha)}$ has an absoltey convergent Fourier series for some $\alpha\in\mathbb{N}_0$ and $f^{(\beta)}$ periodic for all $\beta\in\{0,\dots,\alpha-1\}$ then

$$f(x)=\sum_{k\in\mathbb{Z}}\widehat{f}(k)e^{2\pi \mathrm{i}kx}\quad\longrightarrow\quad f^{(\alpha)}(x)=\sum_{k\in\mathbb{Z}}\widehat{f^{(\alpha)}}(k)e^{2\pi\mathrm{i}kx},\quad \widehat{f^{(\alpha)}}(k)=(2\pi\mathrm{i}k)^\alpha\widehat{f}(k)$$

For B_i the i^{th} Bernoulli polynomial,

$$\mathring{K}_\alpha(x,x')=\sum_{k\in\mathbb{Z}\setminus\{0\}}\frac{e^{2\pi\mathrm{i}(x-x')}}{k^{2\alpha}}=\frac{(-1)^{\alpha+1}(2\pi)^{2\alpha}}{(2\alpha)!}B_{2\alpha}(x\ominus x')=:\mathring{K}_\alpha(x\ominus x')$$

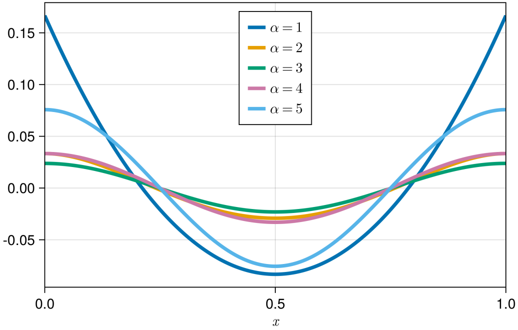
is the kernel of the Sobolev RKHS $\mathring{H}^\alpha$ with inner product

$$\langle f,g\rangle^{\circ}_\alpha=(-1)^\alpha(2\pi)^{2\alpha}\int_0^1 f^{(\alpha)}(x)g^{(\alpha)}(x)\mathrm{d}x$$

Let $\mathring{H}^{\boldsymbol{\alpha}}$ be the RKHS with SI kernel

$$\mathring{K}_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{x}')=\gamma\prod_{j=1}^d\left[1+\eta_j\mathring{K}_{\alpha_j}(x_j\ominus x_j')\right]$$

$$\mathring{K}_\alpha(x) = \frac{(-1)^{\alpha+1}(2\pi)^{2\alpha}}{(2\alpha)!}B_{2\alpha}(x)$$



If $f \in \mathring{H}^{\alpha\mathbf{1}}$ then QMC error $\left\lvert\int_{[0,1]^d} f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \frac{1}{N}\sum_{i=0}^{N-1} f(\boldsymbol{x}_i)\right\rvert$ is $\mathcal{O}(n^{-\alpha+\delta})$ for certain lattices $\{\boldsymbol{x}_i\}_{i=0}^{N-1}$ and certain weights $\boldsymbol{\eta}$ (with $\delta > 0$ arbitrarily small)

Digitwise Operations

Prime base $b \geqslant 2$ expansion of $x \in [0,1)$ is

$$x = .\mathsf{x}_1\mathsf{x}_2\mathsf{x}_3\cdots_b = \sum_{\ell \in \mathbb{N}} \mathsf{x}_\ell b^{-\ell}, \qquad \text{e.g.} \qquad .375 = .011_2,$$

with digitwise subtraction (digitwise exclusive or in base $b=2$)

$$x \ominus y := \sum_{\ell \in \mathbb{N}} ((\mathsf{x}_\ell - \mathsf{y}_\ell) \mod b) b^{-\ell}, \qquad \text{e.g.} \qquad .375 \ominus .625 = .011_2 \ominus .101_2 = .110_2 = .75.$$

Similarly for $k \in \mathbb{N}_0$

$$k = \cdots \mathsf{k}_2\mathsf{k}_1\mathsf{k}_0.0_b = \sum_{\ell \in \mathbb{N}_0} \mathsf{k}_\ell b^{\ell}, \qquad \text{e.g.} \qquad 5 = 101_2,$$

$$k \ominus h := \sum_{\ell \in \mathbb{N}_0} ((\mathsf{k}_\ell + \mathsf{h}_\ell) \mod b) b^{\ell}, \qquad \text{e.g.} \qquad 5 \ominus 6 = 101_2 \ominus 110_2 = 011_2 = 3.$$

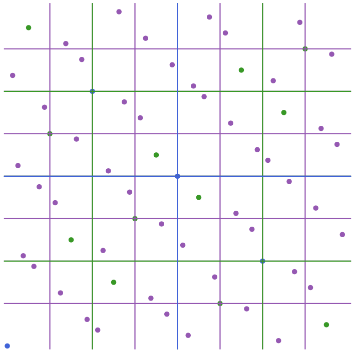
Apply elementwise to vectors

$$\boldsymbol{x} \ominus \boldsymbol{x}' = (x_1 \ominus x_1', \dots, x_d \ominus x_d'), \qquad \boldsymbol{k} \ominus \boldsymbol{k}' = (k_1 \ominus k_1', \dots, k_d \ominus k_d')$$

Digital Nets

$N = b^P$ and fixed $\boldsymbol{g}_{b^0}, \boldsymbol{g}_{b^1}, \dots, \boldsymbol{g}_{b^{P-1}} \in [0,1)^d$ give the **unrandomized digital net**

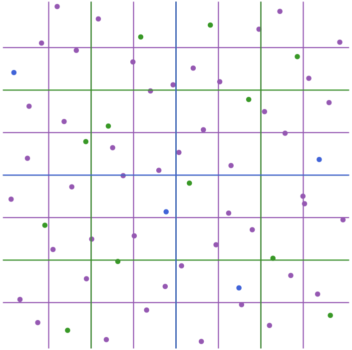
$$\boldsymbol{x}_i^{\text{dig}} = \bigominus_{\ell=0}^{P-1} \mathsf{i}_\ell \boldsymbol{g}_{b^\ell}, \qquad i = \sum_{\ell=0}^{P-1} \mathsf{i}_\ell \in \{0, \dots, b^P - 1\}$$



Digitally Shifted Digital Nets (and other Randomizations)

$\boldsymbol{g}_{b^0}, \boldsymbol{g}_{b^1}, \dots, \boldsymbol{g}_{b^{P-1}} \in [0,1)^d$ and digital shift (DS) $\boldsymbol{\Delta} \in [0,1)^d$ gives the **DS digital net**

$$\boldsymbol{x}_i = \boldsymbol{x}_i^{\mathrm{dig}} \ominus \boldsymbol{\Delta} = \bigominus_{\ell=0}^{P-1} \mathsf{i}_\ell \boldsymbol{g}_{2^\ell} \ominus \boldsymbol{\Delta}, \qquad i = \sum_{\ell=0}^{P-1} \mathsf{i}_\ell \in \{0,\dots,b^P-1\}$$



Linear Matrix Scrambling (LMS) and Owen Scrambling [Owen, 2003] also available

Digitally Shifted Digital Nets + DSI Kernels = RSBT Matrices

Digitally Shift Invariant (DSI) kernels K may be written as

$$K(\boldsymbol{x},\boldsymbol{x}') = \widehat{K}(\boldsymbol{x} \ominus \boldsymbol{x}') \qquad \text{for some} \qquad \widehat{K} : [0,1]^d \to \mathbb{R}$$

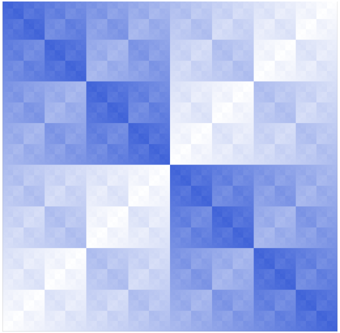
For $\{\boldsymbol{x}_i\}_{i=0}^{2^P-1}$ a $b=2$ **DS digital net** and $0 \leqslant p' < P$ and $i, i' \in \{0,\dots,2^{p'-1}-1\}$

- $\mathsf{K}_{i+2^{p'},i'+2^{p'}} = \mathsf{K}_{i,i'}$ since

$$\begin{aligned}&\boldsymbol{x}_{i+2^{p'}} \ominus \boldsymbol{x}_{i'+2^{p'}}\\ &= (\boldsymbol{x}_i \ominus \boldsymbol{g}_{2^{p'}}) \ominus (\boldsymbol{x}_{i'} \ominus \boldsymbol{g}_{2^{p'}})\\ &= \boldsymbol{x}_i \ominus \boldsymbol{x}_{i'}\end{aligned}$$

- $\mathsf{K}_{i+2^{p'},i'} = \mathsf{K}_{i,i'+2^{p'}}$ since

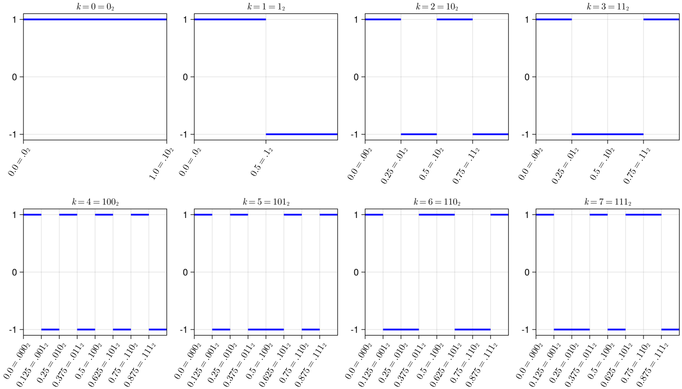
$$\begin{aligned}&\boldsymbol{x}_{i+2^{p'}} \ominus \boldsymbol{x}_{i'}\\ &= \boldsymbol{x}_i \ominus \boldsymbol{g}_{2^{p'}} \ominus \boldsymbol{x}_{i'}\\ &= \boldsymbol{x}_i \ominus \boldsymbol{x}_{i'+2^{p'}}\end{aligned}$$



so K is Recursive Symmetric Block Toeplitz (RSBT)

$b=2$ Walsh Functions $\mathrm{wal}_k(x)$ and Weight Functions $\mu_\alpha(k)$

$\mathrm{wal}_k(x) = (-1)^{\sum_{\ell \in \mathbb{N}_0} \mathsf{k}_\ell \mathsf{x}_{\ell+1}}$ with binary expansions $x = \sum_{\ell \in \mathbb{N}} \mathsf{x}_\ell b^{-\ell}$ and $k = \sum_{\ell \in \mathbb{N}_0} \mathsf{k}_\ell b^\ell$



$\mu_\alpha(k)$ sums the α largest indices of non-zero digits in the base b expansion of k, e.g., for $b=2$, $k = 13 = 1101_2$ has nonzero digit indices $(4,3,1)$ so

$$\mu_1(13) = 4, \quad \mu_2(13) = 4+3, \quad \mu_3(13) = \mu_4(13) = \cdots = 4+3+1$$

Decay of Walsh Coefficients and QMC

For $\alpha \geqslant 2$, let the $d=1$ Sobolev RKHS H^α have kernel K_α and inner product[2]

$$\langle f,g\rangle_\alpha = \sum_{\beta=1}^{\alpha-1}\int_0^1 f^{(\beta)}(x)\mathrm{d}x\int_0^1 g^{(\beta)}(x)\mathrm{d}x + \int_0^1 f^{(\alpha)}(x)g^{(\alpha)}(x)\mathrm{d}x$$

[Dick, 2008, 2009] show that $\exists C_{f,\alpha}>0$ s.t. if $f \in H^\alpha$ then

$$\left|\widehat{f}(k)\right| = \left|\int_0^1 f(x)\overline{\mathrm{wal}_k(x)}\mathrm{d}x\right| < C_{f,\alpha}b^{-\mu_\alpha(k)}$$

Let $H^{\boldsymbol{\alpha}}$ be the RKHS with kernel $K_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{x}') = \gamma\prod_{j=1}^d[1+\eta_j K_{\alpha_j}(x_j,x_j')]$
If $f \in H^{\alpha\mathbf{1}}$ then QMC error $\left|\int_{[0,1]^d} f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \frac{1}{N}\sum_{i=0}^{N-1} f(\boldsymbol{x}_i)\right|$ is $\mathcal{O}(n^{-\alpha+\delta})$ for certain (higher order) digital nets $\{\boldsymbol{x}_i\}_{i=0}^{N-1}$ and certain weights $\boldsymbol{\eta}$ (with $\delta>0$ arbitrarily small)

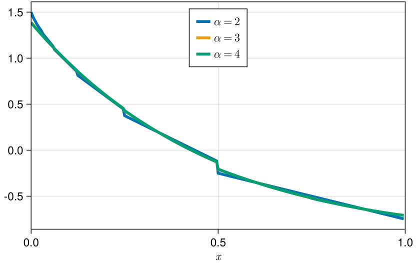
[2] For the $\alpha=1$ case see [Dick and Pillichshammer, 2005]

Examples of Digitally Shift Invariant (DSI) Kernels

⋆ K_α, the kernel of Sobolev RKHS H^α is **not DSI**, however $H^\alpha \subset \widetilde{H}^\alpha$ where $\widetilde{H}^\alpha$ is an RKHS with DSI kernel

$$\widetilde{K}_\alpha(x,y) = \sum_{k \in \mathbb{N}} \frac{\mathrm{wal}_k(x \ominus y)}{b^{\mu_\alpha(k)}} =: \widetilde{K}_\alpha(x \ominus y)$$

Below $\widetilde{K}_\alpha(x)$ with $b=2$ is shown. Discontinuities at $\{2^{-a} : a \in \mathbb{N}\}$ among others.



▸ More info on Walsh functions $\mathrm{wal}_k(x)$, weight functions $\mu_\alpha(k)$, and the decay of Walsh coefficients of smooth functions

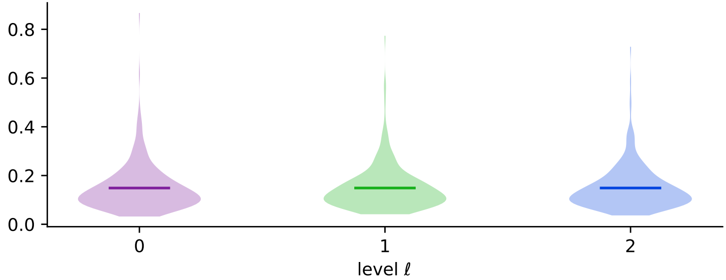
Multilevel (Multi-Task) Modeling

Given a multilevel simulation

$$f : \{1, \dots, L\} \times [0,1]^d \to \mathbb{R},$$

we want to model $f(L, \cdot)$, the true (maximum-fidelity) simulation.

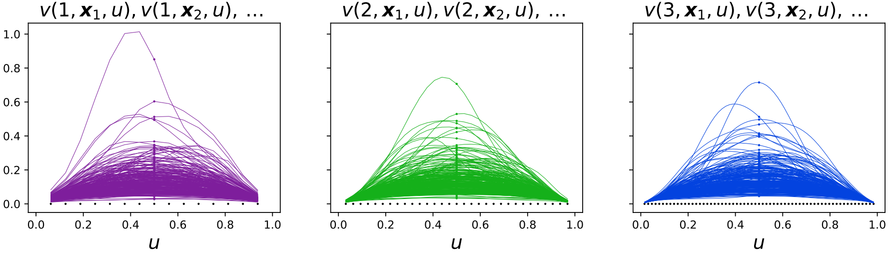
- $f(\ell, \boldsymbol{x})$ simulates at level $\ell \in \{1, \dots, L\}$ and with parameters $\boldsymbol{x} \in [0,1]^d$
- Higher levels are typically more expensive to evaluate
- $f(1, \cdot), f(2, \cdot), \dots, f(L, \cdot)$ are typically highly correlated



Numerical Solutions of PDEs with Random Coefficients

$$f(\ell,\boldsymbol{x}) = \mathcal{F}(v(\ell,\boldsymbol{x},\cdot))$$

- $v : \{1,\dots,L\} \times [0,1]^d \times \Omega$ is the numerical solution to the PDE
- $\boldsymbol{x}$ represent random coefficients, e.g. coefficients in a Karhunen-Loéve expansion
- ℓ controlls the fidelity of the numerical solver e.g. the mesh width is $2^{-\ell}$
- $\mathcal{F}$ is a (possibly non-linear) functional of the PDE solution, e.g.,
    - $\mathcal{F}(v(\ell,\boldsymbol{x},\cdot)) = \mathbb{E}[v(\ell,\boldsymbol{x},\boldsymbol{U})]$ where $\boldsymbol{U} \sim \mathcal{U}(\Omega)$, or
    - $\mathcal{F}(v(\ell,\boldsymbol{x},\cdot)) = v(\ell,\boldsymbol{x},u)$ for some $u \in \Omega$, e.g., $u = 1/2$ shown below



Multi-Task Gaussian Processes (MTGPs)

$$f \sim \mathrm{GP}(0, K)$$

$N = N_1 + \cdots + N_L$ sampling locations $\mathcal{D} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_L$ where $\mathcal{D}_\ell = \{(\ell, \boldsymbol{x}_{\ell i})\}_{i=1}^{N_\ell}$.
Posterior mean and covariance

$$\mathbb{E}[f(\ell, \boldsymbol{x})] = \boldsymbol{K}^T(\ell, \boldsymbol{x})\mathsf{K}^{-1}\boldsymbol{f}$$
$$\mathbb{C}[f(\ell, \boldsymbol{x}), f(\ell', \boldsymbol{x}')] = K((\ell, \boldsymbol{x}), (\ell', \boldsymbol{x}')) - \boldsymbol{K}^T(\ell', \boldsymbol{x}')\mathsf{K}^{-1}\boldsymbol{K}(\ell', \boldsymbol{x}')$$

- $\boldsymbol{K}(\ell, \boldsymbol{x}) = K(\mathcal{D}, (\ell, \boldsymbol{x}))$ and $\boldsymbol{f} = f(\mathcal{D})$ are length N vectors
- $\mathsf{K} = K(\mathcal{D}, \mathcal{D}^T)$ is the $N \times N$ Gram matrix

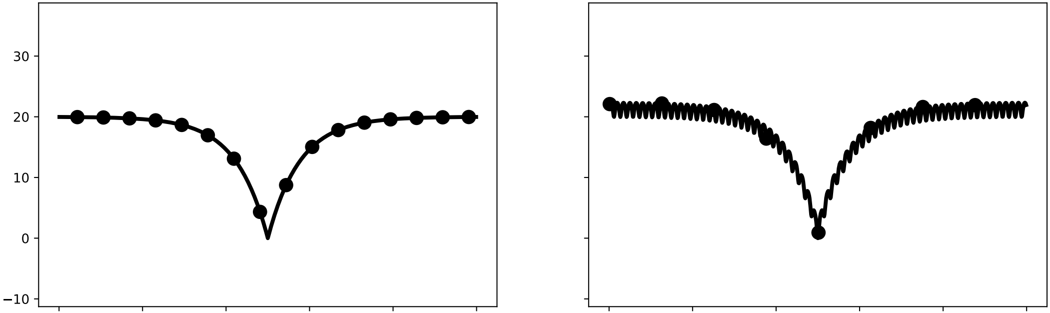
Kernel K depends on hyperparameters $\boldsymbol{\theta}$ e.g. global scale, lengthscale, etc.
Hyperparameters $\boldsymbol{\theta}$ often chosen to minimize negative marginal log likelihood (NMLL)

$$\mathrm{NMLL} \propto \boldsymbol{f}^T\mathsf{K}^{-1}\boldsymbol{f} + \log|\mathsf{K}| + \log(2\pi)N$$

$\therefore$ MTGP fitting requires computing $\mathsf{K}^{-1}\boldsymbol{f}$ and $\log|\mathsf{K}| \implies$ standard cost $\mathcal{O}(N^3)$

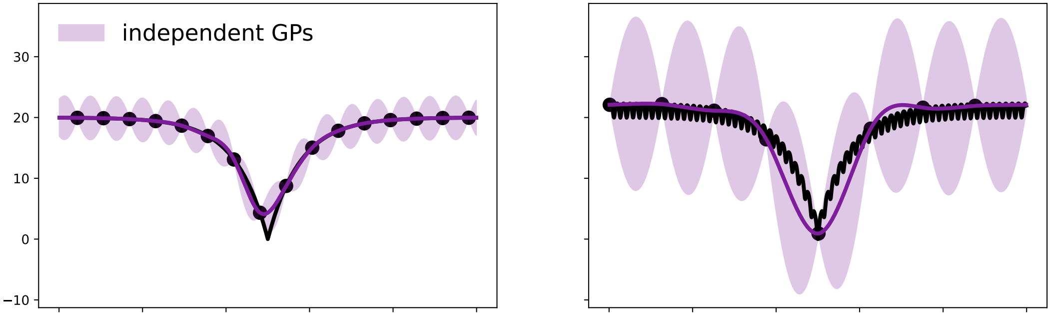
Independent GPs vs a Multi-Task GP Example

Low Fidelity Left, High Fidelity Right



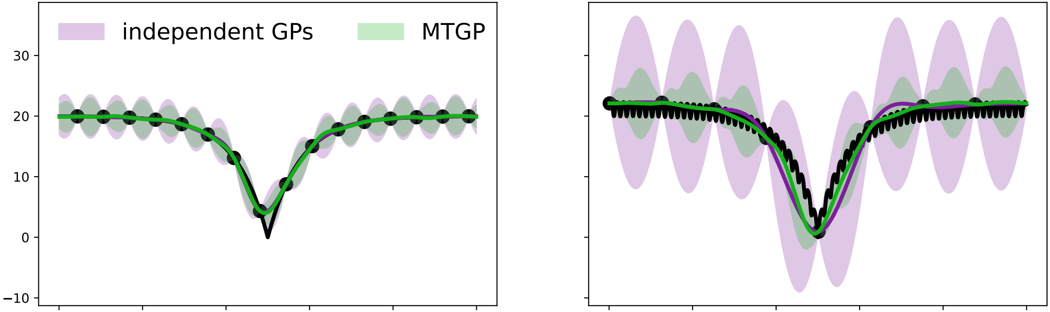
Independent GPs vs a Multi-Task GP Example

Low Fidelity Left, High Fidelity Right



Independent GPs vs a Multi-Task GP Example

Low Fidelity Left, High Fidelity Right



Multilevel Monte Carlo (MLMC) with MTGPs

Quantity of interest

$$\widetilde{f}_\ell = \int f(\ell, \boldsymbol{x})$$

In our Bayesian setting, the posterior cubature mean and covariance are

$$\mathbb{E}\left[\widetilde{f}_\ell\right] = \widetilde{\boldsymbol{K}}_\ell^T \mathsf{K}^{-1} \boldsymbol{f}$$

$$\mathbb{C}\left[\widetilde{f}_\ell, \widetilde{f}_{\ell'}\right] = \widetilde{K}_{\ell\ell'} - \widetilde{\boldsymbol{K}}_\ell^T \mathsf{K}^{-1} \widetilde{\boldsymbol{K}}_{\ell'}$$

- Integrals understood to be over $[0,1]^d$ with respect to $\boldsymbol{x}$
- $\widetilde{\boldsymbol{K}}_\ell = \int \boldsymbol{K}(\ell, \boldsymbol{x})$ a length N vector
- Scalar $\widetilde{K}_{\ell\ell'} = \int\int K((\ell, \boldsymbol{x}), (\ell', \boldsymbol{x}')) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{x}'$

$\therefore$ Quantity of interest

$$\widetilde{f}_L \sim \mathcal{N}\left(\widetilde{\boldsymbol{K}}_L^T \mathsf{K}^{-1} \boldsymbol{f}, \widetilde{K}_{LL} - \widetilde{\boldsymbol{K}}_L^T \mathsf{K}^{-1} \widetilde{\boldsymbol{K}}_L\right)$$

Product Kernels for Multi-Task GPs

Common to assume

$$K((\ell,\boldsymbol{x}),(\ell',\boldsymbol{x}')) = R(\ell,\ell')Q(\boldsymbol{x},\boldsymbol{x}')$$

- $R : \{1,\dots,L\} \times \{1,\dots,L\} \to \mathbb{R}$ an SPD kernel over levels e.g.

$$\mathsf{R} = \{R(\ell,\ell')\}_{\ell,\ell'=1}^{L} = \mathsf{B}\mathsf{B}^T + \mathrm{diag}(\boldsymbol{\nu}), \quad \boldsymbol{\nu} \in \mathbb{R}_+^L, \quad \mathsf{B} \in \mathbb{R}^{L\times r}, \quad \text{rank } r \leqslant L$$

- $Q : [0,1]^d \times [0,1]^d \to \mathbb{R}$ an SPD kernel over parameters

$$\mathsf{K} = \begin{pmatrix} \mathsf{K}_{11} & \cdots & \mathsf{K}_{1L} \\ \vdots & \ddots & \vdots \\ \overline{\mathsf{K}_{1L}} & \cdots & \mathsf{K}_{LL} \end{pmatrix} = \begin{pmatrix} R_{11}\mathsf{Q}_{11} & \cdots & R_{1L}\mathsf{Q}_{1L} \\ \vdots & \ddots & \vdots \\ \overline{R_{1L}\mathsf{Q}_{1L}} & \cdots & R_{LL}\mathsf{Q}_{LL} \end{pmatrix}$$

- $R_{\ell\ell'} = R(\ell,\ell')$ is a scalar
- $\mathsf{Q}_{\ell\ell'} = Q(\mathcal{D}_\ell, \mathcal{D}_{\ell'}^T)$ is an $N_\ell \times N_{\ell'}$ Gram matrix

Fast MTGPs

$$\mathsf{K} = \begin{pmatrix} R_{11}\mathsf{Q}_{11} & \cdots & R_{1L}\mathsf{Q}_{1L} \\ \vdots & \ddots & \vdots \\ \overline{R_{1L}\mathsf{Q}_{1L}} & \cdots & R_{LL}\mathsf{Q}_{LL} \end{pmatrix}$$

Idea: Force "nice" structure in $\mathsf{Q}_{\ell\ell'}$ through special pairings of $\mathsf{X}_\ell = \{\boldsymbol{x}_{\ell i}\}_{i=1}^{N_\ell}$ and Q

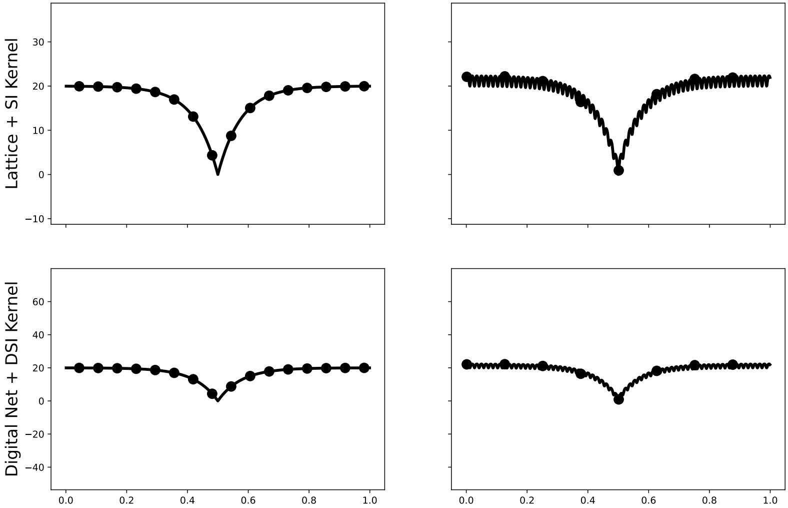
1. X_ℓ a lattice and Q a shift invariant (SI) kernel $\implies$ $\mathsf{Q}_{\ell\ell'}$ block circulant
2. X_ℓ a (base 2) digital net and Q a digitally SI (DSI) kernel $\implies$ $\mathsf{Q}_{\ell\ell}$ block RSBT

Technicalities

- Lattices and digital nets require sample sizes $N_\ell = 2^{m_\ell}$
- Lattices $\mathsf{X}_1,\dots,\mathsf{X}_L$: same generating vector, possibly different random shifts
- Circulant matrices diagonalizable by FFT
- Digital nets $\mathsf{X}_1,\dots,\mathsf{X}_L$: same generating matrices, possibly different digital shifts
- RSBT matrices diagonalizable by FWHT

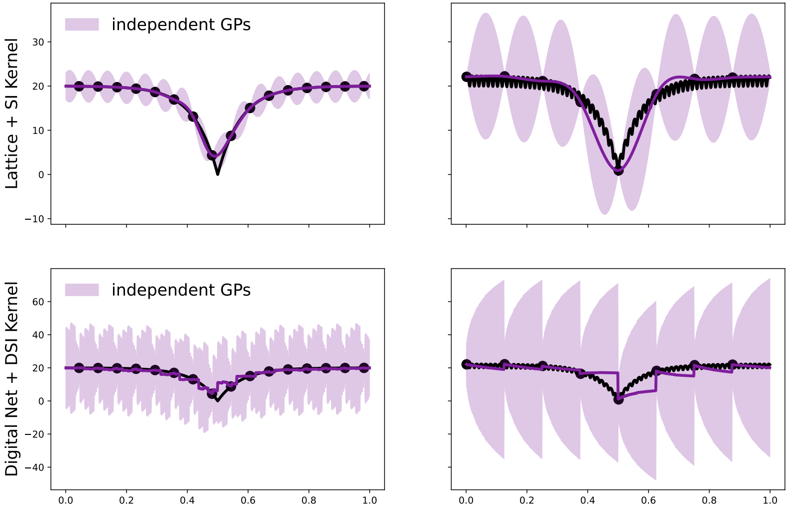
Independent Fast GPs vs Fast MTGPs Example

Low Fidelity Left, High Fidelity Right



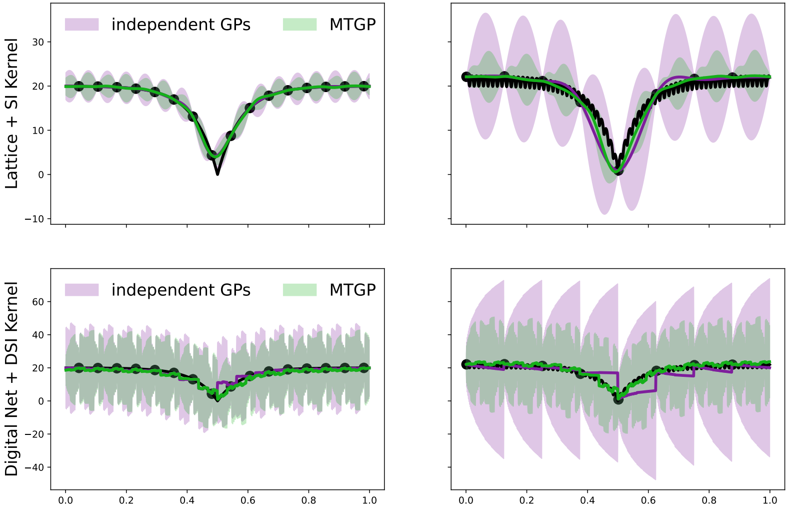
Independent Fast GPs vs Fast MTGPs Example

Low Fidelity Left, High Fidelity Right



Independent Fast GPs vs Fast MTGPs Example

Low Fidelity Left, High Fidelity Right



Fast MTGPs Continued

$$\mathsf{K}_{\ell\ell'} = R_{\ell\ell'}\mathsf{Q}_{\ell\ell'} = \mathsf{V}_{m_\ell}\Sigma_{\ell\ell'}\overline{\mathsf{V}_{m_{\ell'}}}$$

- $\overline{\mathsf{V}_m}$ a $2^m \times 2^m$ *fast transform matrix*
  1. Lattice X_ℓ with SI Q makes $\overline{\mathsf{V}_{m_\ell}}$ the Fast Fourier Transform
  2. Digital Net X_ℓ with DSI Q makes $\overline{\mathsf{V}_{m_\ell}}$ the Fast Walsh Hadamard Transform
- $\mathsf{V}_m\boldsymbol{a}$ and $\overline{\mathsf{V}_m}\boldsymbol{a}$ both cost only $\mathcal{O}(m2^m)$ to compute
- The first column of $\overline{\mathsf{V}_m}$ is $\mathbf{1}_m/\sqrt{2^m}$
- $\Sigma_{\ell\ell'}$ a diagonal block matrix characterized by

$$\boldsymbol{\sigma}_{\ell\ell'} = \Sigma_{\ell\ell'}\mathbf{1}_{m_{\ell'}} = \sqrt{2^{m_{\ell'}}}\overline{\mathsf{V}_{m_\ell}}\boldsymbol{k}_{\ell\ell',1}$$

where $\boldsymbol{k}_{\ell\ell',1}$ is the first column of $\mathsf{K}_{\ell\ell'}$ and we assume $m_\ell \geqslant m_{\ell'}$

Fast MTGPs NMLL

$$\mathsf{K} = \begin{pmatrix} \mathsf{V}_{m_1} & & \\ & \ddots & \\ & & \mathsf{V}_{m_L} \end{pmatrix} \begin{pmatrix} \Sigma_{11} & \cdots & \Sigma_{1L} \\ \vdots & \ddots & \vdots \\ \overline{\Sigma_{1L}} & \cdots & \Sigma_{LL} \end{pmatrix} \begin{pmatrix} \overline{\mathsf{V}_{m_1}} & & \\ & \ddots & \\ & & \overline{\mathsf{V}_{m_L}} \end{pmatrix} =: \mathsf{V}\Sigma\overline{\mathsf{V}}$$

$$\mathrm{NMLL} \propto \widehat{\boldsymbol{f}}^T \Sigma^{-1} \widehat{\boldsymbol{f}} + \log|\Sigma| + \log(2\pi) N, \qquad \widehat{\boldsymbol{f}} = \overline{\mathsf{V}} \boldsymbol{f}$$

$\therefore$ Fast MTGP fitting requires computing $\Sigma^{-1}\widehat{\boldsymbol{f}}$ and $\log|\Sigma|$

For example, if $N_1 = 8$, $N_2 = 4$, and $N_3 = 2$ then Σ has the following structure

$$\Sigma = \left[\begin{array}{cccccccc|cccc|cc}
\cdot & & & & & & & & \cdot & & & & \cdot & \\
 & \cdot & & & & & & & & \cdot & & & & \cdot \\
 & & \cdot & & & & & & & & \cdot & & \cdot & \\
 & & & \cdot & & & & & & & & \cdot & & \cdot \\
 & & & & \cdot & & & & \cdot & & & & \cdot & \\
 & & & & & \cdot & & & & \cdot & & & & \cdot \\
 & & & & & & \cdot & & & & \cdot & & \cdot & \\
 & & & & & & & \cdot & & & & \cdot & & \cdot \\
\hline
\cdot & & & & \cdot & & & & \cdot & & & & \cdot & \\
 & \cdot & & & & \cdot & & & & \cdot & & & & \cdot \\
 & & \cdot & & & & \cdot & & & & \cdot & & \cdot & \\
 & & & \cdot & & & & \cdot & & & & \cdot & & \cdot \\
\hline
\cdot & & \cdot & & \cdot & & \cdot & & \cdot & & \cdot & & \cdot & \\
 & \cdot & & \cdot & & \cdot & & \cdot & & \cdot & & \cdot & & \cdot
\end{array}\right]$$

Fast MTGPs Storage and Costs

- Assume evaluating $f(\ell,\boldsymbol{x})$ costs C_ℓ for any $\boldsymbol{x} \in [0,1]^d$
- Assume evaluating $\mathsf{K}((\ell,\boldsymbol{x}),(\ell',\boldsymbol{x}'))$ costs $\mathcal{O}(d)$
- Assume $m_1 \geqslant \dots \geqslant m_L$ i.e. less samples on higher levels (or reorder levels)

$$\mathrm{NMLL} \propto \widehat{\boldsymbol{f}}^T \Sigma^{-1} \widehat{\boldsymbol{f}} + \log|\Sigma| + \log(2\pi) N$$

- Evaluate $\boldsymbol{f} = f(\mathcal{D})$ at cost $\mathcal{O}\left(\sum_{\ell=1}^L C_\ell 2^{m_\ell}\right)$
- Evaluate $\widehat{\boldsymbol{f}} = \overline{\mathsf{V}} \boldsymbol{f}$ at cost $\mathcal{O}\left(\sum_{\ell=1}^L m_\ell 2^{m_\ell}\right)$
- Evaluate only first columns of $\mathsf{K}_{\ell\ell'}$ at total cost $\mathcal{O}\left(d \sum_{\ell=1}^L (L-\ell+1) 2^{m_\ell}\right)$
- Evaluate Σ at cost $\mathcal{O}\left(\sum_{\ell=1}^L (L-\ell+1) m_\ell 2^{m_\ell}\right)$
- Store Σ in $\mathcal{O}\left(\sum_{\ell=1}^L (L-\ell+1) 2^{m_\ell}\right)$ (possibly complex) floats
- Evaluate $\Sigma^{-1}\widehat{\boldsymbol{f}}$ and $\log|\Sigma|$ at cost $\mathcal{O}\left(\sum_{\ell'=1}^L 2^{-m_{\ell'}} \left[\sum_{\ell=1}^{\ell'-1} 2^{m_\ell}\right]^2\right)$ algorithm

Python Software

`QMCPy` [github.com/QMCSoftware/QMCSoftware]

- Low discrepancy sequences including lattices and digital nets
- Measures with automatic transforms to integrals over $[0,1]^d$
- Adaptive stopping criteria for IID Monte Carlo and QMC

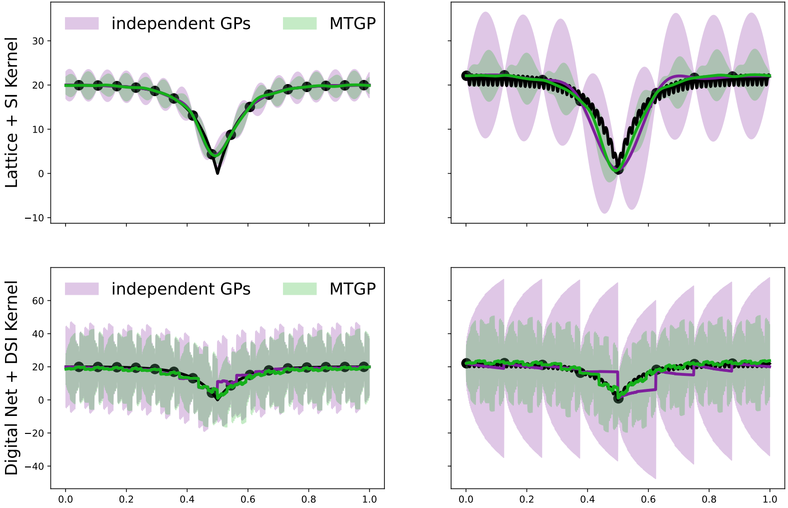
`FastGPs` [alegresor.github.io/fastgps]

- Fast GPs
- Fast MTGPs
- Uses `PyTorch` for efficient optimizing and GPU compatibility
- Efficient MTGP updates when increasing sample sizes via caching

`MLQMCPy` [github.com/PieterjanRobbe/mlqmcpy]

- Multilevel IID Monte Carlo algorithms
- Multilevel QMC algorithms

Thank you for listening!



References I

Josef Dick. Walsh spaces containing smooth functions and quasi–monte carlo rules of arbitrary high order. *SIAM Journal on Numerical Analysis*, 46(3):1519–1553, 2008.

Josef Dick. The decay of the walsh coefficients of smooth functions. *Bulletin of the Australian Mathematical Society*, 80(3):430–453, 2009.

Josef Dick and Friedrich Pillichshammer. Multivariate integration in weighted hilbert spaces based on walsh functions and weighted sobolev spaces. *Journal of Complexity*, 21(2):149–195, 2005.

Josef Dick and Friedrich Pillichshammer. *Digital nets and sequences: discrepancy theory and quasi–Monte Carlo integration*. Cambridge University Press, 2010.

Josef Dick, Frances Y Kuo, and Ian H Sloan. High-dimensional integration: the quasi-monte carlo way. *Acta Numerica*, 22:133–288, 2013.

Nathan Jacob Fine. On the walsh functions. *Transactions of the American Mathematical Society*, 65(3):372–414, 1949.

References II

Fino and Algazi. Unified matrix treatment of the fast walsh-hadamard transform. *IEEE Transactions on Computers*, 100(11):1142–1146, 1976.

Fred J. Hickernell and Lluís Antoni Jiménez Rugama. Reliable adaptive cubature using digital sequences, 2014.

Fred J Hickernell, Lan Jiang, Yuewei Liu, and Art B Owen. Guaranteed conservative fixed width confidence intervals via monte carlo sampling. In *Monte Carlo and Quasi-Monte Carlo Methods 2012*, pages 105–128. Springer, 2013.

Fred J. Hickernell, Lluís Antoni Jiménez Rugama, and Da Li. Adaptive quasi-monte carlo methods for cubature, 2017.

Lluís Antoni Jiménez Rugama and Fred J. Hickernell. Adaptive multidimensional integration based on rank-1 lattices, 2014.

B. D. Keister. Multidimensional quadrature algorithms. *Computers in Physics*, 10: 119–122, 1996. doi: 10.1063/1.168565.

References III

Gerhard Larcher, Harald Niederreiter, and Wolfgang Ch Schmid. Digital nets and sequences constructed over finite rings and their application to quasi-monte carlo integration. *Monatshefte für Mathematik*, 121:231–253, 1996.

Pierre L'Ecuyer, Marvin K Nakayama, Art B Owen, and Bruno Tuffin. Confidence intervals for randomized quasi-monte carlo estimators. In *2023 Winter Simulation Conference (WSC)*, pages 445–456. IEEE, 2023.

Art B Owen. Variance and discrepancy with alternative scramblings. *ACM Transactions of Modeling and Computer Simulation*, 13(4), 2003.

Art B Owen. Error estimation for quasi-monte carlo. *arXiv preprint arXiv:2501.00150*, 2024.

Jagadeeswaran Rathinavel. *Fast automatic Bayesian cubature using matching kernels and designs*. Illinois Institute of Technology, 2019.

References IV

Jagadeeswaran Rathinavel and Fred J. Hickernell. Fast automatic bayesian cubature using lattice sampling. *Statistics and Computing*, 29(6):1215–1229, Sep 2019. ISSN 1573-1375. doi: 10.1007/s11222-019-09895-9. URL http://dx.doi.org/10.1007/s11222-019-09895-9.

Jagadeeswaran Rathinavel and Fred J Hickernell. Fast automatic bayesian cubature using sobol' sampling. In *Advances in Modeling and Simulation: Festschrift for Pierre L'Ecuyer*, pages 301–318. Springer, 2022.

S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved April 9, 2025, from http://www.sfu.ca/~ssurjano.

Joseph L Walsh. A closed set of normal orthogonal functions. *American Journal of Mathematics*, 45(1):5–24, 1923.

Walsh Functions

Introduced for base $b = 2$ in [Walsh, 1923] with important results in [Fine, 1949].
Generalized to finite abelian group with a bijection in [Larcher et al., 1996].

For $k \in \mathbb{N}_0$ with $\mathsf{k} = (\mathsf{k}_0, \mathsf{k}_1, \dots)$ and $x \in [0,1)$ with $\mathsf{x} = (\mathsf{x}_1, \mathsf{x}_2, \dots)$,

$$\mathrm{wal}_k(x) = e^{2\pi \mathrm{i}/b \sum_{\ell=0}^{\infty} \mathsf{k}_\ell \mathsf{x}_{\ell+1}} = e^{2\pi \mathrm{i}/b \mathsf{k}.\mathsf{x}}$$

$$\text{e.g. for } b = 2, \qquad \mathrm{wal}_6(.75) = (-1)^{(0,1,1).(1,1,0)} = -1.$$

For any fixed b, $\{\mathrm{wal}_k : k \in \mathbb{N}_0\}$ is a complete orthonormal system in $\mathcal{L}_2([0,1))$.
Notice similarity to complex exponential basis $\left\{e^{2\pi \mathrm{i} kx} : k \in \mathbb{Z}\right\}$ for Fourier series.

Walsh Function Properties

For any $x, y \in [0,1)$ and $k, h \in \mathbb{N}_0$ and $f \in \mathcal{L}_2([0,1))$

1. $\mathrm{wal}_k(x)\mathrm{wal}_h(x) = \mathrm{wal}_{k \ominus h}(x)$ and $\mathrm{wal}_k(x)\mathrm{wal}_k(y) = \mathrm{wal}_k(x \ominus y)$
2. $$\int_0^1 \mathrm{wal}_k(x)\mathrm{d}x = \begin{cases} 1, & k = 0 \\ 0, & k > 0 \end{cases}$$
3. $$\int_0^1 f(\sigma)\mathrm{d}\sigma = \int_0^1 f(x \ominus \sigma)\mathrm{d}\sigma$$
4. $$\sum_{k=0}^{b^a-1} \mathrm{wal}_k(x) = \begin{cases} b^a, & a < \mathcal{I}(x) - 1 \\ 0, & \text{otherwise} \end{cases}$$

   where $\mathcal{I}(x) = -\lfloor \log_b(x) \rfloor$ is the index first non-zero digit in the base b expansion e.g. with $b = 2$ then $\mathcal{I}(.375) = \mathcal{I}(.011_2) = 2$.

Weight Function

Write $k \in \mathbb{N}$ as

$$k = \sum_{\ell=1}^{\#k} \mathsf{k}_{a_\ell} b^{a_\ell}$$

with $a_1 > \cdots > a_{\#k} \geqslant 0$ and $k_{a_\ell} \in \{1,\dots,b-1\}$.
Weight function for $\alpha \in \mathbb{N}_0$ has

$$\mu_\alpha(k) = \sum_{\ell=1}^{\min(\alpha,\#k)} (a_\ell + 1)$$

with $\mu_0(k) = \mu_\alpha(0) = 0$. μ sums indices of non-zero digits. For example, with $b = 2$

$$k = 13 = 1101_2 \qquad \text{has} \qquad (a_1, a_2, a_3) = (3, 2, 0)$$

$$\mu_1(k) = (3+1), \quad \mu_2(k) = (3+1)+(2+1), \quad \mu_3(k) = (3+1)+(2+1)+(0+1) = \mu_4(k) = \dots.$$

Walsh Series of Smooth Functions

For $\alpha \geqslant 2$ the Sobolev RKHS H^α with inner product

$$\langle f, g\rangle_\alpha = \sum_{\beta=1}^{\alpha-1} \int_0^1 f^{(\beta)}(x)\mathrm{d}x \int_0^1 g^{(\beta)}(x)\mathrm{d}x + \int_0^1 f^{(\alpha)}(x) g^{(\alpha)}(x)\mathrm{d}x$$

has kernel

$$K_\alpha(x,x') = \sum_{\beta=1}^{\alpha-1} \frac{B_\beta(x)B_\beta(x')}{(\beta!)^2} + \overbrace{(-1)^{\alpha+1}\frac{B_{2\alpha}(\{x-x'\})}{(2\alpha)!}}^{\mathring{K}_\alpha((x-x') \mod 1)/(2\pi)^{2\alpha}}.$$

[Dick, 2008, 2009] show that if $f \in H^\alpha$ then for $\widehat{f}(k) = \int_0^1 f(x)\overline{\mathrm{wal}_k(x)}\mathrm{d}x$ we have

$$\sup_{k \in \mathbb{N}_0} \left|\widehat{f}(k)\right| b^{\mu_\alpha(k)} < \infty \qquad \text{i.e.} \qquad \exists C_{f,\alpha} > 0 \quad \text{s.t.} \quad \left|\widehat{f}(k)\right| \leqslant \frac{C_{f,\alpha}}{b^{\mu_\alpha(k)}}.$$

For the $\alpha = 1$ case see [Dick and Pillichshammer, 2005].

Return to slide on **Fast MTGPs Storage and Costs**

Algorithm 1 Inverse and Determinant of Σ

Require: Σ diagonal block matrix with $m_1 \geqslant \cdots \geqslant m_L$.

$\mathsf{A} \leftarrow \Sigma_{11}^{-1}$ diagonal

$\rho \leftarrow |\Sigma_{11}|$

$\ell \leftarrow 2$

while $\ell \leqslant L$ **do**

$\mathsf{D} \leftarrow \Sigma_{\ell\ell}$ diagonal

$\mathsf{C} \leftarrow \Sigma_{1:\ell-1,\ell}$ diagonal blocks

$\mathsf{S}_\ell \leftarrow \mathsf{D} - \overline{\mathsf{C}}\mathsf{A}^{-1}\mathsf{C}$ diagonal Schur complement

$$\mathsf{A} \leftarrow \begin{pmatrix} \mathsf{A}^{-1} + \mathsf{A}^{-1}\mathsf{C}\mathsf{S}_\ell^{-1}\overline{\mathsf{C}}\mathsf{A}^{-1} & -\mathsf{A}^{-1}\mathsf{C}\mathsf{S}_\ell^{-1} \\ -\mathsf{S}_\ell^{-1}\overline{\mathsf{C}}\mathsf{A}^{-1} & \mathsf{S}_\ell^{-1} \end{pmatrix}$$

$\rho \leftarrow \rho|\mathsf{S}_\ell|$

$\ell \leftarrow \ell + 1$

end while

return $\mathsf{A} = \Sigma^{-1} \quad \text{and} \quad \rho = |\Sigma|$