Fast Gaussian Process Regression for Smooth Functions

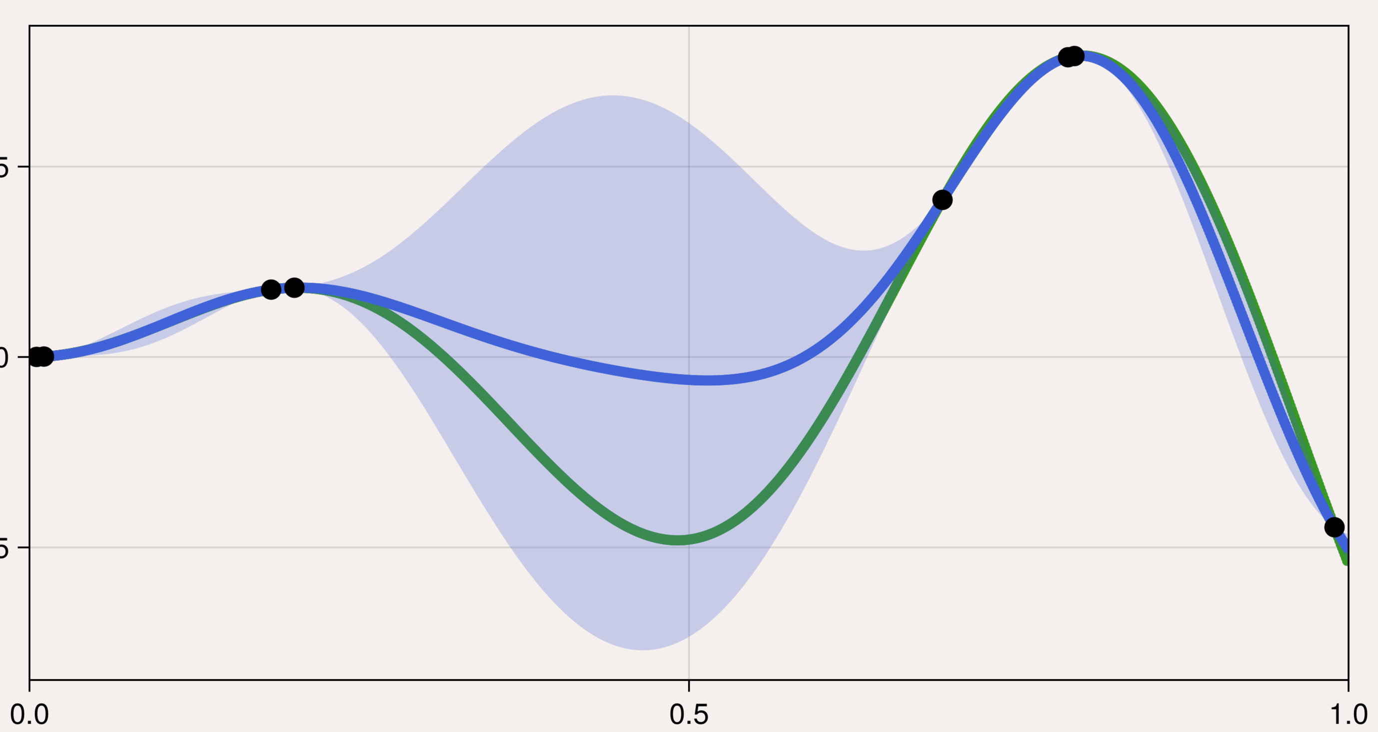
Aleksei G Sorokin & Fred J Hickernell

Illinois Institute of Technology

Quickly Approximate a Function With an Error Aware Model

Why Gaussian Proess Regression (GPR)?

- Encode simulation assumptions into model via kernel e.g. smoothness
- Gives distribution over functions i.e. quantifies uncertainty in predictions



- **Green line:** unknown function f modeled by GPR
- **Black points:** data where we have sampled f, used to fit the GPR
- **Blue line:** GPR approximation (posterior mean)
- **Blue shaded region:** GPR error (95% confidence interval for output)

Overview

Problem: GPR typically costs $\mathcal{O}(n^3)$ to fit to n points
Solution: Structure nodes and kernel for reduced cost $\mathcal{O}(n \log n)$

- Requires control over the design of experiments
- Requires kernel assumptions e.g. periodicity or digital shift invariance

Notaion

- Positive definite kernel $K : [0,1]^d \times [0,1]^d \to \mathbb{R}$
- Sampling nodes $\mathsf{X} = (\boldsymbol{x}_i)_{i=1}^n \in [0,1]^{n \times d}$
- Gram matrix $\mathsf{K} = (K(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^n \in \mathbb{R}^{n \times n}$ with columns $\boldsymbol{k}_1, \dots, \boldsymbol{k}_n \in \mathbb{R}^n$

Problem

- GPR requires solving linear system $\mathsf{K}\boldsymbol{a} = \boldsymbol{b}$ for $\boldsymbol{a} \in \mathbb{R}^n$ given $\boldsymbol{b} \in \mathbb{R}^n$
- When K dense and unstructured solving $\mathsf{K}\boldsymbol{a} = \boldsymbol{b}$ for $\boldsymbol{a}$ costs $\mathcal{O}(n^3)$ (e.g. using Cholesky decomposition or Eigen decomposition)

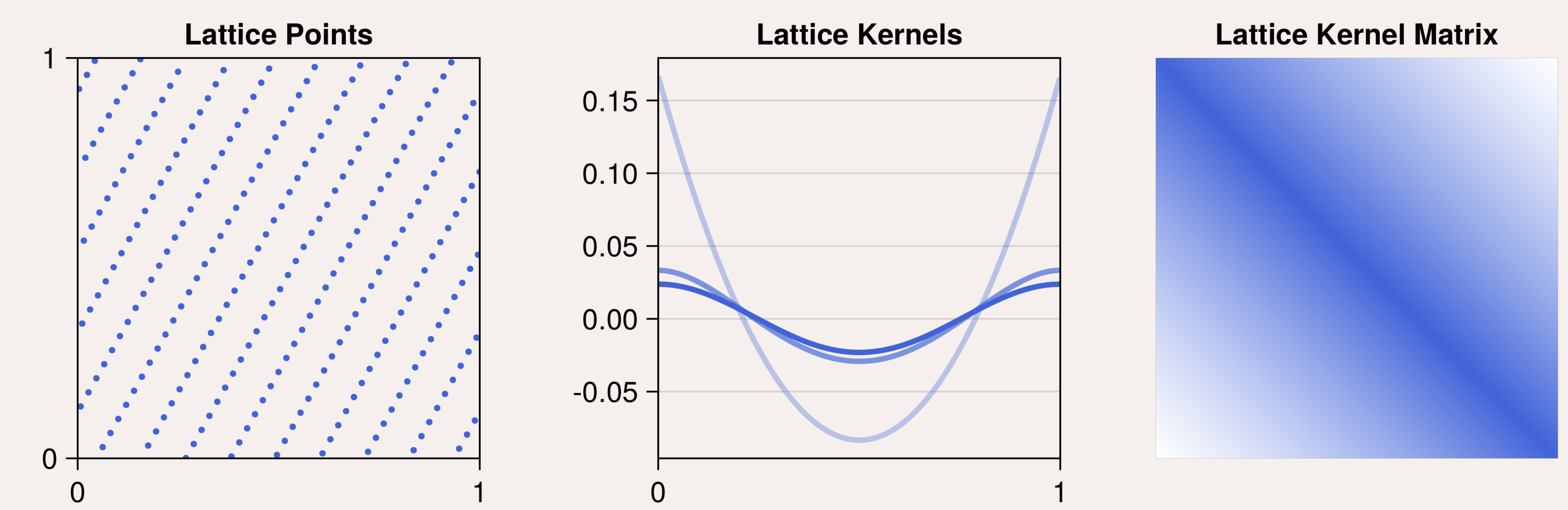
Solution

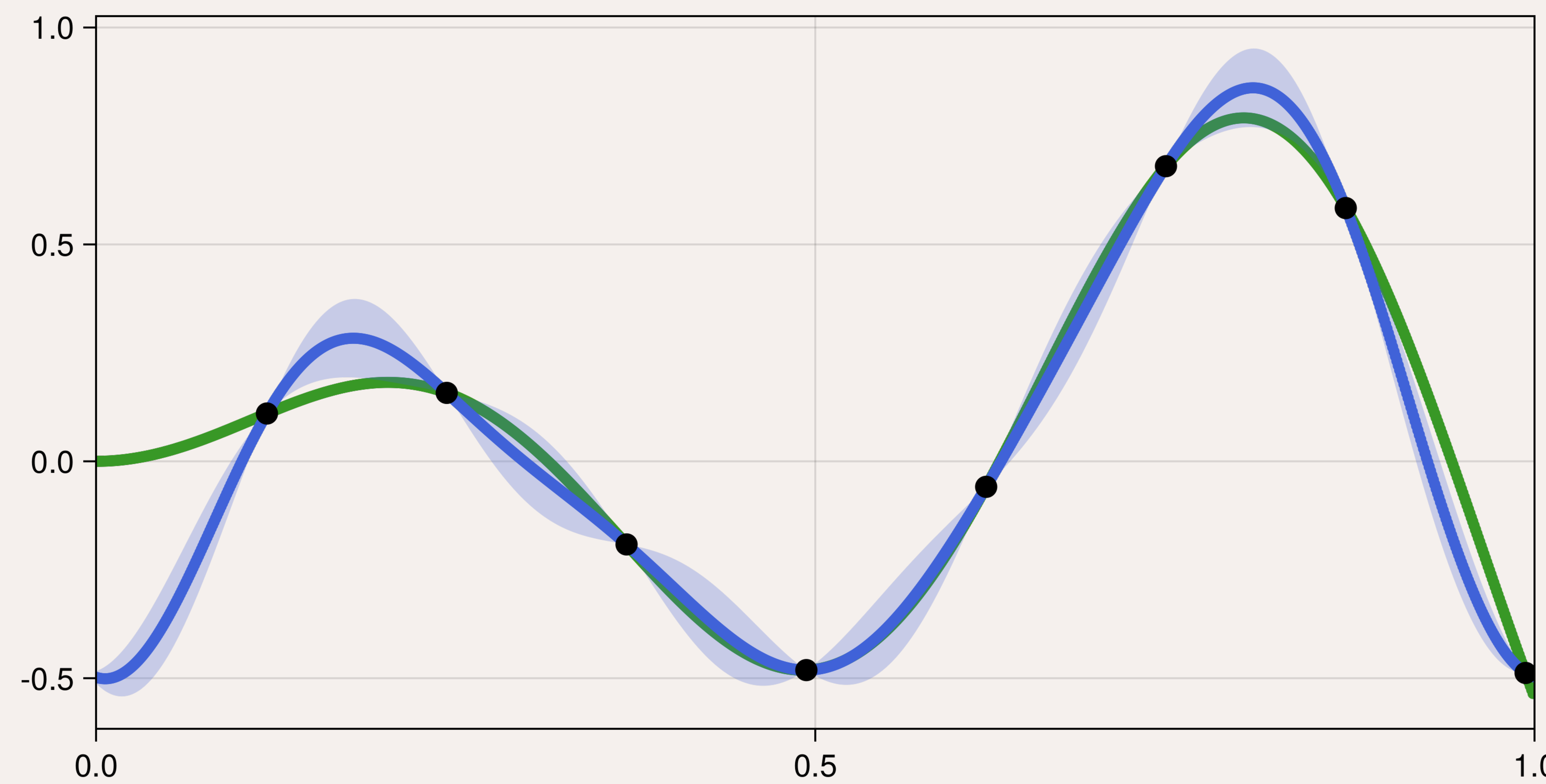
- Induce structure into K so solving $\mathsf{K}\boldsymbol{a} = \boldsymbol{b}$ for $\boldsymbol{a}$ costs $\mathcal{O}(n \log n)$

	Nodes X	Kernel K	Gram matrix K
Method #1	lattice points	shift invariant	circulant
Method #2	digital net	digitally shift invariant	block Toeplitz

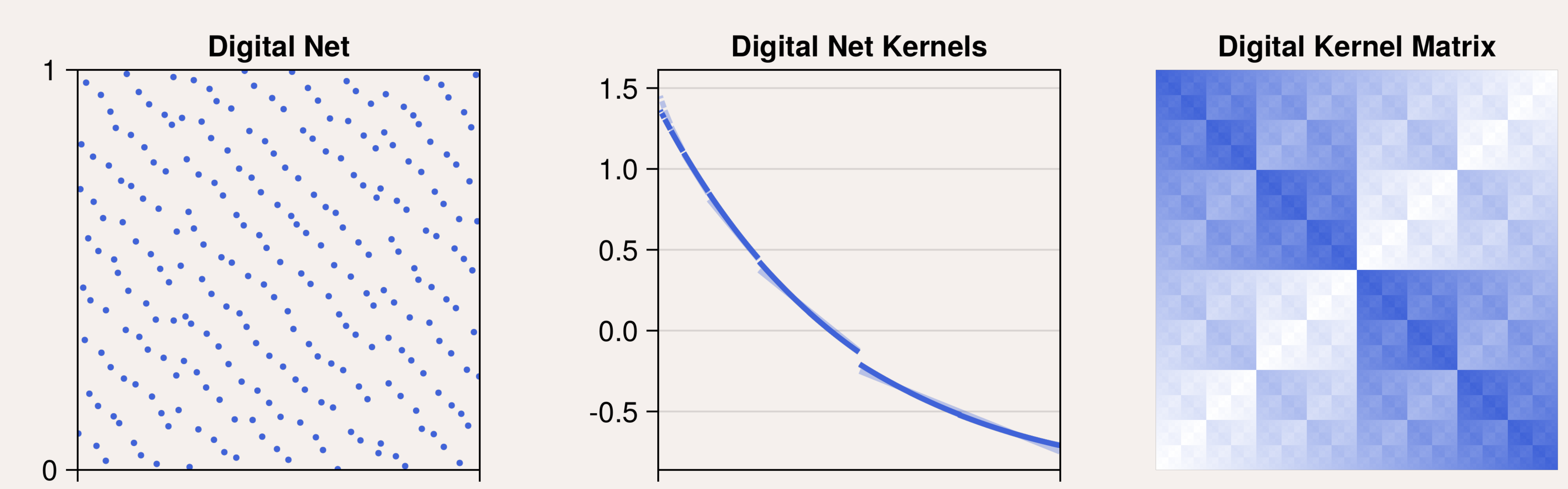
- $\mathsf{K} = \mathsf{V}\Lambda\mathsf{V}^\dagger$ where V is *known* and has columns $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$
- Computing $\mathsf{V}^\dagger \boldsymbol{c}$ costs $\mathcal{O}(n \log n)$ for any $\boldsymbol{c} \in \mathbb{R}^n$
- $\Lambda = \mathrm{diag}(\boldsymbol{\lambda})$ for $\boldsymbol{\lambda} = \sqrt{n}\mathsf{V}^\dagger \boldsymbol{k}_1$ since $\boldsymbol{v}_1 = \boldsymbol{1}/\sqrt{n}$
- ⋆ System solution $\boldsymbol{a} = \mathsf{V}(\mathsf{V}^\dagger \boldsymbol{b}./\boldsymbol{\lambda})$ costs $\mathcal{O}(n \log n)$ to compute
- Kernel parameter optimization also only costs $\mathcal{O}(n \log n)$ per step

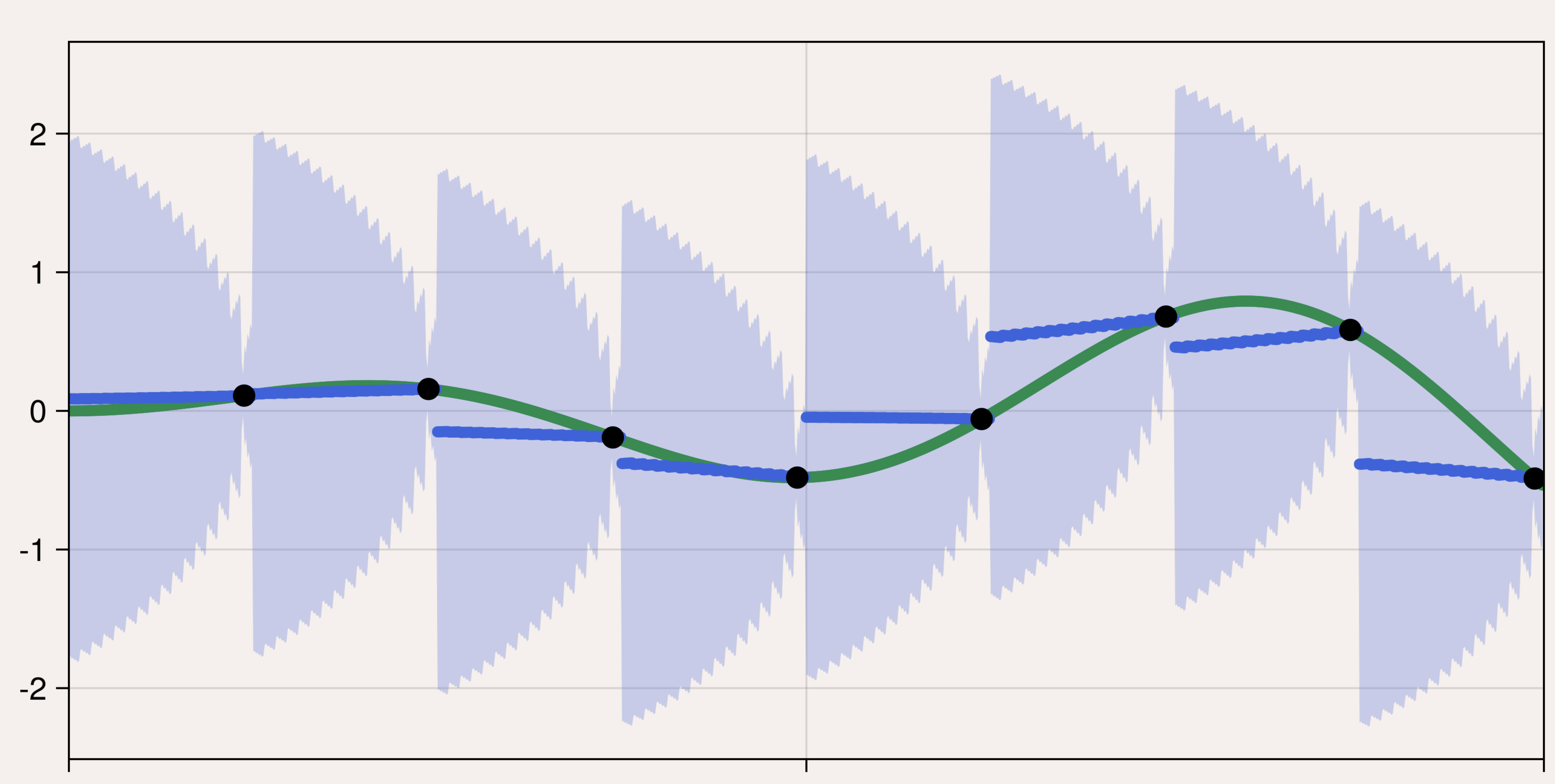
Structure Incuding Method #1
Lattice Nodes with Shift Invariant Kernel





Structure Incuding Method #2
Digital Net Nodes with Digitally Shift Invariant Kernel





Previous Work

- Lattice and digital net pairings for kernel interpolants [14, 15]
- Lattice and digital net pairings for noise-free Bayesian cubature [5, 6, 9]
- Lattice pairing for kernel interpolation in RKHS [7]

Novel Contributions

- Support for noisy observations
- Connections between GPR and kernel interpolation in RKHS
  - GPR posterior mean is optimal kernel interpolant under a certain penalty [8]
  - GPR posterior variance is worst case error of kernel interpolant in certain RKHS [8]
  - GPR kernel parameter optimization automatically parameterizes RKHS
- Extension to support derivative observations
  - Have m derivatives e.g. $m = 1 + d$ when we know f and ∇f
  - GPR typically costs $\mathcal{O}(m^3 n^3)$
  - Generalized structure inducing methods reduce cost to $\mathcal{O}(m^2 n \log n + m^3 n)$
- Developed digitally shift invariant (DSI) kernels for smooth functions
  - Let H_α with $\alpha \geq 1$ be the RKHS with inner product

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

  - H_1 a subset of *known* RKHS with DSI kernel [4]
  - For $\alpha \geq 2$, H_α a subset of *previously unknown* RKHS with DSI kernel
- Application to solving PDEs with random coefficients [13]
- Fast GPR and Quasi-Monte Carlo software [12, 11, 3]

Future Work

- Efficient GPR updates for extensible lattice and digital sequences
- Mix of structured and unstructured sampling points
- Fast GPR for operator learning [1]
- Inducing point methods with lattice or digital nets [10]
- Application to learning nonlinear PDEs with GPR [2]

References

[1] Pau Batlle et al. “Kernel methods are competitive for operator learning”. In: *Journal of Computational Physics* 496 (2024), p. 112549.

[2] Yifan Chen et al. “Solving and learning nonlinear PDEs with Gaussian processes”. In: *Journal of Computational Physics* 447 (2021), p. 110668. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2021.110668. URL: https://www.sciencedirect.com/science/article/pii/S0021999121005635.

[3] Sou-Cheng T. Choi et al. *QMCPy: A Quasi-Monte Carlo Python library*. 2020+. URL: https://github.com/QMCSoftware/QMCSoftware.

[4] Josef Dick and Friedrich Pillichshammer. “Multivariate integration in weighted Hilbert spaces based on Walsh functions and weighted Sobolev spaces”. In: *Journal of Complexity* 21.2 (2005), pp. 149–195.

[5] R. Jagadeeswaran and Fred J. Hickernell. “Fast automatic Bayesian cubature using lattice sampling”. In: *Statistics and Computing* 29.6 (2019), pp. 1215–1229. ISSN: 1573-1375. DOI: 10.1007/s11222-019-09895-9. URL: http://dx.doi.org/10.1007/s11222-019-09895-9.

[6] Rathinavel Jagadeeswaran and Fred J Hickernell. “Fast Automatic Bayesian Cubature Using Sobol Sampling”. In: *Advances in Modeling and Simulation: Festschrift for Pierre L’Ecuyer*. Springer, 2022, pp. 301–318.

[7] Vesa Kaarnioja et al. “Fast approximation by periodic kernel-based lattice-point interpolation with application in uncertainty quantification”. In: *Numerische Mathematik* 150.1 (2022), pp. 33–77.

[8] Motonobu Kanagawa et al. “Gaussian processes and kernel methods: A review on connections and equivalences”. In: *arXiv preprint arXiv:1807.02582* (2018).

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

[10] Edward Snelson and Zoubin Ghahramani. “Sparse Gaussian processes using pseudo-inputs”. In: *Advances in neural information processing systems* 18 (2005).

[11] Aleksei G. Sorokin. *FastGaussianProcesses.jl*. 2023. URL: https://github.com/alegresor/FastGaussianProcesses.jl.

[12] Aleksei G. Sorokin. *QMCGenerators.jl*. 2023. URL: https://github.com/alegresor/QMCGenerators.jl.

[13] Aleksei G. Sorokin et al. *Computationally Efficient and Error Aware Surrogate Construction for Numerical Solutions of Subsurface Flow Through Porous Media*. 2023. arXiv: 2310.13765 [stat.CO].

[14] Xiaoyan Zeng, Peter Kritzer, and Fred J Hickernell. “Spline methods using integration lattices and digital nets”. In: *Constructive Approximation* 30 (2009), pp. 529–555.

[15] Xiaoyan Zeng, King-Tai Leung, and Fred J Hickernell. *Error analysis of splines for periodic problems using lattice designs*. Springer, 2006.