Fast Physics Informed Kernel Methods for Nonlinear PDEs with Unknown Coefficients

SampSci 2024

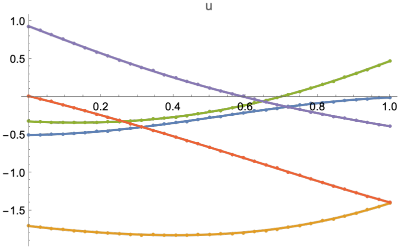
Aleksei G. Sorokin

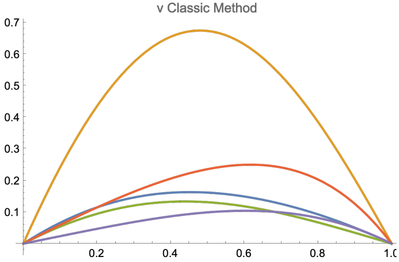
Illinois Institute of Technology, Department of Applied Mathematics

Solving PDEs with Machine Learning

Example PDE: $e^{u(x)}\left[u_x(x)v_x(x) + v_{xx}(x)\right] = -1, \qquad v(0) = 0 = v(1)$

- Try to approximate solution v when input u either deterministic or random
- ML approaches to approximate $u \mapsto v$ include neural networks and kernel methods
- PDE may be non-linear and high dimensional
- Physics informed ML does not rely on reference solver data e.g. finite difference





Neural Networks and Kernel Methods for Solving PDEs

	PDE with deterministic coefficients		PDE with unknown coefficients	
	reference solver	physics informed	reference solver	physics informed
neural networks	[Abiodun et al., 2018]	[Raissi et al., 2019]	[Lu et al., 2021]	[Wang et al., 2021]
kernel methods	[Williams and Rasmussen, 2006]	[Chen et al., 2021]	[Batlle et al., 2024]	proposed solution

- Physics Informed Neural Networks (PINN) [Raissi et al., 2019]
  Loss function of PDE equations using automatic differentiation
- Deep Operator Networks (DeepONets) [Lu et al., 2021]
  Combine network for x with network for u [Wang et al., 2021]

	scalability	convergence guarantees	error rates	interpretability
neural networks	$+$	$+$	$\pm$	$-$
kernel methods	$\pm$	$+$	$+$	$+$

Operator Learning Framework

$$e^{u(x)}\left[u_x(x)v_x(x) + v_{xx}(x)\right] = -1, \qquad v(0) = 0 = v(1)$$

- $u \in \mathcal{U}$ has known distribution and u_x available e.g. a Gaussian process
- $v \in \mathcal{V}$ to be solved for
- **Goal:** Find operator $G^\dagger(u) = v$
- $\phi(u) = (\phi_1(u), \ldots, \phi_n(u))$ linear samples of u e.g. $\phi(u) = (u(x_1), u(x_2), \ldots, u_x(x_1), u_x(x_2), \ldots)$
- $\varphi(v) = (\varphi_1(v), \ldots, \varphi_m(u))$ linear sampler of v e.g. $\varphi(u) = (v(0), v(1), v_x(x_1), v_x(x_2), \ldots, v_{xx}(x_1), v_{xx}(x_2), \ldots)$

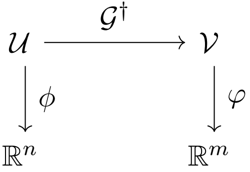
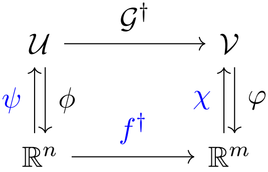


Diagram and framework of [Batlle et al., 2024]

Operator Learning Framework Continued

$$e^{u(x)}\left[u_x(x)v_x(x) + v_{xx}(x)\right] = -1, \qquad v(0) = 0 = v(1)$$

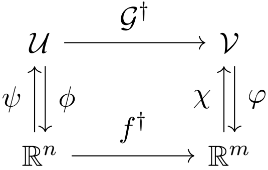
- $\psi(\phi(u)) = \hat{u}$ approximates u from samples $\phi(u) \in \mathbb{R}^n$
- $\chi(\varphi(v)) = \hat{v}$ approximations v from samples $\chi(v) \in \mathbb{R}^m$
- $f^\dagger(\phi(u)) \approx \varphi(v)$ approximates samples of v from samples of u
- $G^\dagger \approx \chi \circ f^\dagger \circ \phi$
  1. Samples u to get $\phi(u)$
  2. Approximates v samples $\varphi(v)$ by $f^\dagger(\phi(u))$
  3. Reconstructs approximate v as $\chi(f^\dagger(\phi(u)))$ from samples

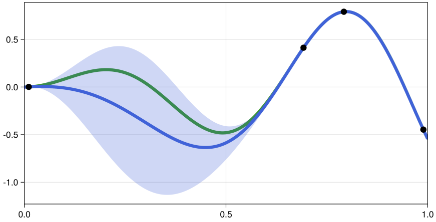


Kernel Methods Idea

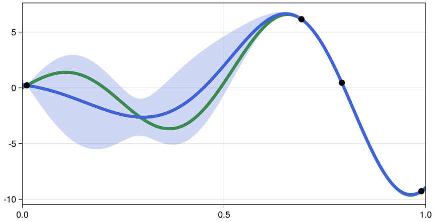
Use RKHS kernel interpolant for ψ, χ, and $f^\dagger$

- ψ rarely used to reconstruct input u
- $f^\dagger$ a vector valued kernel interpolant
- χ an optimal reconstruction map in RKHS
- May reinterpret kernel interpolants as Gaussian processes





GP



GP derivative

Physics Informed Kernel Methods

1. Pick a realization $u \in \mathcal{U}$
2. Sample $\phi(u) \in \mathbb{R}^n$
3. Optimize unknown $\varphi(v) \in \mathbb{R}^m$ to minimize RKHS interpolant norm satisfying PDE
4. Repeat 1. to 3. for many realizations of $u_1, \ldots, u_N$
5. Build kernel interpolant $f^\dagger$ from $\{\phi(u_i)\}_{i=1}^N$ and optimized $\{\varphi(v_i)\}_{i=1}^N$
6. Use mapping $f^\dagger$ and optimal recovery map χ on unseen $\phi(u^*)$

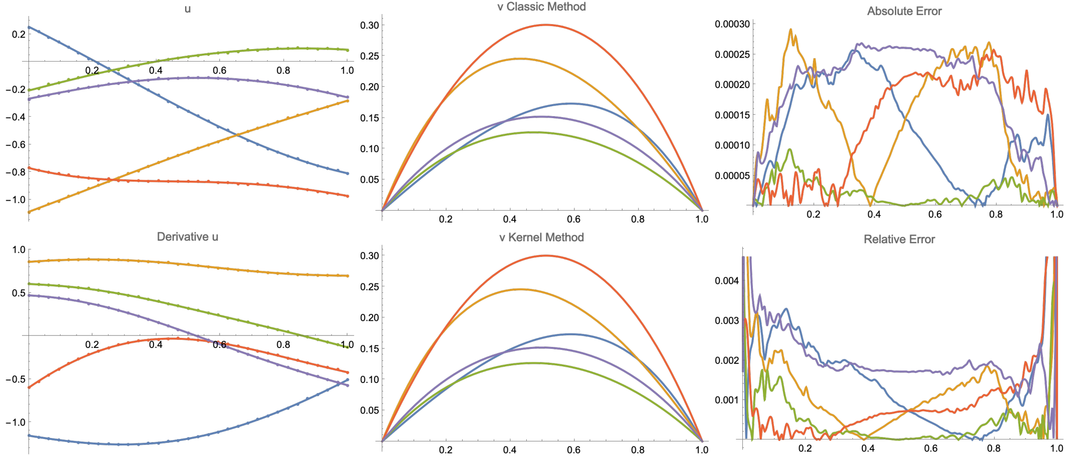
Connections to Existing Kernel Methods for PDEs

- [Chen et al., 2021] is 1. to 3. for deterministic u
- [Batlle et al., 2024] is 5. and 6. for unknown u when reference solver available

Idea: Use physics informed kernel method for deterministic u as the reference solver in kernel operator learning framework

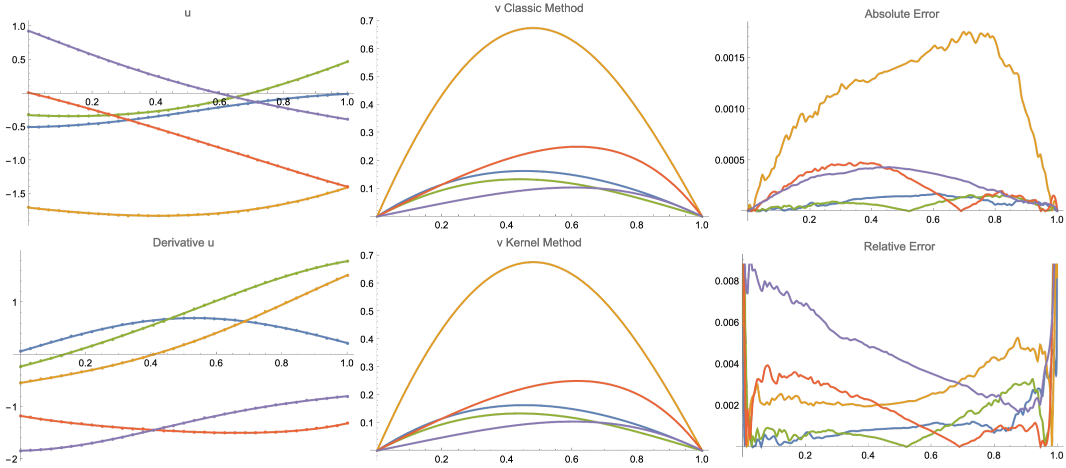
Train 1D Elliptic PDE

$$e^{u(x)}\left[u_x(x)v_x(x)+v_{xx}(x)\right]=-1,\quad v(0)=0=v(1)$$



Test 1D Elliptic PDE

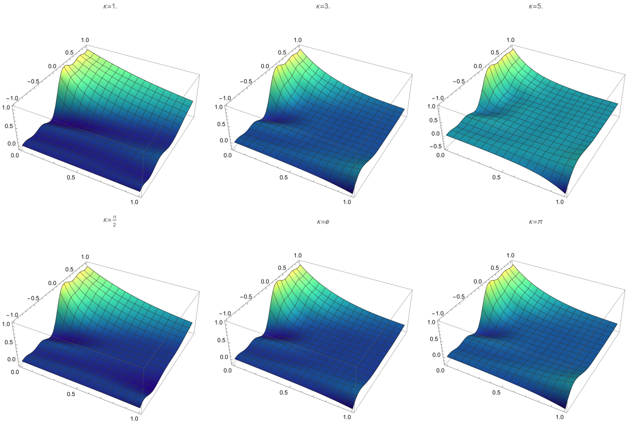
$$e^{u(x)}\left[u_x(x)v_x(x)+v_{xx}(x)\right]=-1,\quad v(0)=0=v(1)$$



Radiative Transfer Equation 1D

Top Row Train. Bottom Row Test

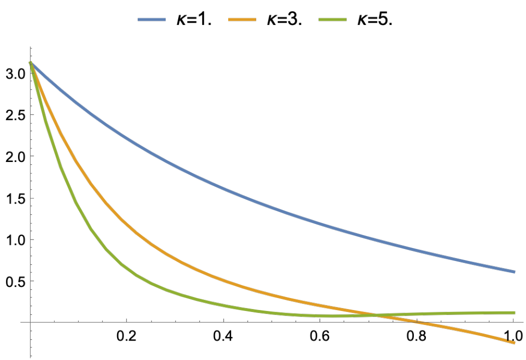
$$sv_x(x,s)+kv(x,s)=0, \qquad v(0,s>0)=1, \qquad v(1,s<0)=0, \qquad \text{scalar } k \text{ unknown}$$

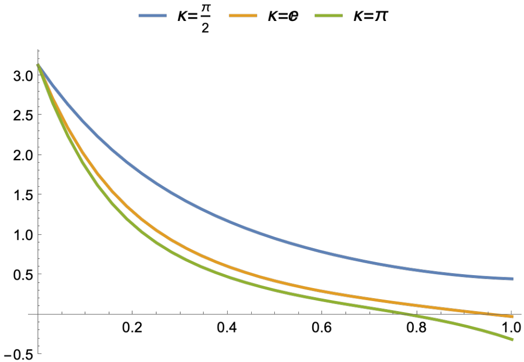


Heat Flux from Radiative Transfer Equation 1D

Left Train. Right Test

$$s v_x(x,s) + \kappa v(x,s) = 0, \qquad v(0, s>0) = 1, \qquad v(1, s<0) = 0,$$ scalar k unknown





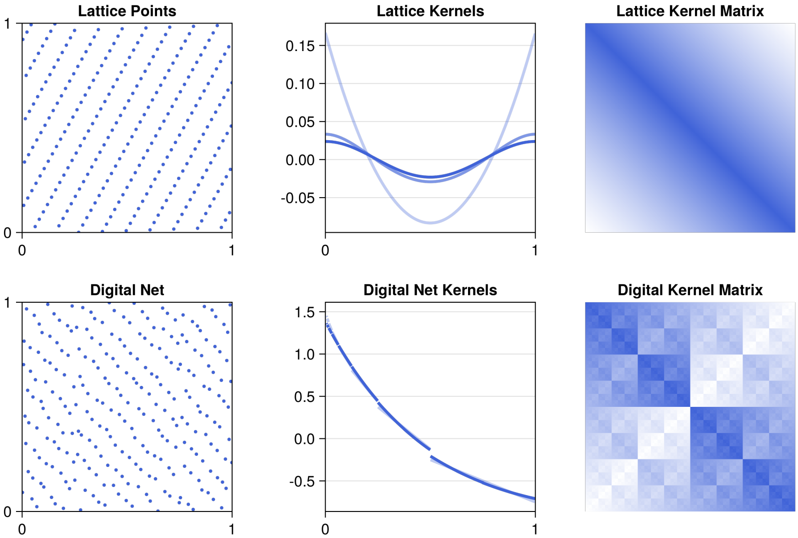
Fast Kernel Methods via Structured Gram Matrices

- RKHS kernel $\mathcal{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ yields Gram matrix $\mathsf{K} = (\mathcal{K}(x_i, x_j))_{i,j=1}^n \in \mathbb{R}^{n \times n}$
- Kernel interpolant fit by solving linear system $\mathsf{K}a = b$ for a
- Choose $\mathcal{K}$ and $\{x_i\}_{i=1}^n$ to induce structure in K allowing faster solve $\mathsf{K}a = b$

samples $\{x_i\}_{i=1}^n$	kernel $\mathcal{K}$	Gram matrix K	Solving $\mathsf{K}a = b$
unstructured	general	dense unstructured	$\mathcal{O}(n^3)$
regular grid	stationary	block Circulant	$\mathcal{O}((n \log n)^d)$
lattice sequence	shift-invariant	circulant	$\mathcal{O}(n \log n)$
digital sequence	digitally shift invariant	block Toeplitz	$\mathcal{O}(n \log n)$

My PhD research extends fast kernel methods with lattice and digital sequence [Jagadeeswaran and Hickernell, 2019, 2022] to accommodate derivative information

Fast Quasi-Monte Carlo Kernel Interpolation



Discussion

Observations

- Proposed method simply couples existing kernel methods for
  1. Solving PDEs with deterministic coefficients [Chen et al., 2021]
  2. Operator learning when a reference solver is available [Batlle et al., 2024]
- Mathematica supports symbolic linear functionals e.g. derivatives and integrals
  - Symbolic computations can sometimes be prohibitively slow
  - Potential opportunity to expand kernel methods to weak formulations

Future Work

- Derive convergence guarantees and rates
- Implement fast kernel methods for this setting
- Apply to challenging PDEs in high dimensions

References I

Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11), 2018.

Pau Batlle, Matthieu Darcy, Bamdad Hosseini, and Houman Owhadi. Kernel methods are competitive for operator learning. *Journal of Computational Physics*, 496: 112549, 2024.

Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Solving and learning nonlinear pdes with gaussian processes. *Journal of Computational Physics*, 447:110668, 2021. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2021.110668. URL https://www.sciencedirect.com/science/article/pii/S0021999121005635.

References II

R. Jagadeeswaran 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.

Rathinavel Jagadeeswaran 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.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

References III

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science advances*, 7(40):eabi8605, 2021.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.