Kernel Methods for Rough PDEs

Ricardo Baptista [1] Edoardo Calvello[1] **Matthieu Darcy**[1] Houman Owhadi [1]
Andrew M. Stuart[1] Xianjin Yang [1]

[1]Department of Computing and Mathematical Sciences
California Institute of Technology

SIAM UQ 2024

Caltech

Talk summary

The problem

We want to **solve with machine learning rough PDEs** of the form

$$\begin{aligned} P(u) &= \xi, \quad x \in \Omega, \\ u &= 0, \quad x \in \partial\Omega \end{aligned} \tag{RPDE}$$

$P : H_0^t(\Omega) \to H^{-s}$ is (non-linear) differential operator.

Canonical example: $P(u) = \Delta u + f(u)$.

$u \in H_0^t(\Omega)$ is the **solution** and $\xi \in H^{-s}$ is the **forcing term**.

Talk summary

The problem

We want to **solve with machine learning rough PDEs** of the form

$$\begin{aligned} P(u) &= \xi, \quad x \in \Omega, \\ u &= 0, \quad x \in \partial\Omega \end{aligned} \tag{RPDE}$$

- $P : H_0^t(\Omega) \to H^{-s}$ is (non-linear) differential operator.

  Canonical example: $P(u) = \Delta u + f(u)$.
- $u \in H_0^t(\Omega)$ is the **solution** and $\xi \in H^{-s}$ is the **forcing term**.

Hierarchy of spaces

$$\underbrace{H_0^t(\Omega)}_{\text{Functions with } t \text{ derivatives}} \qquad L^2(\Omega) \qquad \underbrace{H^{-s}}_{\text{Dual space of } H_0^s}$$

Talk summary

The problem

We want to **solve with machine learning rough PDEs** of the form

$$\begin{aligned} P(u) &= \xi, \quad x \in \Omega, \\ u &= 0, \quad x \in \partial\Omega \end{aligned} \tag{RPDE}$$

$P : H_0^t(\Omega) \to H^{s}$ is (non-linear) differential operator.

Canonical example: $P(u) = \Delta u + f(u)$.

$u \in H_0^t(\Omega)$ is the **solution** and $\xi \in H^{s}$ is the **forcing term**.

The difficulty

Solving (RPDE) is difficult because the forcing term is rough $\xi \notin L^2$ and, as a result, the solution u is also irregular/rough.

Talk summary

The problem

We want to **solve with machine learning rough PDEs** of the form

$$\begin{aligned} P(u) &= \xi, \quad x \in \Omega, \\ u &= 0, \quad x \in \partial\Omega \end{aligned} \qquad \text{(RPDE)}$$

- $P : H_0^t(\Omega) \to H^{-s}$ is (non-linear) differential operator.

  Canonical example: $P(u) = \Delta u + f(u)$.
- $u \in H_0^t(\Omega)$ is the **solution** and $\xi \in H^{-s}$ is the **forcing term**.

The solution

We propose a kernel based method for solving rough PDEs using **negative Sobolev norms** and **weak measurements**, with **provable convergence**.

Outline

Table of Contents

Caltech

Motivation: Stochastic Partial Differential Equations

Stochastic Partial Differential Equations (SPDEs) are PDEs that include **random fluctuations**. A general class of interest are semi-linear SPDEs with additive noise:

$$\frac{d}{dt}u(t,\mathbf{x}) = \Delta u(\mathbf{x},t) + f(u(t,\mathbf{x})) + \xi(t,\mathbf{x}) \tag{SL-SPDE}$$

where:

ξ is **space time white noise**.

Caltech

Motivation: Stochastic Partial Differential Equations

Stochastic Partial Differential Equations (SPDEs) are PDEs that include **random fluctuations**. A general class of interest are semi-linear SPDEs with additive noise:

$$\frac{d}{dt}u(t,\mathbf{x}) = \Delta u(\mathbf{x},t) + f(u(t,\mathbf{x})) + \xi(t,\mathbf{x}) \tag{SL-SPDE}$$

where:

ξ is **space time white noise**.

Examples

Phase field models (Allen-Cahn equation).

Mathematical biology (Nagumo equation).

Filtering and sampling.

Motivation: Solving SPDEs

Numerically solving an SPDE of the form

$$\frac{d}{dt}u(t,\mathbf{x}) = \Delta u(\mathbf{x},t) + f(u(t,\mathbf{x})) + \xi(t,\mathbf{x}) \qquad \text{(SL-SPDE)}$$

typically involves drawing samples of ξ and solving (SL-SPDE) for that realization.

Motivation: Solving SPDEs

Numerically solving an SPDE of the form

$$\frac{d}{dt}u(t,\mathbf{x}) = \Delta u(\mathbf{x},t) + f(u(t,\mathbf{x})) + \xi(t,\mathbf{x}) \tag{SL-SPDE}$$

typically involves drawing samples of ξ and solving (SL-SPDE) for that realization.

Difficulty

Solving (SL-SPDE) is difficult because

1. The forcing term $\xi \notin L^2$ a.s. and is not pointwise defined.
2. The solution u is irregular/rough.

This motivates us to develop methods to solve PDEs with very rough forcing terms and/or irregular solutions.

Caltech

Table of Contents

Caltech

Solving PDEs with ML

Machine learning methods are a novel way of solving PDEs in a *data-driven way*

$$u^\dagger := \arg\min_{u \in S} \underbrace{\gamma_1 \|P(u) - \xi\|^2_{L^2(\Omega)}}_{\text{PDE data}} + \underbrace{\gamma_2 \|u\|^2_{L^2(\partial\Omega)}}_{\text{Boundary data}} + \underbrace{\gamma_3 R(u)}_{\text{Regularization}}$$

Infinite data

Caltech

Solving PDEs with ML

Machine learning methods are a novel way of solving PDEs in a *data-driven way*

$$u^y := \underset{u2S}{\arg\min} \underbrace{\gamma_1 jjP(u) \quad \xi jj^2_{L^2(\Omega)}}_{\text{PDE data}} + \underbrace{\gamma_2 jjujj^2_{L^2(\partial\Omega)}}_{\text{Boundary data}} + \underbrace{\gamma_3 R(u)}_{\text{Regularization}} \qquad \text{Infinite data}$$

$$\underset{u2S}{\arg\min} \underbrace{\frac{\gamma_1}{N_\Omega} \sum_{i=1}^{N_\Omega} jP(u)(x_i) \quad \xi(x_i)j^2}_{\text{PDE data approximation}} + \underbrace{\frac{\gamma_2}{N_{\partial\Omega}} \sum_{j=1}^{N_{\partial\Omega}} ju(x_j)j^2}_{\text{Boundary approximation}} + \gamma_3 R(u) \qquad \text{Finite data}$$

Caltech

Solving PDEs with ML

Machine learning methods are a novel way of solving PDEs in a *data-driven way*

$$u^y := \arg\min_{u2S} \underbrace{\gamma_1 jjP(u) \quad \xi jj^2_{L^2(\Omega)}}_{\text{PDE data}} + \underbrace{\gamma_2 jjujj^2_{L^2(\partial\Omega)}}_{\text{Boundary data}} + \underbrace{\gamma_3 R(u)}_{\text{Regularization}} \qquad \text{Infinite data}$$

$$\arg\min_{u2S} \underbrace{\frac{\gamma_1}{N_\Omega}\sum_{i=1}^{N_\Omega} jP(u)(x_i) \quad \xi(x_i)j^2}_{\text{PDE data approximation}} + \underbrace{\frac{\gamma_2}{N_{\partial\Omega}}\sum_{j=1}^{N_{\partial\Omega}} ju(x_j)j^2}_{\text{Boundary approximation}} + \gamma_3 R(u) \qquad \text{Finite data}$$

Three difficulties with Rough PDEs

1. The L^2 norm is not appropriate.
2. Solves the PDE pointwise.
3. Requires that u may be well approximated by the class S.

Solving PDEs with ML

The **Physics Informed Neural Network** (PINN) approach selects u_θ^y to be a neural network.
The **Kernel/Gaussian process approach**[1] selects u^y from a Reproducing Kernel Hilbert Space (RKHS).

Advantages of the kernel approach

- Provable convergence.
- Uncertainty quantification.

[1] Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. "Solving and learning nonlinear PDEs with Gaussian processes". In: *Journal of Computational Physics* (2021)

Kernel Methods and Gaussian Processes

Reproducing Kernel Hilbert Space

A kernel/covariance function $K : X \times X \to \mathbb{R}$ defines a Hilbert space of functions H_K and corresponding Gaussian Process $\zeta \sim \mathcal{N}(0, K)$.

Kernel Methods and Gaussian Processes

Reproducing Kernel Hilbert Space

A kernel/covariance function $K : X \quad X \to \mathbb{R}$ defines a Hilbert space of functions H_K and corresponding Gaussian Process $\zeta \quad \mathcal{N}(0, K)$.

GP and kernel regression

Given some noisy observations

$$u \;(x_j) = y_j + \varepsilon_j, \quad j = 1, \ldots, N$$

we can recover u by **conditioning** or solving a **linear least-squares** problem:

$$u^y = \mathbb{E}\left[\zeta \mid \zeta(x_j) = y_j + \varepsilon_j\right] \qquad (\;) \qquad u^y = \operatorname*{arg\,min}_{u \in H_K} \|u(X) \quad Y\|^2 + \gamma \|u\|^2_{H_K}$$

Gaussian Processes for Solving PDEs

MAP Estimator of a GP[2]

Solving a **non-linear least squares**

$$u^\dagger = \arg\min_{u \in H_K} \underbrace{\|P(u) - \xi\|^2_{L^2}}_{\text{PDE data}} + \underbrace{\gamma \|u\|^2_{H_K}}_{\text{RKHS regularization}}$$

$$\text{s.t. } \underbrace{u = 0 \text{ on } \partial\Omega}_{\text{Boundary term}}$$

Interpreted as a **MAP estimator** of a GP conditioned on non-linear measurements.

Classical solution

This method solves the PDE in strong form:

$$P(u)(x_i) = \xi(x_i) \quad i = 1, \ldots, N$$

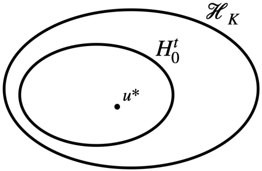
[2] Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. "Solving and learning nonlinear PDEs with Gaussian processes". In: *Journal of Computational Physics* (2021)

Table of Contents

Challenges and solutions

Three difficulties with Rough PDEs

1. The L^2 norm is not appropriate.
2. Solves the PDE pointwise.
3. Requires $u \in H_K$ for the convergence theory.



Solutions

Our main contributions are to solve the three main difficulties:

1. Using a **negative Sobolev norm** H^{-s} instead of the usual L^2 norm.
2. **Efficient approximation** of the H^{-s} norm with weak measurements.
3. **Convergence results** of the method without $u \in H_K$.

A Novel Approach to Solving Rough PDEs

New problem

$$u^{\gamma} = \arg\min_{u \in H_K} \|P(u) - \xi\|^2_{H^{-s}} + \gamma \|u\|^2_{H_K} \quad \text{Infinite data}$$

$$\text{s.t. } u = 0 \text{ on } \partial\Omega.$$

$$u^{\gamma,N,M} = \arg\min_{u \in H_K} \|P(u) - \xi\|^2_{V^N} + \gamma \|u\|^2_{H_K} \quad \text{Finite data}$$

$$\text{s.t. } u(x_j) = 0 \quad j = 1, \ldots, M \text{ on } \partial\Omega.$$

Caltech

Negative Sobolev Norm approximation

To go from the infinite data problem to the finite data problem, we need to compute an approximation of the negative Sobolev norm:

$$kf k_{H^{\quad s}} \qquad kf jj_{V^N}$$

[3] Houman Owhadi and Clint Scovel. *Operator-Adapted Wavelets, Fast Solvers, and Numerical Homogenization*. Cambridge University Press, Oct. 2019

Negative Sobolev Norm approximation

To go from the infinite data problem to the finite data problem, we need to compute an approximation of the negative Sobolev norm:

$$\|f\|_{H^{-s}} \approx \|f\|_{V^N}$$

Choose Test Space $V^N := \operatorname{span}\{\varphi_i\}_{i=1}^N$ (Ex: Haar basis)

$$[f, \varphi] := \left(\int f \varphi_1, \int f \varphi_2, \ldots, \int f \varphi_n \right), \quad A_{i,j} := \int_\Omega \varphi_i (-\Delta)^{-s} \varphi_j .$$

Then

$$\|f\|_{V^N} := \sqrt{[f, \varphi]^\intercal A [f, \varphi]}.$$

The entries of A can be efficiently computed with fast solvers for elliptic PDEs[3].

[3] Houman Owhadi and Clint Scovel. *Operator-Adapted Wavelets, Fast Solvers, and Numerical Homogenization*. Cambridge University Press, Oct. 2019

Finite dimensional problem

Finite dimensional problem

$$u^{\gamma,N,M} = \underset{u \in H_K}{\arg\min}[P(u) \quad \xi,\varphi]A[P(u) \quad \xi,\varphi] + \gamma\|u\|_{H_K}^2$$

$$\text{s.t. } u(x_j) = 0 \quad j = 1,\ldots,M \text{ on } \partial\Omega.$$

Weak solution

This method can be interpreted as solving the PDE in weak form:

$$[P(u),\varphi_i] = [\xi,\varphi_i] \quad i = 1,\ldots M.$$

This problem can be solved efficiently with the representer theorem and a non-linear least squares optimization techniques such as a variant of the Gauss-Newton algorithm.

Caltech

Table of Contents

Caltech

Convergence: Assumptions

Assumptions on the PDE

The operator $P : H_0^t \to H^{-s}$ is continuous.

The solution operator is (locally) stable

$$\|u - u\|_{H_0^t} \leq C\|P(u) - \xi\|_{H^{-s}}$$

Assumptions on the method

The space V^N is dense in H^{-s} as $N \to \infty$.

The fill distance on the boundary goes to 0 as $M \to \infty$.

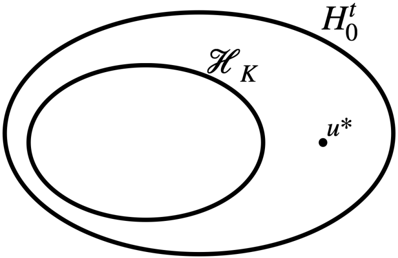
$H_K \hookrightarrow H_0^t(\Omega)$ and H_K is dense in $H_0^t(\Omega)$ (satisfied for Matérn kernels).

Convergence: Main Theorem

Theorem (Convergence to the True Solution)

Let $u^{\gamma,M,N} \in H_K$ solve the approximate problem, then

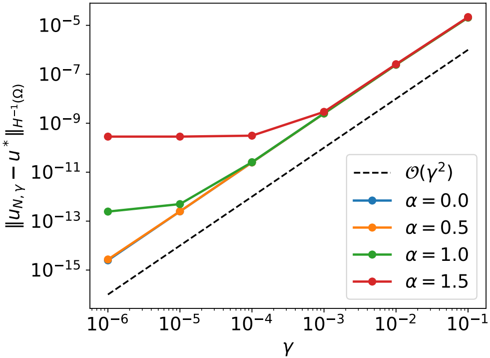
$$\lim_{\gamma \to 0} \lim_{M \to \infty} \lim_{N \to \infty} \left\| u^{M,N,\gamma} - u^* \right\|_{H_0^t} = 0.$$



Caltech

Table of Contents

Example: Linear Rough PDE



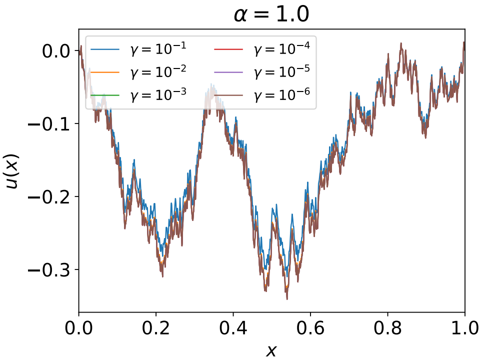


Figure: Linear PDE: $u_{xx} = f$ with Dirichlet BC for $f(x) = \sum_{k=1}^{1} k\xi_k\varphi_k(x)$ and eigenfunction measurements.

Caltech

Example: Non-linear Rough PDE

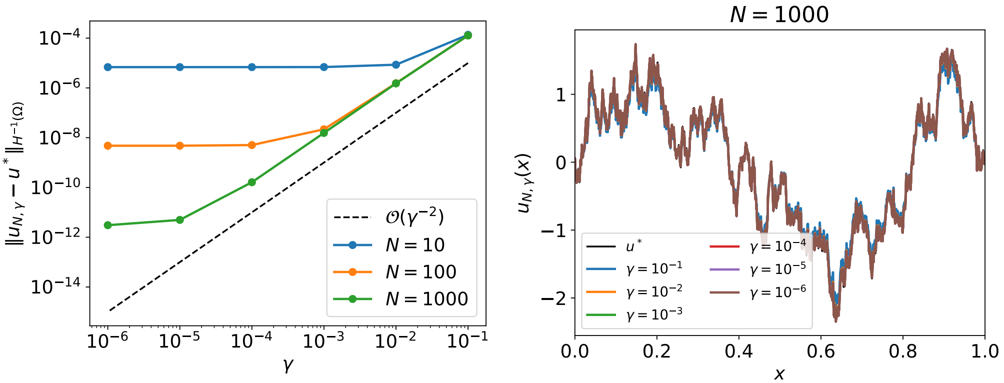


Figure: Nonlinear PDE: $u_{xx} + \sin(u) = f$ with periodic BC and $f(x) = \sum_{k=1}^{1} k^{\alpha} \xi_k \varphi_k(x)$, for $\alpha = 0.49$.

Conclusion

We propose a kernel-based framework for solving PDEs with irregular forcing terms which addresses 3 difficulties:

1. The L^2 norm is not appropriate $\rightarrow$ Use a weaker Sobolev norm.
2. Requires pointwise solution $\rightarrow$ Approximate the norm with weak measurements.
3. Requires that we approximate u $\rightarrow$ Convergence holds even when $u \notin H_K$.

Our numerical experiments show promising initial results and we plan to extend this methodology to solving SPDEs.

mdarcy@caltech.edu