Probabilistic Numerical Methods for Non-Linear Partial Differential Equations: Strong Form Solutions

Jon Cockayne[1]
Chris Oates[2]
Tim Sullivan[3]
Mark Girolami[4]
[1] University of Warwick
[2] University of Technology Sydney
[3] Free University of Berlin
[4] Alan Turing Institute for Data Science

January 5, 2016

Introduction: What is a “Probabilistic Numerical Method”?

Roots in Diaconis [1988].

Recent interest in:

- Quadrature (Briol et al. [2015])
- Optimization (Snoek et al. [2012])
- ODEs and PDEs (Conrad et al. [2015], Schober et al. [2014])

Introduction: What is a “Probabilistic Numerical Method”?

Roots in Diaconis [1988].

Recent interest in:

- Quadrature (Briol et al. [2015])
- Optimization (Snoek et al. [2012])
- ODEs and PDEs (Conrad et al. [2015], Schober et al. [2014])

ProbNum attempts to give probabilistic answers to ‘deterministic’ problems.

Deterministic method $\not\Rightarrow$ accurate solution.

Motivation

PDEs

We wish to solve problems of the form:

$$\begin{aligned} \mathcal{A}u(\mathbf{x}) &= g(\mathbf{x}) && \mathbf{x} \in D \\ \mathcal{B}u(\mathbf{x}) &= 0 && \mathbf{x} \in \partial D \end{aligned}$$

$\mathcal{A}$, $\mathcal{B}$ are differential operators. Think of this as a system of operator equations:

$$\mathcal{O}u := \begin{bmatrix} \mathcal{A} \\ \mathcal{B} \end{bmatrix} u = \begin{bmatrix} g \\ 0 \end{bmatrix}$$

Example: Poisson's Equation with Dirichlet BCs

$$\mathcal{A}u := \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2}$$

$$\mathcal{B}u := u$$

Why ProbNum?

Finite Element method constructs the solution over elements of a mesh.

“Gridding away” discretisation error is often computationally infeasible:

- Complex, high-dimensional domain.
- Complex, nonlinear, parabolic PDE.

Instead try to capture it.

Why ProbNum?

Finite Element method constructs the solution over elements of a mesh.

"Gridding away" discretisation error is often computationally infeasible:

- Complex, high-dimensional domain.
- Complex, nonlinear, parabolic PDE.

Instead try to capture it.

However. . . probabilistic approach may make already costly methods more costly. Why bother?

- Inverse Problems:
  - Incorporate covariance into likelihood.
  - Posterior distribution reflects numerical error.

Linear Problems

Linear PDEs: Theory I

$$\mathcal{A}u(\mathbf{x}) = g(\mathbf{x}) \qquad \mathbf{x} \in D$$

$$\mathcal{B}u(\mathbf{x}) = 0 \qquad \mathbf{x} \in \partial D$$

- $\mathcal{A}$, $\mathcal{B}$ linear (for now). We build a Gaussian Process model for u.
- Prior: $u \sim GP(0, k)$ for some kernel k.

Linear PDEs: Theory I

$$\mathcal{A}u(\mathbf{x}) = g(\mathbf{x}) \qquad \mathbf{x} \in D$$
$$\mathcal{B}u(\mathbf{x}) = 0 \qquad \mathbf{x} \in \partial D$$

- $\mathcal{A}$, $\mathcal{B}$ linear (for now). We build a Gaussian Process model for u.
- Prior: $u \sim GP(0, k)$ for some kernel k.
- For subsets $X = \{x_i\}$, $Y = \{y_j\}$ of $\bar{D}$:

$$K(X, Y) = [k(x_i, y_j)]_{ij}$$
$$\mathcal{A}K(X, Y) = [\mathcal{A}k(x_i, y_j)]_{ij} \qquad \text{etc.}$$

Linear PDEs: Theory I

$$\mathcal{A}u(\mathbf{x}) = g(\mathbf{x}) \qquad \mathbf{x} \in D$$
$$\mathcal{B}u(\mathbf{x}) = 0 \qquad \mathbf{x} \in \partial D$$

- $\mathcal{A}$, $\mathcal{B}$ linear (for now). We build a Gaussian Process model for u.
- Prior: $u \sim GP(0, k)$ for some kernel k.
- For subsets $X = \{x_i\}$, $Y = \{y_j\}$ of $\bar{D}$:

$$K(X, Y) = [k(x_i, y_j)]_{ij}$$
$$\mathcal{A}K(X, Y) = [\mathcal{A}k(x_i, y_j)]_{ij} \qquad \text{etc.}$$

- $\bar{\mathcal{A}}$ denotes the adjoint of $\mathcal{A}$:

$$\mathcal{A}k(x, y) \qquad \bar{\mathcal{A}}k(x, y)$$

Linear PDEs: Theory II

Choose design points $X_0 = X_0^{\mathcal{A}} \cup X_0^{\mathcal{B}}$ & observations $\mathbf{v} = \{v_i\}\ldots$

Choose design points $X_0 = X_0^{\mathcal{A}} \cup X_0^{\mathcal{B}}$ & observations $\mathbf{v} = \{v_i\}\ldots$

Posterior Measure for u

$$u(\mathbf{x})|\mathbf{v} \sim \mathcal{N}(\mu, \Sigma)$$

where:

$$\mu := \bar{\mathcal{O}}K(\mathbf{x}, X_0)\left[\mathcal{O}\bar{\mathcal{O}}K(X_0, X_0)\right]^{-1}\mathbf{v}$$

$$\Sigma := K(\mathbf{x}, \mathbf{x}) - \bar{\mathcal{O}}K(\mathbf{x}, X_0)\left[\mathcal{O}\bar{\mathcal{O}}K(X_0, X_0)\right]^{-1}\mathcal{O}K(X_0, \mathbf{x})$$

Choose design points $X_0 = X_0^{\mathcal{A}} \cup X_0^{\mathcal{B}}$ & observations $\mathbf{v} = \{v_i\}\ldots$

Posterior Measure for u

$$u(\mathbf{x})|\mathbf{v} \sim \mathcal{N}(\mu, \Sigma)$$

where:

$$\mu := \bar{\mathcal{O}}K(\mathbf{x}, X_0)\left[\mathcal{O}\bar{\mathcal{O}}K(X_0, X_0)\right]^{-1}\mathbf{v}$$

$$\Sigma := K(\mathbf{x},\mathbf{x}) - \bar{\mathcal{O}}K(\mathbf{x}, X_0)\left[\mathcal{O}\bar{\mathcal{O}}K(X_0, X_0)\right]^{-1}\mathcal{O}K(X_0, \mathbf{x})$$

$$\mathcal{O}K(X_0, \mathbf{x}) = \begin{bmatrix} \mathcal{A}K(X_0^{\mathcal{A}}, \mathbf{x}) \\ \mathcal{B}K(X_0^{\mathcal{B}}, \mathbf{x}) \end{bmatrix}$$

$$\mathcal{O}\bar{\mathcal{O}}K(X, Y) = \begin{bmatrix} \mathcal{A}\bar{\mathcal{A}}K(X_0^{\mathcal{A}}, X_0^{\mathcal{A}}) & \mathcal{A}\bar{\mathcal{B}}K(X_0^{\mathcal{A}}, X_0^{\mathcal{B}}) \\ \mathcal{B}\bar{\mathcal{A}}K(X_0^{\mathcal{B}}, X_0^{\mathcal{A}}) & \mathcal{B}\bar{\mathcal{B}}K(X_0^{\mathcal{B}}, X_0^{\mathcal{B}}) \end{bmatrix}$$

Canonical Elliptic PDE

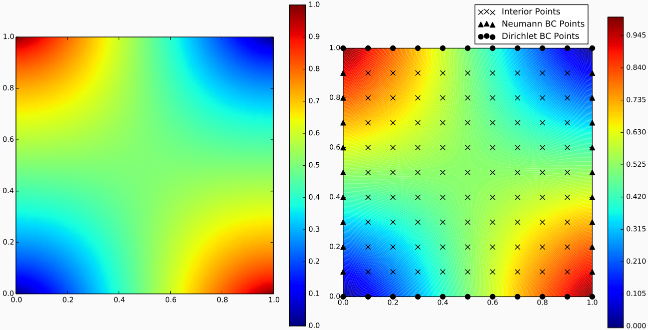
Linear Example

$$
\begin{aligned}
\nabla \cdot \kappa(\mathbf{x};\theta)\nabla u(\mathbf{x}) &= 0 && \text{in } [0,1]^2 \\
u(\mathbf{x}) &= x_1 && \text{at } x_2 = 0 \\
u(\mathbf{x}) &= 1 - x_1 && \text{at } x_2 = 1 \\
\frac{\partial u}{\partial x_1} &= 0 && \text{at } x_1 = 0, 1
\end{aligned}
$$

Approximate κ with a truncated Karhunen-Loève expansion:

$$
\kappa(\mathbf{x};\theta) = \exp\left(\sum_{k=1}^{6} \frac{\theta_k}{k^2} \cos(2\pi(s_k x_1 + r_k x_2))\right)
$$

Canonical Elliptic PDE: Solution

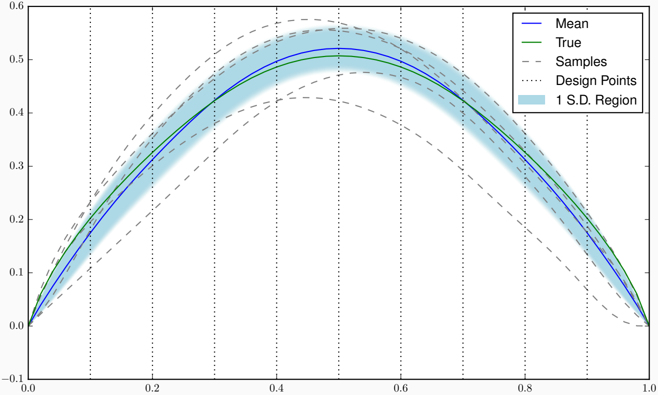


Left: Model solution (generated with FEM, 100×100 mesh).

Right: Mean function of PN solution, design points annotated.

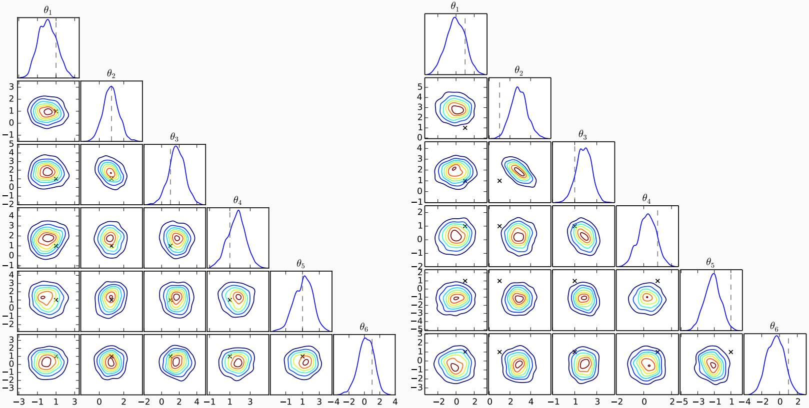
PN solution uses a squared exponential kernel: $k = \exp\left(-\frac{\|x-y\|_2^2}{2\sigma^2}\right)$

Forward Sample Paths



Samples drawn from posterior on a 10×10 grid, along $x_1 = x_2$.

Inverse Problem



Posterior distributions of coefficients of κ, using the PN forward model (left) vs. FEM (right)

Dashed lines / black crosses reflect the true values.

Nonlinear Problems

Nonlinear Case: Theory I

$$\mathcal{A}u(\mathbf{x}) = g(\mathbf{x}) \qquad \mathbf{x} \in D$$
$$\mathcal{B}u(\mathbf{x}) = 0 \qquad \mathbf{x} \in \partial D$$

- $\mathcal{A}$ and/or $\mathcal{B}$ nonlinear. No more nice closed form posterior!
- Think about a class of PDEs which decompose as:

$$(\mathcal{L} + \mathcal{A})\, u = g$$

$\mathcal{L}$ linear, $\mathcal{A}$ invertible.

Nonlinear Case: Theory II

- Introduce a 'dummy variable' z
- Construct a new system:

$$\begin{aligned} \mathcal{L}u &= (g - z) \\ u &= \mathcal{A}^{-1}z \\ \mathcal{B}u &= 0 \end{aligned}$$

Nonlinear Case: Theory II

- Introduce a 'dummy variable' z
- Construct a new system:

$$\begin{aligned}\mathcal{L}u &= (g - z)\\ u &= \mathcal{A}^{-1}z\\ \mathcal{B}u &= 0\end{aligned}$$

- Integrate out z:
  - Generate an approximate solution $\hat{u}$ (eg. using FEM on a coarse mesh)
  - Guess $z = \mathcal{A}\hat{u}$.
  - Place an importance distribution over z.
- Exact likelihood becomes approximate.

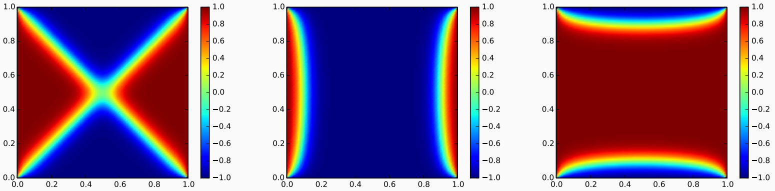
Steady-State Allen-Cahn Equation

$$
\begin{aligned}
-\delta\nabla^2 u + \delta^{-1}(u^3 - u) &= 0 & (x, y) &\in (0,1)^2 \\
u &= 1 & x = 0, x = 1, y &\in (0,1) \\
u &= -1 & y = 0, y &= 1
\end{aligned}
$$

Steady-State Allen-Cahn Equation

$$
\begin{aligned}
-\delta \nabla^2 u + \delta^{-1}(u^3 - u) &= 0 \qquad && (x, y) \in (0,1)^2 \\
u &= 1 && x = 0, x = 1, y \in (0,1) \\
u &= -1 && y = 0, y = 1
\end{aligned}
$$

Three solutions!



Inverse Problem

We want to infer δ.

Multiple solutions makes the inverse problem very hard!

However...in the PN setting we can place a mixture distribution over z.

Inverse Problem

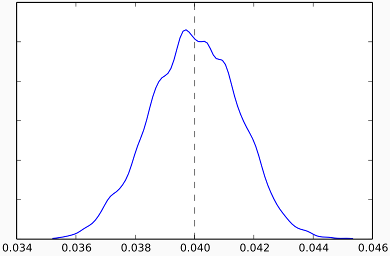
We want to infer δ.

Multiple solutions makes the inverse problem very hard!

However... in the PN setting we can place a mixture distribution over z.

Right: Inference of δ using PN.

- Prior: $\delta \sim U(0.02, 1)$.
- Deflation [Farrell et al., 2014] used to generate the multiple $\hat{u}$ at each δ.



Summary

Summary

- With PN we can capture and propagate uncertainty from numerical error.
- This helps us to make sound inferences!
- Many sampling challenges with nonlinear problems.

Next Steps

Lots still to be done. . .

- Computation — very expensive compared to FEM
- Development needed on the weak form [Conrad et al., 2015].
- Optimal choices of kernel / RKHS [Owhadi, 2014].
- Experimental design — how should we place X_0 to minimise posterior uncertainty?
- Applications to physical problems of interest.

Thanks!

References

François-Xavier Briol, Chris J Oates, Mark Girolami, Michael A Osborne, and Dino Sejdinovic. Probabilistic integration. *arXiv preprint arXiv:1512.00933*, 2015.

Patrick R Conrad, Mark Girolami, Simo Särkkä, Andrew Stuart, and Konstantinos Zygalakis. Probability measures for numerical solutions of differential equations. *arXiv preprint arXiv:1506.04592*, 2015.

Persi Diaconis. Bayesian numerical analysis. *Statistical decision theory and related topics IV*, 1:163–175, 1988.

Patrick E Farrell, Ásgeir Birkisson, and Simon W Funke. Deflation techniques for finding distinct solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1410.5620*, 2014.

Houman Owhadi. Bayesian numerical homogenization. *arXiv preprint arXiv:1406.6668*, 2014.

Michael Schober, David K Duvenaud, and Philipp Hennig. Probabilistic ode solvers with runge-kutta means. In *Advances in Neural Information Processing Systems*, pages 739–747, 2014.

Jasper Snoek, Hugo Larochelle, and Rp Adams. Practical Bayesian Optimization of Machine Learning Algorithms. *Nips*, pages 1–9, 2012. doi: 2012arXiv1206.2944S. URL `https://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf`.