Uncertainty-aware surrogate models for inverse problems

Aretha Teckentrup

School of Mathematics, University of Edinburgh

Joint work with:

Tianming Bai, Kostas Zygalakis (University of Edinburgh)

Christian Jiminez Beltran, Antonio Vergari, Kostas Zygalakis (University of Edinburgh)

Uncertainty Quantification for High-Dimensional Problems

THE UNIVERSITY of EDINBURGH School of Mathematics

MAXWELL INSTITUTE FOR MATHEMATICAL SCIENCES

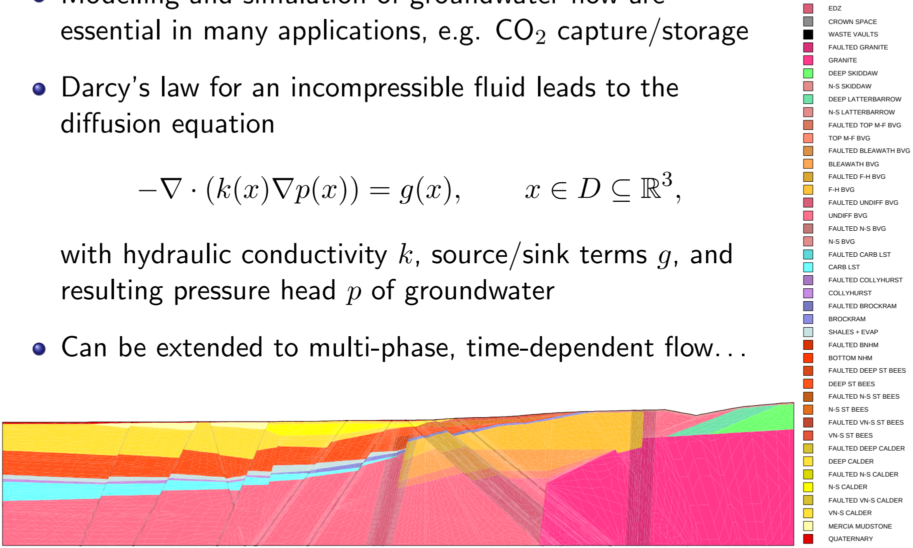
Outline

Motivation: Application in Groundwater Flow

- Modelling and simulation of groundwater flow are essential in many applications, e.g. CO_2 capture/storage
- Darcy's law for an incompressible fluid leads to the diffusion equation

$$-\nabla \cdot (k(x)\nabla p(x)) = g(x), \qquad x \in D \subseteq \mathbb{R}^3,$$

  with hydraulic conductivity k, source/sink terms g, and resulting pressure head p of groundwater
- Can be extended to multi-phase, time-dependent flow...



Motivation: Application in Groundwater Flow II

Challenge:

- To simulate groundwater flow using the Darcy model, we need to know the conductivity k in the entire domain D!
- Typically we only have sparse, noisy data available of k and p, which means there is significant uncertainty in k.

Aim:

- We will in this talk focus on combining expert prior knowledge with available data to infer k in a Bayesian statistical framework.
- A typical parametrisation chosen for k is piece-wise constant, corresponding to different rock types. In particular, we want to infer the values $\theta = \{\theta_1, \ldots, \theta_{d_\theta}\}$ that k takes in D.

Bayesian inverse problems (see e.g. [Kaipio, Somersalo '04])

- We choose a prior density π_0 on θ, incorporating any expert knowledge such as the hydraulic conductivity being positive.

- Using Bayes' Theorem, we obtain the posterior density π^y on $\theta|y$ given by

$$\underbrace{\pi^y(\theta)}_{\text{posterior on }\theta} \quad \eqsim \underbrace{\exp\Big(-\frac{1}{2\gamma^2}\|y-\mathcal{G}(\theta)\|_2^2\Big)}_{\text{likelihood on }y} \underbrace{\pi_0(\theta)}_{\text{prior on }\theta},$$

where

- y is the observed data, e.g. noisy point values of p given by $y = \{p(x_i;\theta)+\eta_i\}_{i=1}^{d_y} =: \mathcal{G}(\theta)+\eta$ and $\eta_i \sim N(0,\gamma^2\mathrm{I})$
- $\mathcal{G}$ is the parameter-to-observation map, which involves the solution operator of the PDE,
- the prior incorporates expert knowledge, the likelihood fits to the observed data, and the posterior is a combination of both.

Bayesian inverse problems II

Computational challenges:

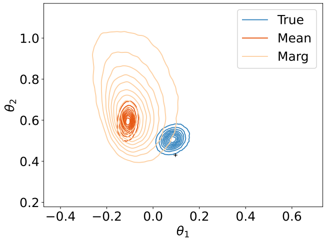
- Algorithms to compute the posterior density, such as Markov chain Monte Carlo, require repeated evaluation of the likelihood, often $10^4 - 10^6$ evaluations in practical applications.
- The evaluation of the likelihood is very costly, since $\mathcal{G}$ involves the solution of a partial differential equation.

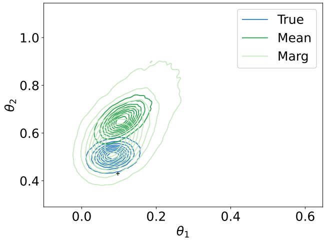
A solution:

- To make computations feasible, we approximate $\mathcal{G}$ by a surrogate model (emulator, reduced order model, meta model...).
- This could be e.g. a coarse numerical solver, or a neural network approximation, or ....
- This leads to an approximate posterior distribution that is feasible to sample from, BUT we have to consider its accuracy!

Bayesian inverse problems III (see e.g. [Bai, T, Zygalakis '24])

- Simply plugging the surrogate model into the posterior typically leads to biased and overconfident predictions.
- We need surrogate models that are:
  - uncertainty aware, to have the right level of confidence,
  - PDE-informed, to improve accuracy and avoid spurious predictions.
- We do this with:
  - PDE-constrained Gaussian processes, or
  - PDE-constrained Bayesian neural networks.



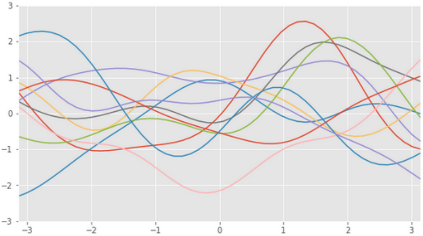


Example: 1d diffusion equation with $d_\theta = 2$

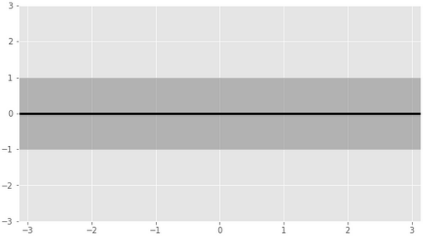
Gaussian process regression

Set-up [Rasmussen, Williams '06]

- Gaussian process regression is a Bayesian methodology to emulate a function $f : \mathcal{T} \to \mathbb{R}$, e.g. $f = \Phi$ or $f = \mathcal{G}_j$, $j = 1, \ldots, d_y$.
- We put a Gaussian process prior $\mathrm{GP}(0, k)$ on f, where k is chosen to reflect properties of f.
  For $\{\theta_i\}_{i=1}^m \subseteq \mathcal{T}$, the random variables $\{f(\theta_i)\}_{i=1}^m$ follow a joint Gaussian distribution with $\mathbb{E}[f(\theta_i)] = 0$ and $\mathbb{C}[f(\theta_i), f(\theta_j)] = k(\theta_i, \theta_j)$.



Sample paths



Mean and standard deviation

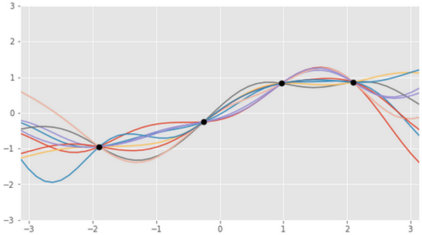
Gaussian process regression

Predictive distribution [Rasmussen, Williams '06]

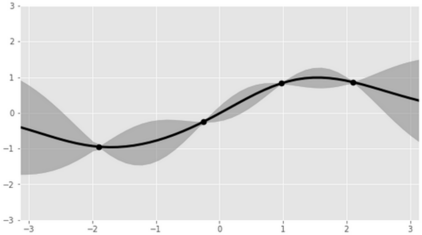
- The Gaussian process posterior $f_N \sim \mathrm{GP}(m_N^f, k_N)$ on $f|d$ is obtained by conditioning the prior on function values $d = \{\theta^n, f(\theta^n)\}_{n=1}^N$:

$$m_N^f(\theta) = k(\theta, D_N)^T K(D_N, D_N)^{-1} f(D_N),$$
$$k_N(\theta, \theta') = k(\theta, \theta') - k(\theta, D_N)^T K(D_N, D_N)^{-1} k(\theta', D_N),$$

where $D_N = \{\theta^n\}_{n=1}^N$, $k(\theta, D_N) = [k(\theta, \theta^1), \ldots, k(\theta, \theta^N)] \in \mathbb{R}^N$ and $K(D_N, D_N) \in \mathbb{R}^{N \times N}$ is the matrix with ij^{th} entry equal to $k(\theta^i, \theta^j)$.



Sample paths



Mean and standard deviation

Gaussian process regression

Approximations of the posterior

- Recall: $\pi^y(\theta) = \frac{1}{Z} \exp\big(- \frac{1}{2\gamma^2} \|y - \mathcal{G}(\theta)\|^2 \big) \pi_0(\theta)$

- For the remainder of the talk, assume that we approximate $\mathcal{G}$ by Gaussian process regression. Similar results hold for $\Phi = \frac{1}{2\gamma^2} \|y - \mathcal{G}(\theta)\|^2$.

- Since the surrogate model $\mathcal{G}_N$ is a stochastic process, a deterministic approximation of π^y is obtained:
    - by taking the mean-based approximation

$$\pi^y_{N,\mathrm{mean}}(\theta) = \frac{1}{Z_N^{\mathrm{mean}}} \exp\Big(- \frac{1}{2} \|y - m_N^{\mathcal{G}}(\theta)\|^2_{\Gamma^{-1}} \Big) \pi_0(\theta),$$

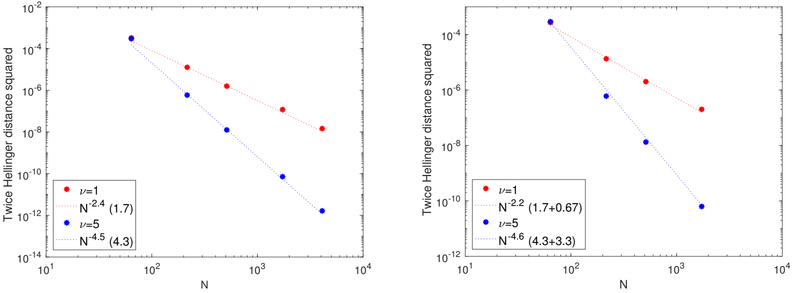
    - or by taking the marginal approximation

$$\pi^y_{N,\mathrm{marg}}(\theta) = \frac{1}{\mathbb{E}(Z_N^{\mathrm{rand}})} \mathbb{E}\left(\exp\Big(- \frac{1}{2} \|y - \mathcal{G}_N(\theta)\|^2_{\Gamma^{-1}} \Big) \right) \pi_0(\theta).$$

Gaussian process regression

Convergence as $N \to \infty$ [Stuart, T '18], [T 20], [Helin, Stuart, T, Zygalakis '23]

- Both approximations converge to the true posterior π^y as $N \to \infty$.
  Example: 1d diffusion equation with $d_\theta = 3$



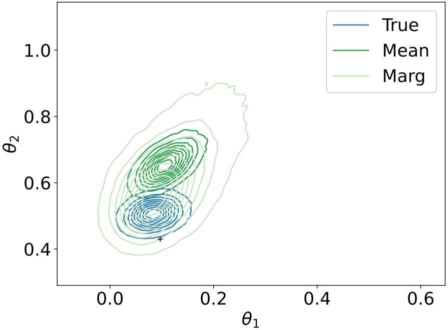
Left: Mean-based approximation. Right: Marginal approximation.

- Error in Hellinger distance depends on $\left\|\mathcal{G} - m_N^{\mathcal{G}}\right\|_{L^2(\mathcal{T};\mathbb{R}^{d_y})}$ and $\left\|\mathbb{E}\left(\|\mathcal{G} - \mathcal{G}_N\|^{1+\delta}\right)^{\frac{1}{1+\delta}}\right\|_{L^2(\mathcal{T})}$ for any $\delta > 0$, respectively.

Gaussian process regression

Mean-based and marginal approximations [Bai, T, Zygalakis '24]

- For small N, the difference between $\pi^y_{N,\mathrm{mean}}$ and $\pi^y_{N,\mathrm{marg}}$ can be significant.
- Using $\pi^y_{N,\mathrm{mean}}$ can lead to biased predictions with high confidence.
- Only $\pi^y_{N,\mathrm{marg}}$ uses the uncertainty in $\mathcal{G}_N$, modelling the error in the surrogate model.



Example: 1d diffusion equation with $d_\theta = 2$ and $N = 4$

Gaussian process regression

Marginal likelihood [Bai, T, Zygalakis '24]

- With $\mathcal{G}_N \sim \mathrm{GP}(m_N^{\mathcal{G}}, k_N)$, we can analytically compute the marginal likelihood $\mathbb{E}\Big(\exp\big(-\frac{1}{2\gamma^2}\|y - \mathcal{G}_N(\theta)\|^2\big)\Big)$.
- We have $\mathcal{G}_N(\theta) = m_N^{\mathcal{G}}(\theta) + \xi$, with $\xi \sim \mathrm{N}(0, k_N(\theta, \theta))$. Hence

$$
\begin{aligned}
&\mathbb{E}\left(\exp\big(-\frac{1}{2}\|y - \mathcal{G}_N(\theta)\|_{\Gamma^{-1}}^2\big)\right) \\
&= \frac{1}{\sqrt{(2\pi)^{d_y}\det(\Sigma(\theta))}} \int_{\mathbb{R}^{d_y}} \exp\left(-\frac{||y - m_N^{\mathcal{G}}(\theta) - \xi||_{\Gamma^{-1}}^2}{2}\right) \exp\left(-\frac{||\xi||_{\Sigma^{-1}(u)}^2}{2}\right) d\xi \\
&\propto \frac{1}{\sqrt{\det(\Gamma + \Sigma(\theta))}} \exp\left(-\frac{||y - m_N^{\mathcal{G}}(\theta)||_{(\Gamma + \Sigma(\theta))^{-1}}^2}{2}\right),
\end{aligned}
$$

where $\Sigma(\theta) = k_N(\theta, \theta)$.

Gaussian process regression

Variance inflation

- Compared to the mean-based likelihood

$$\frac{1}{\sqrt{\det(\Gamma)}} \exp\left(-\frac{||y - m_N^{\mathcal{G}}(\theta)||_{\Gamma^{-1}}^2}{2}\right),$$

the marginal likelihood

$$\frac{1}{\sqrt{\det(\Gamma + \Sigma(\theta))}} \exp\left(-\frac{||y - m_N^{\mathcal{G}}(\theta)||_{(\Gamma+\Sigma(\theta))^{-1}}^2}{2}\right),$$

is a form of variance inflation.

Gaussian process regression

Variance inflation II

- Variance inflation is an emerging tool to improve Bayesian inference in complex models, see e.g. [Conrad et al '17], [Calvetti et al '18], [Cui, Fox, Neumayer '20].

- It is closely related to the well-established inclusion of modelling error [Kennedy, O'Hagan '01]:

$$y = \mathcal{G}(\theta) + \eta + \tilde{\eta},$$

  with $\tilde{\eta} \sim \mathrm{N}(m, C)$.

- Using Gaussian process regression, we have
  - a parameter-dependent variance inflation $\Sigma(\theta)$, rather than assuming that the error in the (surrogate) model is independent of θ.
  - an explicit model for $\Sigma(\theta)$ that is readily tuned.

Markov chain Monte Carlo methods

- In practice, we need to use sampling methods such as MCMC to sample from target density $\pi = \pi^y_{N,\mathrm{mean}}$ or $\pi = \pi^y_{N,\mathrm{marg}}$.

ALGORITHM 1. (Metropolis Hastings)

- Choose $\theta^{(1)}$ with $\pi(\theta^{(1)}) > 0$.
- At state $\theta^{(i)}$, sample a proposal θ' from density $q(\theta' \,|\, \theta^{(i)})$.
- Accept sample θ' with probability

$$\alpha(\theta' \,|\, \theta^{(i)}) = \min\left(1, \frac{\pi(\theta')\, q(\theta^{(i)} \,|\, \theta')}{\pi(\theta^{(i)})\, q(\theta' \,|\, \theta^{(i)})}\right),$$

i.e. $\theta^{(i+1)} = \theta'$ with probability $\alpha(\theta' \,|\, \theta^{(i)})$; otherwise stay at $\theta^{(i+1)} = \theta^{(i)}$.

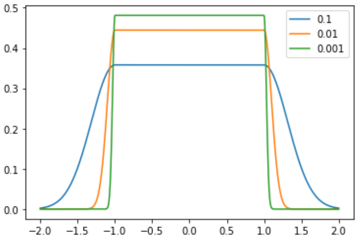
Gaussian process regression

MALA proposals [Roberts, Tweedie '96], [Bai, T, Zygalakis '24]

- In inverse problems, we often have high dimensional parameters θ, and we require a an efficient choice of proposals such as MALA:

$$\theta' = \theta^{(i)} + \beta \nabla \log \pi(\theta^{(i)}) + \sqrt{2\beta}\xi_i, \qquad \text{where} \qquad \xi_i \sim \mathcal{N}(0, \mathrm{I})$$

- For $\pi = \pi^y_{N,\mathrm{mean}}$ and $\pi = \pi^y_{N,\mathrm{marg}}$, the gradient of the log-likelihood exists provided $k(\cdot, \theta^n)$ is differentiable.
- For common choices of k, e.g. $k(\theta, \theta') = \sigma^2 \exp(-\frac{\|\theta - \theta'\|_2^2}{2\lambda^2})$, the gradient can be computed explicitly.
- Some priors, such as the uniform prior, require smoothing using Moreau–Yoshida regularisation [Pereyra '16].



Gaussian process regression

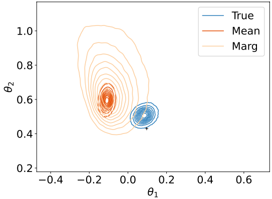
PDE constraints [Bai, T, Zygalakis '24], [Raissi, Perdikaris, Karniadakis '17]

For linear PDEs $\mathcal{L}_\theta\, p(x;\theta) = g(x)$, such as the diffusion equation with $\mathcal{L}_\theta\, p(x;\theta) = -\nabla \cdot (k(x;\theta)\nabla p(x;\theta))$, we can incorporate $\mathcal{L}_\theta$:

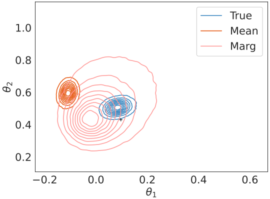
- Put a joint Gaussian process prior on p and g as a function of θ:

$$\begin{bmatrix} p(x_p,\theta) \\ g(x_g,\theta) \end{bmatrix} \sim \mathrm{GP}\left(0, k_1(\theta,\theta') \begin{bmatrix} k_2(x_p,x_p) & \mathcal{L}^{\theta'} k_2(x_p,x_g) \\ \mathcal{L}^{\theta} k_2(x_p,x_g) & \mathcal{L}^{\theta}\mathcal{L}^{\theta'} k_2(x_g,x_g) \end{bmatrix}\right).$$

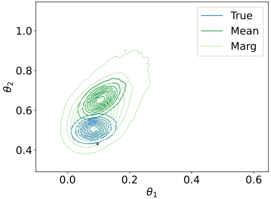
- This gives a joint prior on $\{p(x_i,\theta)\} = \mathcal{G}(\theta)$ and $\{g(\tilde{x}_j,\theta)\}$.
- We update this to a posterior by conditioning on training data in the usual way, and then use the marginal posterior on p.



Independent



Spatially correlated



PDE-constrained

Bayesian neural networks

PDE constraints [Jiminez Beltran et al, '24], [Sirignana, Spiliopoulos '18]

- The deep Galerkin method provides a neural network that
  - takes as inputs x in the spatial domain and parameter value θ,
  - is trained to approximate the solution of the PDE, $f_{\mathrm{W}}(x;\theta) \approx p(x;\theta)$.
- In particular, f_{W} is trained to minimize the following loss:

$$\frac{1}{K}\sum_{i=1}^{K} \ell(D_i;\mathrm{W}) = \frac{1}{K}\sum_{i=1}^{K} \left(\ell_g(D_{g,i};\mathrm{W}) + \ell_b(D_{b,i};\mathrm{W})\right),$$

where
  - at each iteration in training, K collocations for $x_i, \partial x_i$, and θ_i are sampled from densities π^p, π^b, and π^θ, respectively,
  - $\{D_i\} = \{D_{g,i}\} \cup \{D_{b,i}\}$, $D_{g,i} = \{x_i, \theta_i\}$ and $D_{b,i} = \{\partial x_i, \theta_i\}$,
  - $\ell_h(D_{g,i};\mathrm{W}) = \left(\mathcal{L}_{\theta_i}(x_i, f_{\mathrm{W}}(x_i,\theta_i);\theta_i) - g(x_i;\theta_i)\right)^2$ measures the error in the approximation of the differential operator, and
  - $\ell_b(D_{b,i};\mathrm{W}) = \left(f_{\mathrm{W}}(\partial x_i, \theta_i) - b(\partial x_i;\theta_i)\right)^2$ measures the error in the boundary conditions.

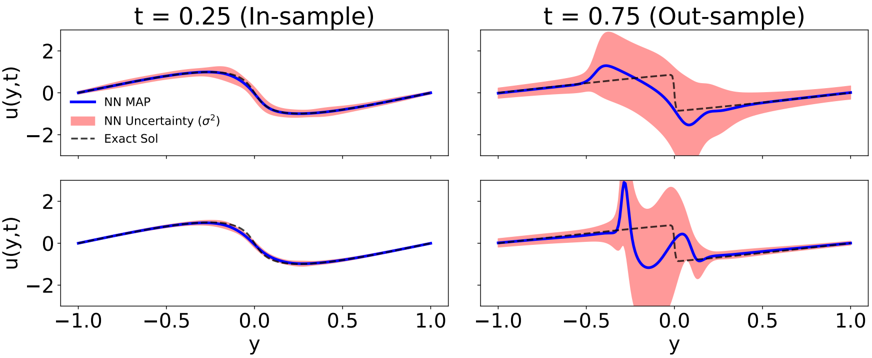
Bayesian neural networks

PDE constraints + Bayesian [Jiminez Beltran et al, '24]

- To provide uncertainty quantification, we interpret the loss as a negative log-likelihood.
- By doing this, the likelihood $p(D|\mathrm{W})$ characterises how well a choice of weights W approximates the PDE solution $p(x;\theta)$, and we obtain a corresponding posterior $p(\mathrm{W}|D)$ on the weights.
- For computational efficiency, we:
    - only consider weights in the penultimate layer of the neural network as random, and
    - use a Laplace approximation of the posterior $p(\mathrm{W}|D)$,

  which is generally enough to deliver good uncertainty estimates.

Numerical example [Jiminez Beltran et al '24]



Burgers Equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial y} - \theta\frac{\partial^2 u}{\partial y^2} = 0, \quad t > 0, \quad \theta > 0 \quad \text{and} \quad y \in [-1, 1],$$

with conditions $u(y, 0) = -\sin(\pi y)$, and $u(-1, t) = u(1, t) = 0$.

Conclusions

- Partial differential equations model many phenomena in science and engineering.
- Many tasks, such as parameter inference in partial differential equation models, can quickly become infeasible.
- It is common to use surrogate models to alleviate the computational burden.
- However, to avoid overconfident and biased predictions, it is crucial to use surrogate models that:
  - come with uncertainty quantification, and
  - incorporate physical constraints.

References I

T. Bai, A. L. Teckentrup, and K. C. Zygalakis, *Gaussian processes for Bayesian inverse problems associated with linear partial differential equations*, Statistics and Computing, 34 (2024), p. 139.

C. Jiminez Beltran, A. Vergari, A. L. Teckentrup, and K. C. Zygalakis, *Galerkin meets Laplace: Fast uncertainty estimation in neural PDEs*, in ICLR 2024 Workshop on AI4DifferentialEquationsInScience, 2024.

J. Kaipio and E. Somersalo, *Statistical and computational inverse problems*, Springer, 2004.

M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Machine learning of linear differential equations using Gaussian processes*, Journal of Computational Physics, 348 (2017), pp. 683–693.

C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, (2006).

J. Sirignano and K. Spiliopoulos, *DGM: A deep learning algorithm for solving partial differential equations*, Journal of computational physics, 375 (2018), pp. 1339–1364.