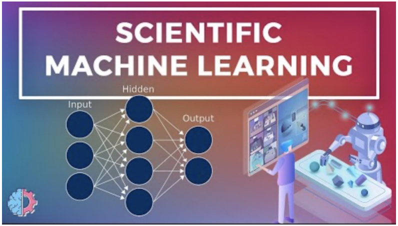
Developments of Multiscale and Probabilistic Methods
for Solving PDEs and Inverse Problems

Yifan Chen, Caltech

Peking University, Feb, 2023

Scientific Computing and Learning



modeling, data, decision-making, ...

plenty of amazing things

simulation, prediction, design, ...

Scientific Computing and Learning

Mathematical Challenges:

Solving equations

- multiscale physics
- heterogeneous material
- large scale PDEs ...

Need many degrees of freedom for enough accuracy

Learning solutions

- trials and errors
- training
- uncertainties ...

Sometimes machine learning automation may not be robust

Research Goal: further accuracy and more reliable automation

Scientific Computing and Learning

Mathematical Challenges:

Solving equations

- multiscale physics
- heterogeneous material
- large scale PDEs ...

Need many degrees of freedom for enough accuracy

Learning solutions

- trials and errors
- training
- uncertainties ...

Sometimes machine learning automation may not be robust

Research Goal: further accuracy and more reliable automation

Outline

1. Exponentially Convergent Multiscale Methods for PDEs
   "how to get very accurate solutions via multiscale analysis"

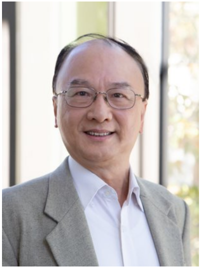
2. Gaussian Processes for PDEs and Inverse Problems
   "how to get reliable automated solutions via Bayes inference"

Outline

1. Exponentially Convergent Multiscale Methods for PDEs
   "how to get very accurate solutions via multiscale analysis"

2. Gaussian Processes for PDEs and Inverse Problems
   "how to get reliable automated solutions via Bayes inference"

Part I: Exponentially Convergent Multiscale Methods



Thomas Y. Hou
Caltech

Yixuan Wang
Caltech

Solving Multiscale PDEs

Model Problem:

$$\boxed{-\nabla \cdot (A\nabla u) + Vu = f, \text{ in } \Omega, \quad \text{w/ boundary conditions}}$$

(subsurface flows, diffusions, elasticity, waves in *composite* media)

Mathematical Condition:

- heterogeneity: $A, V \in L^\infty(\Omega)$ (no scale separation)
  $0 < A_{\min} \le A(x) \le A_{\max} < \infty$
- high frequency: e.g., $V = -k^2$ (Helmholtz's equation)
- regularity of force: $f \in L^2(\Omega)$

Solving Multiscale PDEs

Model Problem:

$$\boxed{-\nabla \cdot (A\nabla u) + Vu = f, \text{ in } \Omega, \quad \text{w/ boundary conditions}}$$

(subsurface flows, diffusions, elasticity, waves in *composite* media)

Mathematical Condition:

- heterogeneity: $A, V \in L^\infty(\Omega)$ (no scale separation)
  $0 < A_{\min} \leq A(x) \leq A_{\max} < \infty$
- high frequency: e.g., $V = -k^2$ (Helmholtz's equation)
- regularity of force: $f \in L^2(\Omega)$

Galerkin's Method:

- find a space S of basis functions to approximate the solution
- quasi-optimality: solution err $\sim$ approximation err

Challenges:

- heterogeneity $\Rightarrow u$ is oscillatory
  (!) approx-err of FEM can be arbitrarily bad [Babuška, Osborn 2000]
- high frequency $\Rightarrow$ stability issues[1]

  example: $\|u\|_{\mathcal{H}(\Omega)} \leq C_{\mathsf{stab}}(k)\|f\|_{L^2(\Omega)}$ for $C_{\mathsf{stab}}(k) \succeq 1 + k^\gamma$

  (!) approx-err amplified; quasi-optimality also deteriorates
  known as pollution effect [Babuška, Sauter, 1997]

[1] $\mathcal{H}(\Omega)$ is the energy norm

Numerical Challenges

Galerkin's Method:

- find a space S of basis functions to approximate the solution
- quasi-optimality: solution err $\sim$ approximation err

Challenges:

- heterogeneity $\Rightarrow u$ is oscillatory
  (!) approx-err of FEM can be arbitrarily bad [Babuška, Osborn 2000]
- high frequency $\Rightarrow$ stability issues[1]

  example: $\|u\|_{\mathcal{H}(\Omega)} \leq C_{\mathsf{stab}}(k)\|f\|_{L^2(\Omega)}$ for $C_{\mathsf{stab}}(k) \succeq 1 + k^\gamma$

  (!) approx-err amplified; quasi-optimality also deteriorates
  known as pollution effect [Babuška, Sauter, 1997]

[1] $\mathcal{H}(\Omega)$ is the energy norm

Numerical Challenges

Galerkin's Method:

- find a space S of basis functions to approximate the solution
- quasi-optimality: solution err $\sim$ approximation err

Challenges:

- heterogeneity $\Rightarrow$ u is oscillatory
  (!) approx-err of FEM can be arbitrarily bad [Babuška, Osborn 2000]
- high frequency $\Rightarrow$ stability issues[1]

  example: $\|u\|_{\mathcal{H}(\Omega)} \leq C_{\mathsf{stab}}(k)\|f\|_{L^2(\Omega)}$ for $C_{\mathsf{stab}}(k) \succeq 1 + k^\gamma$

  (!) approx-err amplified; quasi-optimality also deteriorates
  known as pollution effect [Babuška, Sauter, 1997]

[1] $\mathcal{H}(\Omega)$ is the energy norm

Multiscale Methods / Numerical Homogenization / ...

Idea: find better basis functions adapted to A and V

- tremendous literature with different constructions (find structures) (hp-FEM, GFEM, MsFEM, HMM, VMS, LOD, ...)

Our Focus: push approximation err further, for exponential convergence

- previous work for elliptic eqns based on GFEM [Babuška, Lipton 2011][2]

Our contribution: ExpMsFEM [Chen, Hou, Wang 2021,2021,2022]

A general multiscale framework for elliptic and Helmholtz eqns

[2] further generalization to Helmholtz eqns [Ma, Alber, Scheichl 2021]

Multiscale Methods / Numerical Homogenization / ...

Idea: find better basis functions adapted to A and V

- tremendous literature with different constructions (find structures) (hp-FEM, GFEM, MsFEM, HMM, VMS, LOD, ...)

Our Focus: push approximation err further, for exponential convergence

- previous work for elliptic eqns based on GFEM [Babuška, Lipton 2011][2]

Our contribution: ExpMsFEM [Chen, Hou, Wang 2021,2021,2022]

A general multiscale framework for elliptic and Helmholtz eqns

[2] further generalization to Helmholtz eqns [Ma, Alber, Scheichl 2021]

How Could Exponential Convergence Be Achieved?

Principle: how exponential convergence possible for nonsmooth funcs?

- coarse-fine scale decomposition: diff-scales treated differently
- localize the approximation for both the coarse and fine components
- find low complexity structures of the coarse scale component

Instantiation in ExpMsFEM for finding exp-convergent representation

1. generalized harmonic-bubble splitting
2. edge localization
3. oversampling and exponentially decaying spectral problems

How Could Exponential Convergence Be Achieved?

Principle: how exponential convergence possible for nonsmooth funcs?

- coarse-fine scale decomposition: diff-scales treated differently
- localize the approximation for both the coarse and fine components
- find low complexity structures of the coarse scale component

Instantiation in ExpMsFEM for finding exp-convergent representation

1. generalized harmonic-bubble splitting
2. edge localization
3. oversampling and exponentially decaying spectral problems

How Could Exponential Convergence Be Achieved?

Principle: how exponential convergence possible for nonsmooth funcs?

- coarse-fine scale decomposition: diff-scales treated differently
- localize the approximation for both the coarse and fine components
- find low complexity structures of the coarse scale component

Instantiation in ExpMsFEM for finding exp-convergent representation

1. generalized harmonic-bubble splitting
2. edge localization
3. oversampling and exponentially decaying spectral problems

How Could Exponential Convergence Be Achieved?

Principle: how exponential convergence possible for nonsmooth funcs?

- coarse-fine scale decomposition: diff-scales treated differently
- localize the approximation for both the coarse and fine components
- find low complexity structures of the coarse scale component

Instantiation in ExpMsFEM for finding exp-convergent representation

1. generalized harmonic-bubble splitting
2. edge localization
3. oversampling and exponentially decaying spectral problems

How Could Exponential Convergence Be Achieved?

Principle: how exponential convergence possible for nonsmooth funcs?

- coarse-fine scale decomposition: diff-scales treated differently
- localize the approximation for both the coarse and fine components
- find low complexity structures of the coarse scale component

Instantiation in ExpMsFEM for finding exp-convergent representation

1. generalized harmonic-bubble splitting
2. edge localization
3. oversampling and exponentially decaying spectral problems

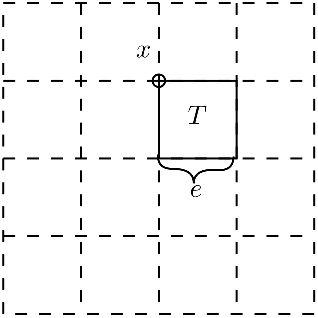
Step 1: Generalized Harmonic-bubble Splitting[3]

- mesh: $H = O(1/k)$
- split the solution locally:
  in each T, $u = u_T^{\mathsf{h}} + u_T^{\mathsf{b}}$

$$\begin{cases} -\nabla \cdot (A \nabla u_T^{\mathsf{h}}) + V u_T^{\mathsf{h}} = 0, & \text{in } T \\ u_T^{\mathsf{h}} = u, & \text{on } \partial T \end{cases}$$

$$\begin{cases} -\nabla \cdot (A \nabla u_T^{\mathsf{b}}) + V u_T^{\mathsf{b}} = f, & \text{in } T \\ u_T^{\mathsf{b}} = 0, & \text{on } \partial T \end{cases}$$

- global function:
  $u^{\mathsf{h}}(x) = u_T^{\mathsf{h}}(x)$ locally "harmonic"
  $u^{\mathsf{b}}(x) = u_T^{\mathsf{b}}(x)$ locally computable
  when $x \in T$ for each T



$x \in \mathcal{N}_H, e \in \mathcal{E}_H, T \in \mathcal{T}_H$

$$\boxed{\text{First Decomposition} : u = u^{\mathsf{h}} + u^{\mathsf{b}}}$$

[3] [Hetmaniuk, Lehoucq 2010], [Hou, Liu 2016]

Step 2: Edge Localization

Recall $u = u^{\mathsf{h}} + u^{\mathsf{b}}$ "locally harmonic + locally computable"

$$u^{\mathsf{h}} = Q\tilde{u} \qquad (Q\text{: "harmonic" extension operator; } \tilde{u} = u|_{\text{edges}})$$

$$= Q(\tilde{u} - I_H\tilde{u}) + QI_H\tilde{u} \qquad (I_H\text{: nodal interpolation on edges})$$

$$= Q(\tilde{u} - I_H\tilde{u}) + \sum_{x_i \in \mathcal{N}_H} u(x_i)\psi_i$$

(ψ_i: basis funcs in MsFEM [Hou, Wu 1997])

$$= \sum_{e \in \mathcal{E}_H} QR_e u + \sum_{x_i \in \mathcal{N}_H} u(x_i)\psi_i \qquad (R_e u = (\tilde{u} - I_H\tilde{u})|_e)$$

$u^{\mathsf{h}} =$ "sum of terms dependent on each edge + a term represented by ψ_i"

Step 2: Edge Localization

Recall $u = u^{\mathsf{h}} + u^{\mathsf{b}}$ "locally harmonic + locally computable"

$$
\begin{aligned}
u^{\mathsf{h}} &= Q\tilde{u} && (Q\text{: "harmonic" extension operator; } \tilde{u} = u|_{\text{edges}}) \\
&= Q(\tilde{u} - I_H\tilde{u}) + QI_H\tilde{u} && (I_H\text{: nodal interpolation on edges}) \\
&= Q(\tilde{u} - I_H\tilde{u}) + \sum_{x_i \in \mathcal{N}_H} u(x_i)\psi_i \\
& && (\psi_i\text{: basis funcs in MsFEM [Hou, Wu 1997]}) \\
&= \sum_{e \in \mathcal{E}_H} QR_e u + \sum_{x_i \in \mathcal{N}_H} u(x_i)\psi_i && (R_e u = (\tilde{u} - I_H\tilde{u})|_e)
\end{aligned}
$$

u^{h} = "sum of terms dependent on each edge + a term represented by ψ_i"

Step 2: Edge Localization

Recall $u = u^{\mathsf{h}} + u^{\mathsf{b}}$ "locally harmonic + locally computable"

$$\begin{aligned} u^{\mathsf{h}} &= Q\tilde{u} && (Q\text{: "harmonic" extension operator; } \tilde{u} = u|_{\text{edges}}) \\ &= Q(\tilde{u} - I_H\tilde{u}) + QI_H\tilde{u} && (I_H\text{: nodal interpolation on edges}) \\ &= Q(\tilde{u} - I_H\tilde{u}) + \sum_{x_i \in \mathcal{N}_H} u(x_i)\psi_i \\ & && (\psi_i\text{: basis funcs in MsFEM [Hou, Wu 1997]}) \\ &= \sum_{e \in \mathcal{E}_H} QR_e u + \sum_{x_i \in \mathcal{N}_H} u(x_i)\psi_i && (R_e u = (\tilde{u} - I_H\tilde{u})|_e) \end{aligned}$$

u^{h} = "sum of terms dependent on each edge + a term represented by ψ_i"

Step 2: Edge Localization

Recall $u = u^{\mathsf{h}} + u^{\mathsf{b}}$ "locally harmonic + locally computable"

$$
\begin{aligned}
u^{\mathsf{h}} &= Q\tilde{u} && (Q\text{: "harmonic" extension operator; } \tilde{u} = u|_{\text{edges}}) \\
&= Q(\tilde{u} - I_H\tilde{u}) + QI_H\tilde{u} && (I_H\text{: nodal interpolation on edges}) \\
&= Q(\tilde{u} - I_H\tilde{u}) + \sum_{x_i \in \mathcal{N}_H} u(x_i)\psi_i \\
& && (\psi_i\text{: basis funcs in MsFEM [Hou, Wu 1997]}) \\
&= \sum_{e \in \mathcal{E}_H} QR_e u + \sum_{x_i \in \mathcal{N}_H} u(x_i)\psi_i && (R_e u = (\tilde{u} - I_H\tilde{u})|_e)
\end{aligned}
$$

$u^{\mathsf{h}} =$ "sum of terms dependent on each edge + a term represented by ψ_i"

Step 3: Oversampling[4] and Low Complexity Structure

Oversampling: consider $e \subset \omega_e$

$$\text{on } e: \; QR_e u = QR_e(u|_{\omega_e}) = QR_e u^{\mathsf{h}}_{\omega_e} + QR_e u^{\mathsf{b}}_{\omega_e}$$

Here, $u^{\mathsf{h}}_{\omega_e}, u^{\mathsf{b}}_{\omega_e}$: oversampling harmonic / bubble part in ω_e

Recall the definition:

$$\begin{cases} -\nabla \cdot (A \nabla u^{\mathsf{h}}_{\omega_e}) + V u^{\mathsf{h}}_{\omega_e} = 0, & \text{in } \omega_e \\ u^{\mathsf{h}}_{\omega_e} = u, & \text{on } \partial \omega_e \end{cases}$$

$$\begin{cases} -\nabla \cdot (A \nabla u^{\mathsf{b}}_{\omega_e}) + V u^{\mathsf{b}}_{\omega_e} = f, & \text{in } \omega_e \\ u^{\mathsf{b}}_{\omega_e} = 0, & \text{on } \partial \omega_e \end{cases}$$

interior edge

$QR_e u =$ "restriction of local harmonic funcs + locally computable"

[4] historically proposed in [Hou, Wu 1997] to reduce the resonance error in MsFEM

Step 3: Oversampling[4] and Low Complexity Structure

Oversampling: consider $e \subset \omega_e$

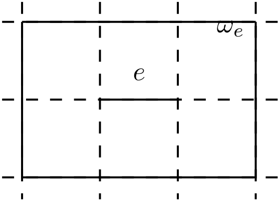
$$\text{on } e: \ QR_e u = QR_e(u|_{\omega_e}) = QR_e u^{\mathsf{h}}_{\omega_e} + QR_e u^{\mathsf{b}}_{\omega_e}$$

Here, $u^{\mathsf{h}}_{\omega_e}, u^{\mathsf{b}}_{\omega_e}$: oversampling harmonic / bubble part in ω_e

Recall the definition:

$$\begin{cases} -\nabla \cdot (A \nabla u^{\mathsf{h}}_{\omega_e}) + V u^{\mathsf{h}}_{\omega_e} = 0, & \text{in } \omega_e \\ u^{\mathsf{h}}_{\omega_e} = u, & \text{on } \partial\omega_e \end{cases}$$

$$\begin{cases} -\nabla \cdot (A \nabla u^{\mathsf{b}}_{\omega_e}) + V u^{\mathsf{b}}_{\omega_e} = f, & \text{in } \omega_e \\ u^{\mathsf{b}}_{\omega_e} = 0, & \text{on } \partial\omega_e \end{cases}$$



interior edge

$QR_e u =$ "restriction of local harmonic funcs + locally computable"

[4] historically proposed in [Hou, Wu 1997] to reduce the resonance error in MsFEM

Step 3: Oversampling[4] and Low Complexity Structure

Oversampling: consider $e \subset \omega_e$

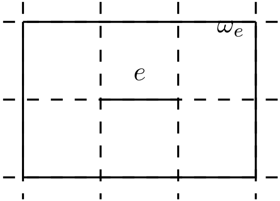
$$\text{on } e: \ QR_e u = QR_e(u|_{\omega_e}) = QR_e u^{\mathsf{h}}_{\omega_e} + QR_e u^{\mathsf{b}}_{\omega_e}$$

Here, $u^{\mathsf{h}}_{\omega_e}, u^{\mathsf{b}}_{\omega_e}$: oversampling harmonic / bubble part in ω_e

Recall the definition:

$$\begin{cases} -\nabla \cdot (A \nabla u^{\mathsf{h}}_{\omega_e}) + V u^{\mathsf{h}}_{\omega_e} = 0, & \text{in } \omega_e \\ u^{\mathsf{h}}_{\omega_e} = u, & \text{on } \partial\omega_e \end{cases}$$

$$\begin{cases} -\nabla \cdot (A \nabla u^{\mathsf{b}}_{\omega_e}) + V u^{\mathsf{b}}_{\omega_e} = f, & \text{in } \omega_e \\ u^{\mathsf{b}}_{\omega_e} = 0, & \text{on } \partial\omega_e \end{cases}$$



interior edge

$QR_e u =$ "restriction of local harmonic funcs + locally computable"

[4] historically proposed in [Hou, Wu 1997] to reduce the resonance error in MsFEM

Step 3: Oversampling and Low Complexity Structure

Low complexity: restriction of "harmonic" funcs [Babuška, Lipton 2011]

- generalize to our context: singular values of the operator

$$QR_e : (U(\omega_e), \|\cdot\|_{\mathcal{H}(\omega_e)}) \to (\mathcal{H}(\Omega), \|\cdot\|_{\mathcal{H}(\Omega)})$$

decay exponentially fast [Chen, Hou, Wang 2021] , where

$$U(\omega_e) := \{v \in \mathcal{H}(\omega_e) : -\nabla \cdot (A\nabla v) + Vv = 0, \text{ in } \omega_e\}$$

- equivalently, for $m > 0$, there exists $b_{e,j}, v_{e,j}, 1 \le j \le m$ s.t.

$$\|QR_e u^{\mathsf{h}}_{\omega_e} - \sum_{1\le j\le m} b_{e,j} v_{e,j}\|_{\mathcal{H}(\Omega)} \le C \exp\left(-bm^{\frac{1}{d+1}}\right) \|u^{\mathsf{h}}_{\omega_e}\|_{\mathcal{H}(\omega_e)}$$

$QR_e u^{\mathsf{h}}_{\omega_e} =$ "a term represented by m basis funcs + small err"

Step 3: Oversampling and Low Complexity Structure

Low complexity: restriction of "harmonic" funcs [Babuška, Lipton 2011]

- generalize to our context: singular values of the operator

$$QR_e : (U(\omega_e), \|\cdot\|_{\mathcal{H}(\omega_e)}) \to (\mathcal{H}(\Omega), \|\cdot\|_{\mathcal{H}(\Omega)})$$

decay exponentially fast [Chen, Hou, Wang 2021] , where

$$U(\omega_e) := \{v \in \mathcal{H}(\omega_e) : -\nabla \cdot (A\nabla v) + Vv = 0, \text{ in } \omega_e\}$$

- equivalently, for $m > 0$, there exists $b_{e,j}, v_{e,j}, 1 \le j \le m$ s.t.

$$\|QR_e u^{\mathsf{h}}_{\omega_e} - \sum_{1\le j\le m} b_{e,j} v_{e,j}\|_{\mathcal{H}(\Omega)} \le C \exp\left(-bm^{\frac{1}{d+1}}\right)\|u^{\mathsf{h}}_{\omega_e}\|_{\mathcal{H}(\omega_e)}$$

$QR_e u^{\mathsf{h}}_{\omega_e} =$ "a term represented by m basis funcs + small err"

Step 3: Oversampling and Low Complexity Structure

Low complexity: restriction of "harmonic" funcs [Babuška, Lipton 2011]

- generalize to our context: singular values of the operator

$$QR_e : (U(\omega_e), \|\cdot\|_{\mathcal{H}(\omega_e)}) \to (\mathcal{H}(\Omega), \|\cdot\|_{\mathcal{H}(\Omega)})$$

decay exponentially fast [Chen, Hou, Wang 2021] , where

$$U(\omega_e) := \{v \in \mathcal{H}(\omega_e) : -\nabla \cdot (A\nabla v) + Vv = 0, \text{ in } \omega_e\}$$

- equivalently, for $m > 0$, there exists $b_{e,j}, v_{e,j}, 1 \le j \le m$ s.t.

$$\|QR_e u^{\mathsf{h}}_{\omega_e} - \sum_{1\le j\le m} b_{e,j} v_{e,j}\|_{\mathcal{H}(\Omega)} \le C \exp\left(-bm^{\frac{1}{d+1}}\right) \|u^{\mathsf{h}}_{\omega_e}\|_{\mathcal{H}(\omega_e)}$$

$QR_e u^{\mathsf{h}}_{\omega_e} =$ "a term represented by m basis funcs + small err"

The Representation and Algorithm

$$u = \left(\sum_{e\in\mathcal{E}_H} \sum_{1\le j\le m} b_{e,j} v_{e,j} + \sum_{x_i\in\mathcal{N}_H} u(x_i)\psi_i \right) + \left(u^{\mathsf{b}} + \sum_{e\in\mathcal{E}_H} QR_e u^{\mathsf{b}}_{\omega_e} \right) + O\left(\exp\Big(-bm^{\frac{1}{d+1}}\Big)(\|u\|_{\mathcal{H}(\Omega)} + \|f\|_{L^2(\Omega)}) \right)$$

Offline: one-time model reduction

- compute $\{v_{e,j}\}, 1 \le j \le m$ for each e, and ψ_i for each node (local SVD and harmonic extension; parallelizable)

Online: efficient for multiple f

- compute $u^{\mathsf{n}} = u^{\mathsf{b}} + \sum_{e\in\mathcal{E}_H} QR_e u^{\mathsf{b}}_{\omega_e}$ (solve local equations involving f; parallelizable)
- Galerkin's method using basis funcs $\{v_{e,j}\}$ and ψ_i to solve

$$a(u_{H,m}, v) = (f, v)_\Omega - a(u^{\mathsf{n}}, v)$$

$$\|u - u_{H,m} - u^{\mathsf{n}}\|_{\mathcal{H}(\Omega)} = O\left(\exp\Big(-bm^{\frac{1}{d+1}}\Big)(\|u\|_{\mathcal{H}(\Omega)} + \|f\|_{L^2(\Omega)}) \right)$$

The Representation and Algorithm

$$u = \left(\sum_{e \in \mathcal{E}_H} \sum_{1 \le j \le m} b_{e,j} v_{e,j} + \sum_{x_i \in \mathcal{N}_H} u(x_i) \psi_i \right) + \left(u^{\mathsf{b}} + \sum_{e \in \mathcal{E}_H} Q R_e u^{\mathsf{b}}_{\omega_e} \right) + O\left(\exp\left(-b m^{\frac{1}{d+1}} \right) \left(\|u\|_{\mathcal{H}(\Omega)} + \|f\|_{L^2(\Omega)} \right) \right)$$

Offline: one-time model reduction

- compute $\{v_{e,j}\}, 1 \le j \le m$ for each e, and ψ_i for each node (local SVD and harmonic extension; parallelizable)

Online: efficient for multiple f

- compute $u^{\mathsf{n}} = u^{\mathsf{b}} + \sum_{e \in \mathcal{E}_H} Q R_e u^{\mathsf{b}}_{\omega_e}$ (solve local equations involving f; parallelizable)
- Galerkin's method using basis funcs $\{v_{e,j}\}$ and ψ_i to solve

$$a(u_{H,m}, v) = (f, v)_\Omega - a(u^{\mathsf{n}}, v)$$

$$\|u - u_{H,m} - u^{\mathsf{n}}\|_{\mathcal{H}(\Omega)} = O\left(\exp\left(-b m^{\frac{1}{d+1}} \right) \left(\|u\|_{\mathcal{H}(\Omega)} + \|f\|_{L^2(\Omega)} \right) \right)$$

The Representation and Algorithm

$$u = \left(\sum_{e \in \mathcal{E}_H} \sum_{1 \le j \le m} b_{e,j} v_{e,j} + \sum_{x_i \in \mathcal{N}_H} u(x_i) \psi_i \right) + \left(u^{\mathsf{b}} + \sum_{e \in \mathcal{E}_H} Q R_e u^{\mathsf{b}}_{\omega_e} \right) + O\left(\exp\left(-b m^{\frac{1}{d+1}} \right) (\|u\|_{\mathcal{H}(\Omega)} + \|f\|_{L^2(\Omega)}) \right)$$

Offline: one-time model reduction

- compute $\{v_{e,j}\}, 1 \le j \le m$ for each e, and ψ_i for each node (local SVD and harmonic extension; parallelizable)

Online: efficient for multiple f

- compute $u^{\mathsf{n}} = u^{\mathsf{b}} + \sum_{e \in \mathcal{E}_H} Q R_e u^{\mathsf{b}}_{\omega_e}$ (solve local equations involving f; parallelizable)
- Galerkin's method using basis funcs $\{v_{e,j}\}$ and ψ_i to solve

$$a(u_{H,m}, v) = (f, v)_\Omega - a(u^{\mathsf{n}}, v)$$

$$\|u - u_{H,m} - u^{\mathsf{n}}\|_{\mathcal{H}(\Omega)} = O\left(\exp\left(-b m^{\frac{1}{d+1}} \right) (\|u\|_{\mathcal{H}(\Omega)} + \|f\|_{L^2(\Omega)}) \right)$$

The Representation and Algorithm

$$u = \left(\sum_{e \in \mathcal{E}_H} \sum_{1 \le j \le m} b_{e,j} v_{e,j} + \sum_{x_i \in \mathcal{N}_H} u(x_i) \psi_i \right) + \left(u^{\mathsf{b}} + \sum_{e \in \mathcal{E}_H} Q R_e u^{\mathsf{b}}_{\omega_e} \right) + O\left(\exp\left(-b m^{\frac{1}{d+1}} \right) (\|u\|_{\mathcal{H}(\Omega)} + \|f\|_{L^2(\Omega)}) \right)$$

Offline: one-time model reduction

- compute $\{v_{e,j}\}, 1 \le j \le m$ for each e, and ψ_i for each node (local SVD and harmonic extension; parallelizable)

Online: efficient for multiple f

- compute $u^{\mathsf{n}} = u^{\mathsf{b}} + \sum_{e \in \mathcal{E}_H} Q R_e u^{\mathsf{b}}_{\omega_e}$ (solve local equations involving f; parallelizable)
- Galerkin's method using basis funcs $\{v_{e,j}\}$ and ψ_i to solve

$$a(u_{H,m}, v) = (f, v)_\Omega - a(u^{\mathsf{n}}, v)$$

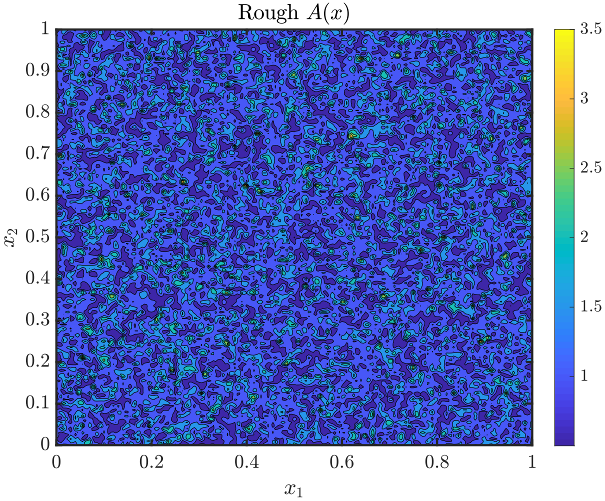
$$\|u - u_{H,m} - u^{\mathsf{n}}\|_{\mathcal{H}(\Omega)} = O\left(\exp\left(-b m^{\frac{1}{d+1}} \right) (\|u\|_{\mathcal{H}(\Omega)} + \|f\|_{L^2(\Omega)}) \right)$$

Numerical Experiments: Helmholtz's Equation

The problem set-up

- equation
$$-\nabla \cdot (A\nabla u) + Vu = f, \text{ in } \Omega = [0,1]^2$$
- boundary condition: mixed (Dirichlet + Neumann + Robin)
- $A(x) = |\xi(x)| + 0.5$ where $\xi(x)$ is piecewise linear functions with values as unit Gaussians r.v.; piecewise scale: 2^{-7}
- $-V/k^2$ draws from the same random field; $k = 2^5$
- $f(x_1, x_2) = x_1^4 - x_2^3 + 1$

Visualization of the Field



Numerical Experiments: Helmholtz's Equation

The mesh

- quadrilateral mesh
- fine mesh size $h = 2^{-10}$, coarse mesh size $H = 2^{-5}$

The accuracy of ExpMsFEM's solution compared to fine mesh solution

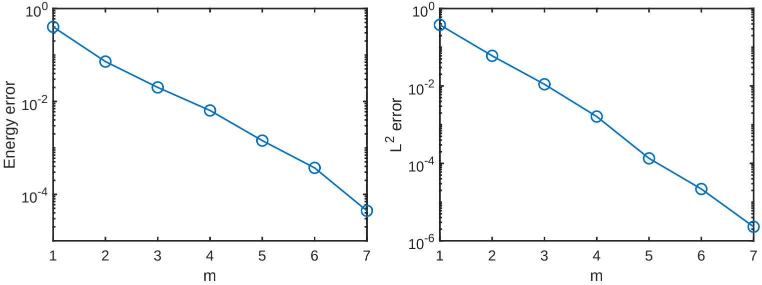


Figure: Numerical results for the mixed boundary and rough field example. Left: $e_{\mathcal{H}}$ versus m; right: e_{L^2} versus m. Number of basis functions $(2m+1)/H^2$

Summary of Part I

Exponentially convergent function representation

- multiscale (coarse-fine) decomposition is the key
- low complexity of the coarse part: restriction of harmonic-type funcs
- locality of the fine part: locally solvable

Future directions:

- advection-dominated problems
- time dependent problems
- non-intrusive model reduction and operator learning

multiscale analysis + low complexity structures

Outline

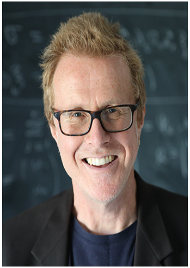
1 Exponentially Convergent Multiscale Methods for PDEs
"how to get very accurate solutions via multiscale analysis"

2 Gaussian Processes for PDEs and Inverse Problems
"how to get reliable automated solutions via Bayes inference"

Part II: Gaussian Processes for PDEs and Inverse Problems

Bamdad Hosseini
Univ. of Washington

Houman Owhadi
Caltech

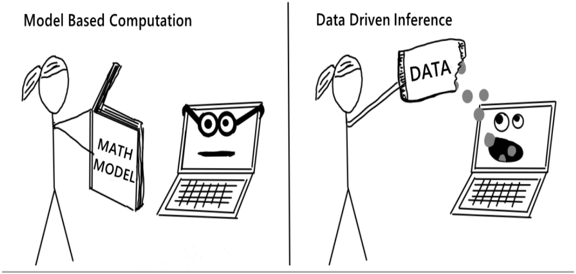


Andrew M. Stuart
Caltech

Scientific Machine Learning Automation

Expert designed numerical analysis: analyzing the equation

- finite difference/element/volume, spectral, multiscale methods ...
- well developed convergence theory, and robustness/efficiency tradeoff



Automatic machine learning paradigm: equation as data

- PINNs, deep Ritz methods, operator learning ...
- unify solving PDEs and inverse problems (IPs), algorithmically
  many empirical success; theory more complicated

Scientific Machine Learning Automation

Our Focus: Bridging the gap utilizing a Bayes framework[5]

Gaussian processes for automating solving nonlinear PDEs/IPs

[Chen, Hosseni, Owhadi, Stuart 2021]

[5] Information based complexity, Bayes probabilistic numerics, ...

The Methodology for Solving PDEs

A nonlinear elliptic PDE example

- Consider the stationary elliptic PDE

$$\begin{cases} -\Delta u(\mathbf{x}) + \tau(u(\mathbf{x})) = f(\mathbf{x}), & \forall \mathbf{x} \in \Omega, \\ u(\mathbf{x}) = g(\mathbf{x}), & \forall \mathbf{x} \in \partial\Omega. \end{cases}$$

  - Domain $\Omega \subset \mathbb{R}^d$.
  - PDE data $f, g : \Omega \to \mathbb{R}$.

- PDE has a unique strong/classical solution $u^\star$.

The Methodology[6]: Finding the MAP estimator

1. Choose a kernel $K : \overline{\Omega} \times \overline{\Omega} \to \mathbb{R}$ (Choose the prior $\mathcal{GP}(0, K)$)
   - Corresponding RKHS $\mathcal{U}$ with norm $\|\cdot\|$
2. Sample some collocation points (Choose the data/likelihood)
   - $X^{\text{int}} = \{\mathbf{x}_1, ..., \mathbf{x}_{M_\Omega}\} \subset \Omega$
   - $X^{\text{bd}} = \{\mathbf{x}_{M_\Omega+1}, ..., \mathbf{x}_{M_\Omega+M_{\partial\Omega}}\} \subset \partial\Omega$
3. Solve the optimization problem (Find the "MAP")

$$\begin{cases} \underset{u\in\mathcal{U}}{\text{minimize}} \ \|u\| \\ \text{s.t.} \quad -\Delta u(\mathbf{x}_m) + \tau(u(\mathbf{x}_m)) = f(\mathbf{x}_m), \quad \text{for } \mathbf{x}_m \subset X^{\text{int}} \\ \qquad\qquad\qquad\qquad\qquad u(\mathbf{x}_n) = g(\mathbf{x}_n), \quad \text{for } \mathbf{x}_n \subset X^{\text{bd}} \end{cases}$$

Convergence of solution as number of points approaches infinity

[Chen, Hosseni, Owhadi, Stuart 2021]

[6] Generalize many mesh-free methods and Bayes probabilistic numerics

The Methodology[6]: Finding the MAP estimator

1. Choose a kernel $K : \overline{\Omega} \times \overline{\Omega} \to \mathbb{R}$ (Choose the prior $\mathcal{GP}(0, K)$)
   - Corresponding RKHS $\mathcal{U}$ with norm $\|\cdot\|$
2. Sample some collocation points (Choose the data/likelihood)
   - $X^{\text{int}} = \{\mathbf{x}_1, ..., \mathbf{x}_{M_\Omega}\} \subset \Omega$
   - $X^{\text{bd}} = \{\mathbf{x}_{M_\Omega+1}, ..., \mathbf{x}_{M_\Omega+M_{\partial\Omega}}\} \subset \partial\Omega$
3. Solve the optimization problem (Find the "MAP")

$$\begin{cases} \underset{u \in \mathcal{U}}{\text{minimize}} \ \|u\| \\ \text{s.t.} \quad -\Delta u(\mathbf{x}_m) + \tau(u(\mathbf{x}_m)) = f(\mathbf{x}_m), \quad \text{for } \mathbf{x}_m \subset X^{\text{int}} \\ \qquad\qquad\qquad\qquad u(\mathbf{x}_n) = g(\mathbf{x}_n), \quad \text{for } \mathbf{x}_n \subset X^{\text{bd}} \end{cases}$$

Convergence of solution as number of points approaches infinity

[Chen, Hosseni, Owhadi, Stuart 2021]

[6] Generalize many mesh-free methods and Bayes probabilistic numerics

The Methodology[6]: Finding the MAP estimator

1. Choose a kernel $K : \overline{\Omega} \times \overline{\Omega} \to \mathbb{R}$ (Choose the prior $\mathcal{GP}(0, K)$)
   - Corresponding RKHS $\mathcal{U}$ with norm $\|\cdot\|$
2. Sample some collocation points (Choose the data/likelihood)
   - $X^{\mathsf{int}} = \{\mathbf{x}_1, ..., \mathbf{x}_{M_\Omega}\} \subset \Omega$
   - $X^{\mathsf{bd}} = \{\mathbf{x}_{M_\Omega+1}, ..., \mathbf{x}_{M_\Omega+M_{\partial\Omega}}\} \subset \partial\Omega$
3. Solve the optimization problem (Find the "MAP")

$$
\begin{cases}
\underset{u \in \mathcal{U}}{\text{minimize}} \ \|u\| \\
\text{s.t.} \quad -\Delta u(\mathbf{x}_m) + \tau(u(\mathbf{x}_m)) = f(\mathbf{x}_m), \quad \text{for } \mathbf{x}_m \subset X^{\mathsf{int}} \\
\qquad\qquad\qquad\qquad\qquad u(\mathbf{x}_n) = g(\mathbf{x}_n), \quad \text{for } \mathbf{x}_n \subset X^{\mathsf{bd}}
\end{cases}
$$

Convergence of solution as number of points approaches infinity

[Chen, Hosseni, Owhadi, Stuart 2021]

[6] Generalize many mesh-free methods and Bayes probabilistic numerics

The Methodology[6]: Finding the MAP estimator

1. Choose a kernel $K : \overline{\Omega} \times \overline{\Omega} \to \mathbb{R}$ (Choose the prior $\mathcal{GP}(0, K)$)
   - Corresponding RKHS $\mathcal{U}$ with norm $\|\cdot\|$
2. Sample some collocation points (Choose the data/likelihood)
   - $X^{\mathsf{int}} = \{\mathbf{x}_1, ..., \mathbf{x}_{M_\Omega}\} \subset \Omega$
   - $X^{\mathsf{bd}} = \{\mathbf{x}_{M_\Omega+1}, ..., \mathbf{x}_{M_\Omega+M_{\partial\Omega}}\} \subset \partial\Omega$
3. Solve the optimization problem (Find the "MAP")

$$\begin{cases} \underset{u\in\mathcal{U}}{\text{minimize}} \ \|u\| \\ \text{s.t.} \quad -\Delta u(\mathbf{x}_m) + \tau(u(\mathbf{x}_m)) = f(\mathbf{x}_m), \quad \text{for } \mathbf{x}_m \subset X^{\mathsf{int}} \\ \qquad\qquad\qquad\qquad\qquad\quad u(\mathbf{x}_n) = g(\mathbf{x}_n), \quad \text{for } \mathbf{x}_n \subset X^{\mathsf{bd}} \end{cases}$$

Convergence of solution as number of points approaches infinity

[Chen, Hosseni, Owhadi, Stuart 2021]

[6] Generalize many mesh-free methods and Bayes probabilistic numerics

How to Solve: Separating Nonlinearity

$$
\begin{cases}
\underset{u\in\mathcal{U}}{\text{minimize}} \ \|u\| \\
\text{s.t.} \quad -\Delta u(\mathbf{x}_m) + \tau(u(\mathbf{x}_m)) = f(\mathbf{x}_m), \quad \text{for } \mathbf{x}_m \subset X^{\mathsf{int}} \\
\qquad\qquad u(\mathbf{x}_n) = g(\mathbf{x}_n), \quad \text{for } \mathbf{x}_n \subset X^{\mathsf{bd}}
\end{cases}
$$

$$\Updownarrow \ (N = M^{\mathsf{bd}} + 2M^{\mathsf{int}})$$

$$
\begin{cases}
\underset{\mathbf{z}=(\mathbf{z}^{\mathsf{bd}},\mathbf{z}^{\mathsf{int}},\mathbf{z}^{\mathsf{int}}_{\Delta})\in\mathbb{R}^N}{\text{minimize}}
\begin{cases}
\underset{u\in\mathcal{U}}{\text{minimize}} \ \|u\| \\
\text{s.t.} \quad u(X^{\mathsf{bd}}) = \mathbf{z}^{\mathsf{bd}} \\
\qquad u(X^{\mathsf{int}}) = \mathbf{z}^{\mathsf{int}} \\
\qquad \Delta u(X^{\mathsf{int}}) = \mathbf{z}^{\mathsf{int}}_{\Delta}
\end{cases} \\
\text{s.t.} \quad -\mathbf{z}^{\mathsf{int}}_{\Delta} + \tau(\mathbf{z}^{\mathsf{int}}) = f(X^{\mathsf{int}}) \\
\qquad \mathbf{z}^{\mathsf{bd}} = g(X^{\mathsf{bd}})
\end{cases}
$$

How to Solve: Separating Nonlinearity

$$\begin{cases} \underset{u\in\mathcal{U}}{\text{minimize}} \; \|u\| \\ \text{s.t.} \quad -\Delta u(\mathbf{x}_m) + \tau(u(\mathbf{x}_m)) = f(\mathbf{x}_m), \quad \text{for } \mathbf{x}_m \subset X^{\mathsf{int}} \\ \qquad\qquad\qquad\qquad u(\mathbf{x}_n) = g(\mathbf{x}_n), \qquad \text{for } \mathbf{x}_n \subset X^{\mathsf{bd}} \end{cases}$$

$$\Updownarrow \, (N = M^{\mathsf{bd}} + 2M^{\mathsf{int}})$$

$$\begin{cases} \underset{\mathbf{z}=(\mathbf{z}^{\mathsf{bd}},\mathbf{z}^{\mathsf{int}},\mathbf{z}^{\mathsf{int}}_{\Delta})\in\mathbb{R}^N}{\text{minimize}} \begin{cases} \underset{u\in\mathcal{U}}{\text{minimize}} \; \|u\| \\ \text{s.t.} \quad u(X^{\mathsf{bd}}) = \mathbf{z}^{\mathsf{bd}} \\ \qquad u(X^{\mathsf{int}}) = \mathbf{z}^{\mathsf{int}} \\ \qquad \Delta u(X^{\mathsf{int}}) = \mathbf{z}^{\mathsf{int}}_{\Delta} \end{cases} \\ \text{s.t.} \quad -\mathbf{z}^{\mathsf{int}}_{\Delta} + \tau(\mathbf{z}^{\mathsf{int}}) = f(X^{\mathsf{int}}) \\ \qquad \mathbf{z}^{\mathsf{bd}} = g(X^{\mathsf{bd}}) \end{cases}$$

How to Solve: Inner optimization

- The inner problem has linear constraints

$$\begin{aligned}&\underset{u\in\mathcal{U}}{\text{minimize}}\ \ \|u\|\\ &\text{s.t.}\quad u(X^{\mathsf{bd}})=\mathbf{z}^{\mathsf{bd}}, u(X^{\mathsf{int}})=\mathbf{z}^{\mathsf{int}}, \Delta u(X^{\mathsf{int}})=\mathbf{z}_{\Delta}^{\mathsf{int}}\end{aligned}$$

Explicit formula for minimizer $u(\mathbf{x})=K(\mathbf{x},\phi)K(\phi,\phi)^{-1}\mathbf{z}$

- Measurement vector $\phi := (\delta_{X^{\mathsf{bd}}}, \delta_{X^{\mathsf{int}}}, \delta_{X^{\mathsf{int}}}\circ\Delta)\in(\mathcal{U}^*)^{\otimes N}$
- Kernel vector and matrix

$$K(\mathbf{x},\phi)=\left(K(\mathbf{x},X^{\mathsf{bd}}),K(\mathbf{x},X^{\mathsf{int}}),\Delta_{\mathbf{y}}K(\mathbf{x},X^{\mathsf{int}})\right)\in\mathbb{R}^N$$

$$K(\phi,\phi)=$$

$$\begin{pmatrix} K(X^{\mathsf{bd}},X^{\mathsf{bd}}) & K(X^{\mathsf{bd}},X^{\mathsf{int}}) & \Delta_{\mathbf{y}}K(X^{\mathsf{bd}},X^{\mathsf{int}})\\ K(X^{\mathsf{int}},X^{\mathsf{bd}}) & K(X^{\mathsf{int}},X^{\mathsf{int}}) & \Delta_{\mathbf{y}}K(X^{\mathsf{int}},X^{\mathsf{int}})\\ \Delta_{\mathbf{x}}K(X^{\mathsf{int}},X^{\mathsf{bd}}) & \Delta_{\mathbf{x}}K(X^{\mathsf{int}},X^{\mathsf{int}}) & \Delta_{\mathbf{x}}\Delta_{\mathbf{y}}K(X^{\mathsf{int}},X^{\mathsf{int}})\end{pmatrix}\in\mathbb{R}^{N\times N}$$

How to Solve: Inner optimization

- The inner problem has linear constraints

$$\begin{aligned} &\underset{u\in\mathcal{U}}{\text{minimize}} \ \ \|u\| \\ &\text{s.t.} \quad u(X^{\mathsf{bd}}) = \mathbf{z}^{\mathsf{bd}}, u(X^{\mathsf{int}}) = \mathbf{z}^{\mathsf{int}}, \Delta u(X^{\mathsf{int}}) = \mathbf{z}^{\mathsf{int}}_{\Delta} \end{aligned}$$

Explicit formula for minimizer $u(\mathbf{x}) = K(\mathbf{x}, \boldsymbol{\phi})K(\boldsymbol{\phi}, \boldsymbol{\phi})^{-1}\mathbf{z}$

- Measurement vector $\boldsymbol{\phi} := (\delta_{X^{\mathsf{bd}}}, \delta_{X^{\mathsf{int}}}, \delta_{X^{\mathsf{int}}} \circ \Delta) \in (\mathcal{U}^*)^{\otimes N}$
- Kernel vector and matrix

$$K(\mathbf{x}, \boldsymbol{\phi}) = \big(K(\mathbf{x}, X^{\mathsf{bd}}), K(\mathbf{x}, X^{\mathsf{int}}), \Delta_{\mathbf{y}} K(\mathbf{x}, X^{\mathsf{int}})\big) \in \mathbb{R}^N$$

$$K(\boldsymbol{\phi}, \boldsymbol{\phi}) = \begin{pmatrix} K(X^{\mathsf{bd}}, X^{\mathsf{bd}}) & K(X^{\mathsf{bd}}, X^{\mathsf{int}}) & \Delta_{\mathbf{y}} K(X^{\mathsf{bd}}, X^{\mathsf{int}}) \\ K(X^{\mathsf{int}}, X^{\mathsf{bd}}) & K(X^{\mathsf{int}}, X^{\mathsf{int}}) & \Delta_{\mathbf{y}} K(X^{\mathsf{int}}, X^{\mathsf{int}}) \\ \Delta_{\mathbf{x}} K(X^{\mathsf{int}}, X^{\mathsf{bd}}) & \Delta_{\mathbf{x}} K(X^{\mathsf{int}}, X^{\mathsf{int}}) & \Delta_{\mathbf{x}} \Delta_{\mathbf{y}} K(X^{\mathsf{int}}, X^{\mathsf{int}}) \end{pmatrix} \in \mathbb{R}^{N\times N}$$

How to Solve: Representation of the Minimizer

Representer theorem [Chen, Hosseni, Owhadi, Stuart 2021]

Every minimizer $u^\dagger$ can be represented as

$$u^\dagger(\mathbf{x}) = K(\mathbf{x}, \boldsymbol{\phi})K(\boldsymbol{\phi}, \boldsymbol{\phi})^{-1}\mathbf{z}^\dagger,$$

where the vector $\mathbf{z}^\dagger \in \mathbb{R}^N$ is a minimizer of

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^N} & \mathbf{z}^T K(\boldsymbol{\phi}, \boldsymbol{\phi})^{-1}\mathbf{z} \\ \text{s.t.} & F(\mathbf{z}) = \mathbf{y} \end{cases}$$

- Function $F : \mathbb{R}^N \to \mathbb{R}^M$ depends on PDE collocation constraints
- $\mathbf{y}$ contains PDE boundary and RHS data

Towards A Practical Algorithm

Quadratic optimization with nonlinear constraints

- A simple linearization algorithm $\mathbf{z}^k \to \mathbf{z}^{k+1}$

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^N} & \mathbf{z}^T K(\boldsymbol{\phi},\boldsymbol{\phi})^{-1}\mathbf{z} \\ \text{s.t.} & F(\mathbf{z}^k) + F'(\mathbf{z}^k)(\mathbf{z}-\mathbf{z}^k) = \mathbf{y}. \end{cases}$$

"Newton's iteration for the nonlinear PDE, faster than SGD"

- Poor conditioning of $K(\phi,\phi)$, and scale imbalance between blocks
Solution: adding scale-aware Tikhonov regularization

$$K(\phi,\phi) \leftarrow K(\phi,\phi) + \lambda \mathrm{diag}(K(\phi,\phi))$$

Towards A Practical Algorithm

Quadratic optimization with nonlinear constraints

- A simple linearization algorithm $\mathbf{z}^k \rightarrow \mathbf{z}^{k+1}$

$$\begin{cases} \min_{\mathbf{z}\in\mathbb{R}^N} & \mathbf{z}^T K(\boldsymbol{\phi},\boldsymbol{\phi})^{-1}\mathbf{z} \\ \text{s.t.} & F(\mathbf{z}^k) + F'(\mathbf{z}^k)(\mathbf{z}-\mathbf{z}^k) = \mathbf{y}. \end{cases}$$

"Newton's iteration for the nonlinear PDE, faster than SGD"

- Poor conditioning of $K(\boldsymbol{\phi},\boldsymbol{\phi})$, and scale imbalance between blocks
  Solution: adding scale-aware Tikhonov regularization

$$K(\boldsymbol{\phi},\boldsymbol{\phi}) \leftarrow K(\boldsymbol{\phi},\boldsymbol{\phi}) + \lambda \mathrm{diag}(K(\boldsymbol{\phi},\boldsymbol{\phi}))$$

Numerical Experiments

- Nonlinear Elliptic Equation, $\tau(u) = u^3$

$$\begin{cases} -\Delta u(\mathbf{x}) + \tau(u(\mathbf{x})) = f(\mathbf{x}), & \forall \mathbf{x} \in \Omega, \\ u(\mathbf{x}) = g(\mathbf{x}), & \forall \mathbf{x} \in \partial\Omega. \end{cases}$$

- Truth: $d = 2$, $u^\star(\mathbf{x}) = \sin(\pi x_1)\sin(\pi x_2) + 4\sin(4\pi x_1)\sin(4\pi x_2)$
- Kernel: $K(\mathbf{x}, \mathbf{y}; \sigma) = \exp\left(-\frac{|\mathbf{x}-\mathbf{y}|^2}{2\sigma^2}\right)$

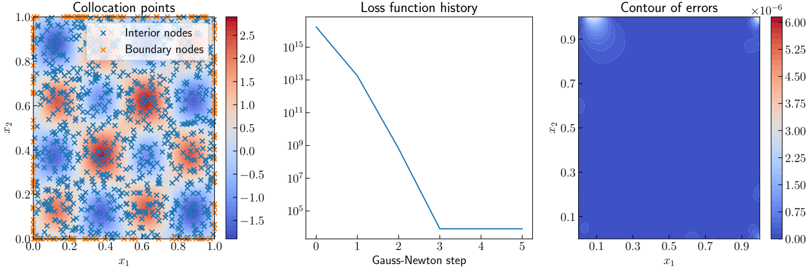


Figure: $N_{\text{domain}} = 900, N_{\text{boundary}} = 124$

Convergence Study

- For $\tau(u) = 0, u^3$, use Gaussian kernel with lengthscale σ
- L^2, L^∞ accuracy, compared with Finite Difference (FD)

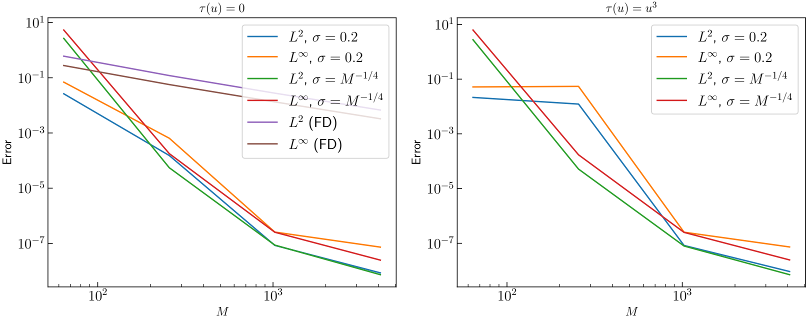
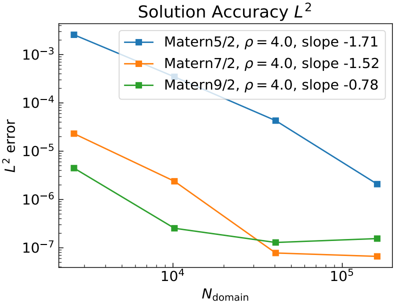
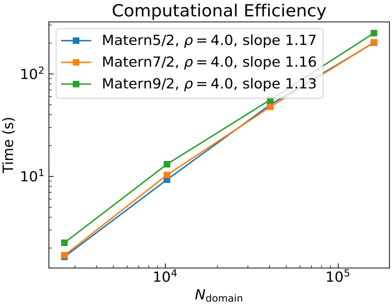


Figure: Convergence of the kernel method is fast, since the solution is smooth

Scalability: Sparse Cholesky Factorization

- Sparse Cholesky of $K(\phi,\phi)^{-1}$ under coarse to fine ordering of ϕ
  screening effects [Stein 2002], [Schäfer, Sullivan, Owhadi 2021]
- Complexity: $O(N\rho^d)$ memory and $O(N\rho^{2d})$ time; ρ is a parameter
  theory: $\rho = \log(N/\epsilon) \Rightarrow \epsilon$-approximation of $K(\phi,\phi)^{-1}$ even when ϕ contains derivatives (best complexity so far) [Chen, Schäfer, Owhadi, 2023]





Matérn kernel with different ν. Run 3 Newton's iterations. Accuracy floor due to finite ρ and regularization

Numerical Experiments: Inverse Problems

Darcy Flow inverse problems

$$\begin{cases} \min_{u,a} \|u\|_K^2 + \|a\|_\Gamma^2 + \frac{1}{\gamma^2}\sum_{j=1}^{I}|u(\mathbf{x}_j) - o_j|^2, \\ \text{s.t.} \quad -\mathsf{div}(\exp(a)\nabla u)(\mathbf{x}_m) = 1, \quad \forall \mathbf{x}_m \in (0,1)^2 \\ \qquad\qquad u(\mathbf{x}_m) = 0, \quad \forall \mathbf{x}_m \in \partial(0,1)^2. \end{cases}$$

- Recover a from pointwise measurements of u
- Model (u, a) as independent GPs
- Impose PDE constraints and formulate Bayesian inverse problem

Numerical Experiments: Darcy Flow

- Kernel $K(\mathbf{x},\mathbf{x}';\sigma)=\exp\left(-\frac{|\mathbf{x}-\mathbf{x}'|^2}{2\sigma^2}\right)$ for both u and a

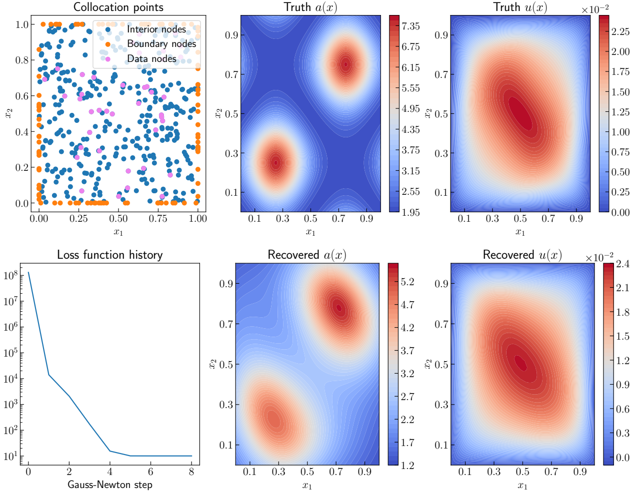


Figure: $N_{\text{domain}}=400, N_{\text{boundary}}=100, N_{\text{observation}}=50$

Further Directions

GPs Model Misspecification: hierarchical learning to select k_θ

- analysis of large data consistency and implicit bias in learning θ
  [Chen, Owhadi, Stuart 2021]

GPs Fast Solvers: multiscale algo for kernel matrices using probability

- randomly pivoted Cholesky: provably effcient low rank approximation
  [Chen, Epperly, Tropp, Webber 2022]
- sparse Cholesky: state-of-the-art complexity $O(N \log^{2d}(N/\epsilon))$ in time
  [w/ Florian Schäfer, Houman Owhadi]

Uncertainty Quantification: beyond point estimators; sampling

- affine invariant gradient flows, Gaussian mixtures, climate applications ...
  [Chen, Huang, Huang, Reich, Stuart, 2023], ...

Fine-grained multiscale analysis + probabilistic inference

Thanks!

https://yifanc96.github.io