Deep adaptive sampling for surrogate modeling

PKU-Changsha Institute for Computing and Digital Economy

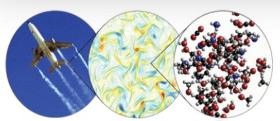
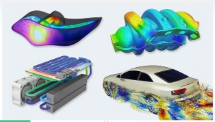
Kejun Tang
joint work with Xili Wang (PKU), Jiayu Zhai (ShanghaiTech), Xiaoliang Wan (LSU) and Chao Yang (PKU)

July 18, 2024
SciCADE 2024

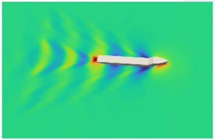
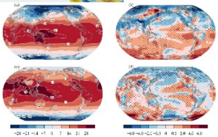
Outline

Background

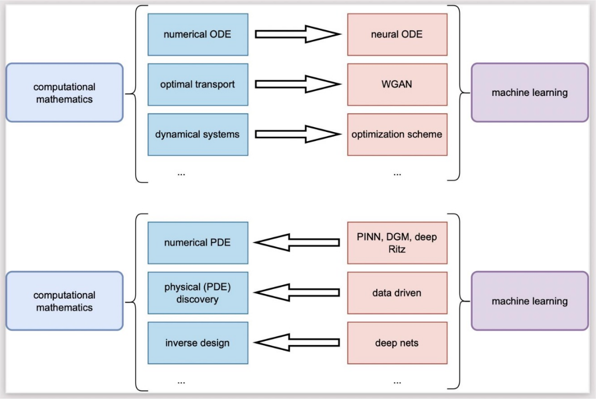
- Uncertainty quantification
- Inverse design
- Digital twins
- Shape optimization
- Operator learning
- ...

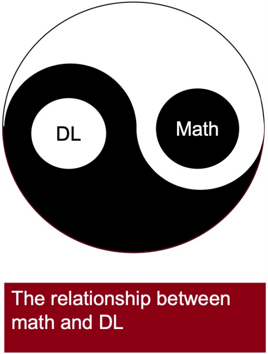




......

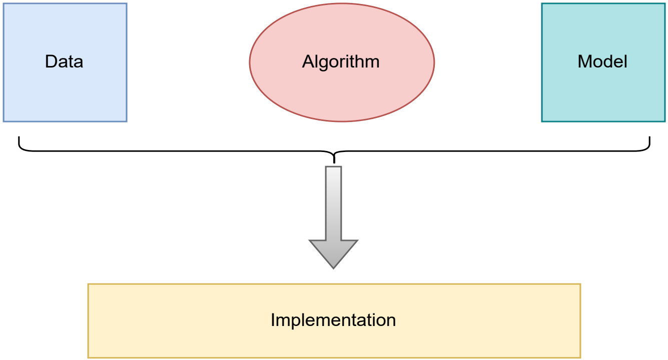




Background


computational mathematics
numerical ODE
optimal transport
dynamical systems
...
neural ODE
WGAN
optimization scheme
...
machine learning
computational mathematics
numerical PDE
physical (PDE) discovery
inverse design
...
PINN, DGM, deep Ritz
data driven
deep nets
...
machine learning

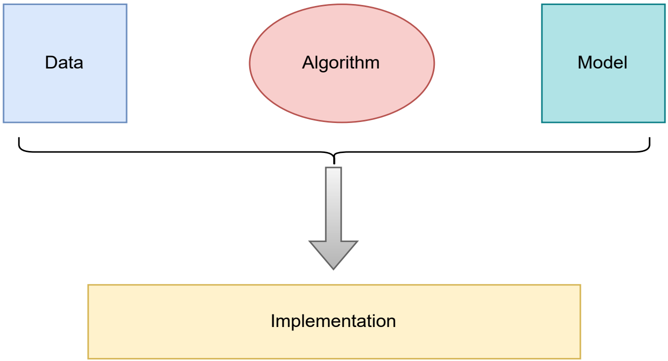

DL
Math
The relationship between math and DL

Big data era: data-driven



- Model: deep neural networks, physical model, or coupling
- Data: labeled, unlabled, random samples ....
- Algorithm: various optimization methods

Big data era: data-driven



data is oil

- model is driven by data
- data has the influence on generalization

Goal

Traditional numerical methods

- high fidelity
- suffers from the curse of dimensionality

Machine (deep) learning approaches

- low fidelity
- weaker dependence on dimensionality

our purpose:
Develop adaptive sampling methods for neural network-based surrogates

Parametric PDEs

Parametric differential equations

$$\begin{aligned} \mathcal{L}\left(\mathbf{x}, \xi; u\left(\mathbf{x}, \xi\right)\right) &= s(\mathbf{x}, \xi) \qquad \forall\left(\mathbf{x}, \xi\right) \in \Omega_s \times \Omega_p, \\ \mathcal{B}\left(\mathbf{x}, \xi; u\left(\mathbf{x}, \xi\right)\right) &= g(\mathbf{x}, \xi) \qquad \forall\left(\mathbf{x}, \xi\right) \in \partial\Omega_s \times \Omega_p. \end{aligned}$$

For any ξ, compute the solution efficiently without solving the differential equation repeatedly.

- $\mathcal{L}$: differential operator; $\mathcal{B}$: boundary operator
- $\Omega_s \subset \mathbb{R}^n$: spatial domain with smooth boundary $\partial\Omega_s$
- $\mathbf{x} \in \Omega_s$: spatial variable
- $\Omega_p \subset \mathbb{R}^d$: parametric space
- $\xi \in \Omega_p$: parameters
- we denote $\Omega = \Omega_s \times \Omega_p$ and $\partial\Omega = \partial\Omega_s \times \Omega_p$ for simplicity

Physics-informed surrogate modeling

Why

- fast inference
- tackle high dimensional problems

How: a deep net $u(\mathbf{x}, \xi; \Theta) \rightarrow u(\mathbf{x}, \xi)$

$$J(u(\mathbf{x}, \xi; \Theta)) = \|r(\mathbf{x}, \xi; \Theta)\|_{2,\Omega}^2 + \gamma \|b(\mathbf{x}, \xi; \Theta)\|_{2,\partial\Omega}^2,$$

$$\|r(\mathbf{x}, \xi; \Theta)\|_{2,\Omega}^2 = \int_\Omega r^2(\mathbf{x}, \xi; \Theta) d\mathbf{x} d\xi,$$

$$\|b(\mathbf{x}, \xi; \Theta)\|_{2,\partial\Omega}^2 = \int_{\partial\Omega} b^2(\mathbf{x}, \xi; \Theta) d\mathbf{x} d\xi$$

An optimization problem: $\min_{\Theta} J(u(\mathbf{x}, \xi; \Theta))$

Illustration of the error

Discretization of the loss

$$J_N\left(u(\mathbf{x},\xi;\Theta)\right) = \frac{1}{N_r}\sum_{i=1}^{N_r} r^2(\mathbf{x}_\Omega^{(i)},\xi^{(i)};\Theta) + \gamma\frac{1}{N_b}\sum_{i=1}^{N_b} b^2(\mathbf{x}_{\partial\Omega}^{(i)},\xi^{(i)};\Theta),$$

$\mathbf{x}_\Omega^{(i)}$ drawn from Ω_s, $\mathbf{x}_{\partial\Omega}^{(i)}$ drawn from $\partial\Omega_s$, and $\xi^{(i)}$ drawn from Ω_p.

$$u(\mathbf{x},\xi;\Theta^*) = \arg\min_\Theta J(u(\mathbf{x},\xi;\Theta)),$$

$$u(\mathbf{x},\xi;\Theta_N^*) = \arg\min_\Theta J_N(u(\mathbf{x},\xi;\Theta)).$$

$$\mathbb{E}\left(\left\|u_{\Theta_N^*} - u\right\|_\Omega\right) \le \underbrace{\mathbb{E}\left(\left\|u_{\Theta_N^*} - u_{\Theta^*}\right\|_\Omega\right)}_{\text{statistical error}} + \underbrace{\left\|u_{\Theta^*} - u\right\|_\Omega}_{\text{approximation error}}$$

Illustration of the error

Where do the errors come from?

the capability of neural networks $\rightarrow$ approximation error
the strategy of loss discretization $\rightarrow$ statistical error

In this work, we focus on how to reduce the statistical error

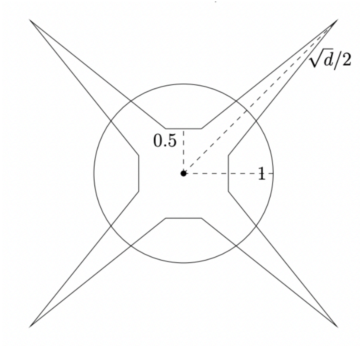
Difficulities

low regularities or high-dimensional

Key point: the strategy to discretize the loss. Uniform random sampling? Quasi-random sampling?

Geometric properties of high-dimensional spaces

uniformly distributed points in high-dimensional spaces



Most of the volume of a high-dimensional cube is located around its corner [Vershynin, High-Dimensional Probability, 2020]. Cube: $[-1, 1]^d$

$$\mathbb{P}(\|\mathbf{x}\|_2^2 \leq 1) \leq \exp(-\frac{d}{10}).$$

Sampling strategy

PDF for sampler

$$p(\mathbf{x},\xi) = p(\mathbf{x}|\xi)p(\xi) \quad \text{or} \quad p(\mathbf{x},\xi) = p(\xi|\mathbf{x})p(\mathbf{x})$$

In practice, the above two PDF models can be further simplified.

- Sample from a joint PDF

$$p(\mathbf{x},\xi) = \hat{r}(\mathbf{x},\xi) \propto r^2(\mathbf{x},\xi;\theta)h(\mathbf{x},\xi),$$

$$p_{\mathbf{x},\xi}(\mathbf{x},\xi;\theta_f) = p_{\mathbf{z}|\xi}(f_{\text{KRnet}}(\mathbf{x},\xi;\theta_f)) \left|\det \nabla_{\mathbf{x}} f_{\text{KRnet}}\right|.$$

- Sample from a marginal PDF

$$p(\xi) = \tilde{r}^2(\xi;\theta) = \int_{\Omega_s} r^2(\mathbf{x},\xi;\theta)d\mathbf{x},$$

$$p_{\xi}(\xi;\theta_f) = p_{\mathbf{z}}(f_{\text{KRnet}}(\xi;\theta_{\mathbf{f}})) \left|\det \nabla_{\xi} f_{\text{KRnet}}\right|.$$

Deep adaptive sampling for surrogates (DAS^2)

A viewpoint of variance reduction (both $\mathbf{x}$ and ξ)

$$J_r(u(\mathbf{x},\xi;\Theta)) = \int_\Omega \frac{r^2(\mathbf{x},\xi;\Theta)}{p(\mathbf{x},\xi)} p(\mathbf{x},\xi) d\mathbf{x} d\xi \approx \frac{1}{N_r}\sum_{i=1}^{N_r} \frac{r^2(\mathbf{x}_\Omega^{(i)},\xi^{(i)};\Theta)}{p(\mathbf{x}_\Omega^{(i)},\xi^{(i)})},$$

where $\{\mathbf{x}_\Omega^{(i)},\xi^{(i)}\}_{i=1}^{N_r}$ from $p(\mathbf{x},\xi)$ instead of a uniform distribution.

A viewpoint of variance reduction (only ξ)

$$J_r(u(\mathbf{x},\xi;\Theta)) = \int_{\Omega_p} \frac{\tilde{r}^2(\xi;\Theta)}{p(\xi)} p(\xi) d\xi \approx \frac{1}{N_{\tilde{r}}}\sum_{i=1}^{N_{\tilde{r}}} \frac{\tilde{r}^2(\xi^{(i)};\Theta)}{p(\xi^{(i)})},$$

where $\tilde{r}^2(\xi;\Theta) \approx \frac{1}{m_\mathbf{x}}\sum_{i=1}^{m_\mathbf{x}} r^2(\mathbf{x}^{(i)},\xi;\Theta)$, $\{\mathbf{x}^{(i)}\}_{i=1}^{m_\mathbf{x}}$ in the spatial domain, $\{\xi^{(i)}\}_{i=1}^{N_{\tilde{r}}}$ from $p(\xi)$.

Deep adaptive sampling for surrogates (DAS^2)

Importance sampling

$$p^* = r^2(\mathbf{x},\xi;\Theta)/\mu,\ \mu = \int_\Omega r^2(\mathbf{x},\xi;\Theta)d\mathbf{x}d\xi.$$

Sample from $p(\mathbf{x},\xi)$ for a fixed Θ: a deep generative model

$$p_{KRnet}(\mathbf{x},\xi;\Theta_f) \approx \mu^{-1} r^2(\mathbf{x},\xi;\Theta)$$

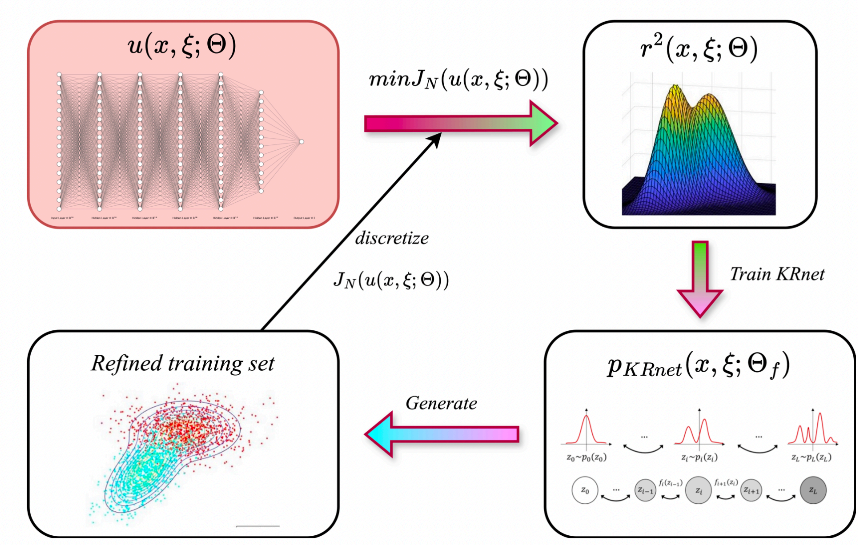
where $p_{KRnet}(\mathbf{x},\xi;\Theta_f)$ is a PDF induced by KRnet [Tang, Wan and Liao, 2020]; [Tang, Wan and Liao, 2021]

"Error estimator": $\hat{r}(\mathbf{x},\xi) \propto r^2(\mathbf{x},\xi;\Theta)$

$$D_{KL}(\hat{r}(\mathbf{x},\xi)\|p_{KRnet}(\mathbf{x},\xi;\Theta_f)) = \int_B \hat{r}\log\hat{r}d\mathbf{x}d\xi - \int_B \hat{r}\log p_{KRnet}d\mathbf{x}d\xi.$$

$$\min_{\Theta_f} H(\hat{r},p_{KRnet}) = -\int_B \hat{r}\log p_{KRnet}d\mathbf{x}d\xi.$$

Algorithm of DAS[2]



Analysis

Assumptions [T. De Ryck and S. Mishra, 2022]

- $\theta \in \Theta = [-a, a]^D$: trainable parameters of u_θ where $a > 0$ is a constant.
- $\mathscr{M}_1 : \theta \mapsto J_{r,N}$ and $\mathscr{M}_2 : \theta \mapsto J_r$: Lipschitz continuous in the ℓ_∞ sense with Lipschitz constant $\mathfrak{L}$ for $\theta \in \Theta$.
- Let $c > 0$ be a constant that is independent of Θ. Assume that $J_{r,N} \in [0, c]$ for all $\theta \in \Theta$.

Theorem (Wang, Tang, Zhai, Wan, and Yang, 2024)

Let θ_N^* be a minimizer of $J_{r,N}$ where the collocation points are independently drawn from a given probability distribution. Given $\varepsilon \in (0, 1)$, the following inequality holds under the above assumptions

$$J_r(u_{\theta_N^*}) \leq \varepsilon^2 + J_{r,N}(u_{\theta_N^*})$$

with probability at least $1 - (4a\mathfrak{L}/\varepsilon^2)^D \exp(-N_r\varepsilon^4/2c^2)$.

Numerical results: physics-informed operator learning

The following dynamical system

$$\begin{cases} \dfrac{\mathrm{d}u(x,\xi)}{\mathrm{d}x} = e^{-D\|\boldsymbol{\xi}-\mathbf{0.5}\|^2} f(x,\xi), \quad x \in [0,1], \\ u(0,\xi) = 0, \end{cases}$$

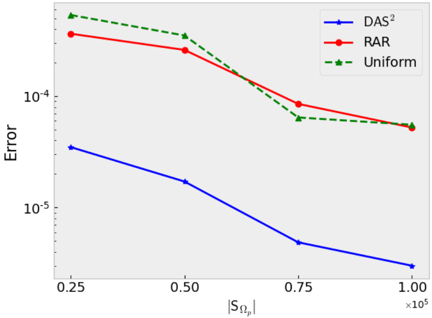
- $D = 6$: a fixed parameter
- $\xi \in \Omega_p = [-1,1]^8$

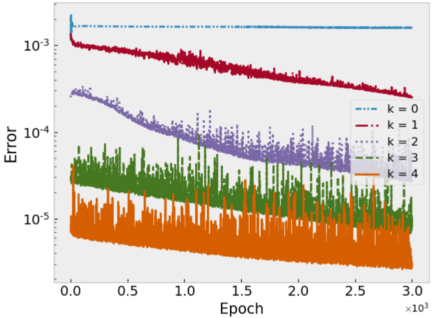
Goal: learn the sulution operator from f to the solution u without any paired input-output data

f is drawn from V_{poly} where

$$V_{\mathrm{poly}} = \left\{ \sum_{i=0}^{d-1} \xi_i T_i(x) : |\xi_i| \le M \right\}.$$

Numerical results: physics-informed operator learning





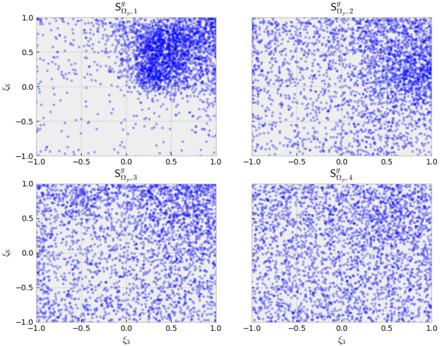
$$u_\theta(x, \xi) \approx \sum_{i=1}^{I} q_{\theta_1}^{(i)}(x) t_{\theta_2}^{(i)}(\xi) + b_0,$$

marginal PDF for sampling

Numerical results: physics-informed operator learning

The results and evolution of samples

sampling strategy \ $\lvert S_{\Omega_p} \rvert$	2.5×10^4	5×10^4	7.5×10^4	1×10^5
Uniform (0.006s)	5.4×10^{-4}	3.5×10^{-4}	6.4×10^{-5}	5.5×10^{-5}
RAR (0.006s)	3.6×10^{-4}	2.6×10^{-4}	8.5×10^{-5}	5.2×10^{-5}
DAS^2 (0.03s)	3.5×10^{-5}	1.7×10^{-5}	4.9×10^{-6}	3.0×10^{-6}



Numerical results: parametric optimal control problems

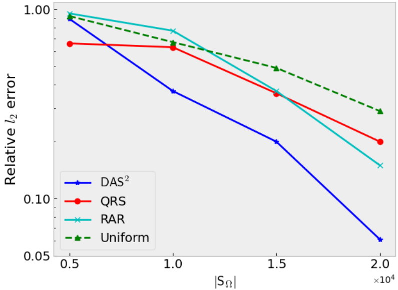
$$
\begin{cases}
\min\limits_{y(\mathbf{x},\xi),u(\mathbf{x},\xi)} J\left(y\left(\mathbf{x},\xi\right),u\left(\mathbf{x},\xi\right)\right)=\dfrac{1}{2}\left\|y(\mathbf{x},\xi)-y_d(\mathbf{x},\xi)\right\|_{2,\Omega}^2+\dfrac{\alpha}{2}\left\|u(\mathbf{x},\xi)\right\|_{2,\Omega}^2, \\
\text{subject to} \begin{cases} -\Delta y(\mathbf{x},\xi)=u(\mathbf{x},\xi) & \text{in } \Omega, \\ y(\mathbf{x},\xi)=1 & \text{on } \partial\Omega, \end{cases} \\
\text{and} \quad u_a \leq u(\mathbf{x},\xi) \leq u_b \quad \text{a.e. in } \Omega,
\end{cases}
$$

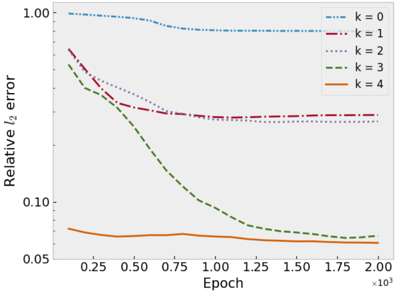
where $\Omega_p=(\xi_1,\xi_2)$ is the paramter.
$\Omega(\boldsymbol{\xi})=([0,2]\times[0,1])\backslash B((1.5,0.5),\xi_1)$ and the desired state is given by

$$
y_d(\boldsymbol{\xi})=\begin{cases} 1 & \text{in } \Omega_1=[0,1]\times[0,1], \\ \xi_2 & \text{in } \Omega_2(\boldsymbol{\xi})=([1,2]\times[0,1])\backslash B((1.5,0.5),\xi_1), \end{cases}
$$

where $B((1.5,0.5),\xi_1)$ is a ball of radius ξ_1 with center $(1.5,0.5)$, $\alpha=0.001$ and $\boldsymbol{\xi}\in\Omega_p=[0.05,0.45]\times[0.5,2.5]$.

Numerical results: parametric optimal control problems





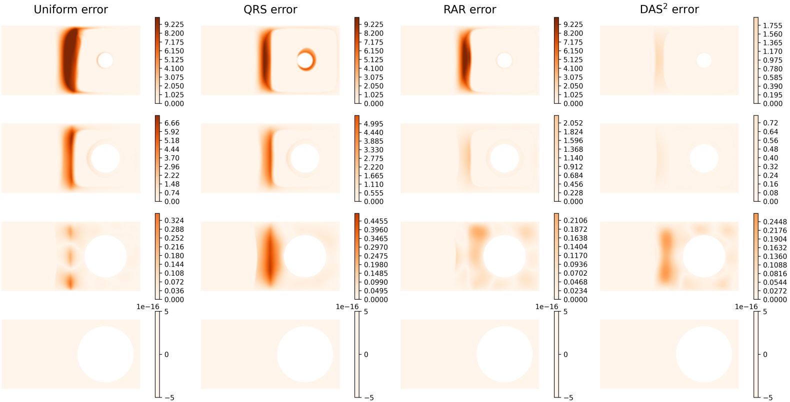
$$l(\mathbf{x},\xi) = x_1(2-x_1)x_2(1-x_2)(\xi_1^2-(x_1-1.5)^2-(x_2-0.5)^2).$$

$$u(\mathbf{x},\xi) \approx u_{\theta_u}(\mathbf{x},\xi),\ y(\mathbf{x},\xi) \approx l(\mathbf{x},\xi)y_{\theta_y}(\mathbf{x},\xi)+1,\ p(\mathbf{x},\xi) \approx l(\mathbf{x},\xi)p_{\theta_p}(\mathbf{x},\xi)$$

$$\Omega := \{(\mathbf{x},\xi)|0 \le x_1 \le 2, 0 \le x_2 \le 1, 0.05 \le \xi_1 \le 0.45, 0.5 \le \xi_2 \le 2.5,$$
$$(x_1-1.5)^2+(x_2-0.5)^2 \ge \xi_1^2\}.$$

joint PDF model for sampling

Numerical results: parametric optimal control problems



top to bottom: $\xi = (0.10, 2.5)$ $\xi = (0.20, 2.0)$ $\xi = (0.30, 1.5)$ $\xi = (0.40, 0.5)$.

Numerical results: parametric optimal control problems

sampling strategy \ $\lvert S_\Omega \rvert$	0.5×10^4	1×10^4	1.5×10^4	2×10^4
Uniform (0.1s)	0.92	0.67	0.49	0.29
QRS (0.1s)	0.66	0.63	0.36	0.20
RAR (0.1s)	0.95	0.77	0.37	0.15
DAS^2 (0.1s)	0.89	0.37	0.20	0.06

- 11×11 grid in the parametric space
- dolfin-adjoint solver for a fixed parameter
- dolfin-adjoint solver : 18804 seconds

Parametric lid-driven cavity flow problems

$$\begin{cases} \mathbf{u}(\mathbf{x},\xi)\cdot\nabla\mathbf{u}(\mathbf{x},\xi)+\nabla p(\mathbf{x},\xi)=\dfrac{1}{Re(\xi)}\Delta\mathbf{u}(\mathbf{x},\xi) & \text{in } \Omega, \\ \nabla\cdot\mathbf{u}(\mathbf{x},\xi)=0 & \text{in } \Omega, \\ \mathbf{u}(\mathbf{x},\xi)=\mathbf{g}(\mathbf{x},\xi) & \text{on } \partial\Omega, \end{cases}$$

- $\mathbf{u}(\mathbf{x},\xi)=[u(\mathbf{x},\xi),v(\mathbf{x},\xi)]^{\mathsf{T}},\mathbf{x}=[x,y]^{\mathsf{T}}$
- $Re(\xi)=\xi\in\Omega_p=[100,1000]$
- The physical domain is $\Omega_s=[0,1]\times[0,1]$
- Boundary conditions

$$\mathbf{g}(\mathbf{x},\xi)=\begin{cases}[1,0]^{\mathsf{T}}, y=1; \\ [0,0]^{\mathsf{T}},\ \text{otherwise.}\end{cases}$$

Goal: obtaining all-at-once solutions for $Re\in[100,1000]$

Parametric lid-driven cavity flow problems

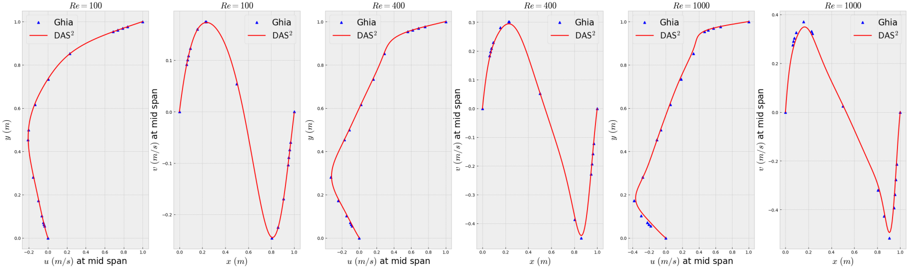
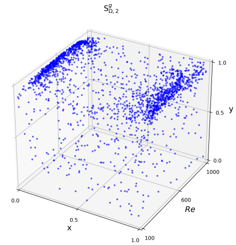
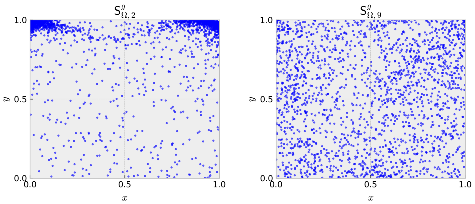


Figure: The velocity components at the location of mid-span lines for surrogate modeling of parametric lid-driven cavity flow problems ($Re \in [100, 1000]$). The results for $Re = 100, 400, 1000$ are chosen for visualization.

Parametric lid-driven cavity flow problems



(a) 3d points



(b) 2d projection onto *xy*-plane

Parametric lid-driven cavity flow problems

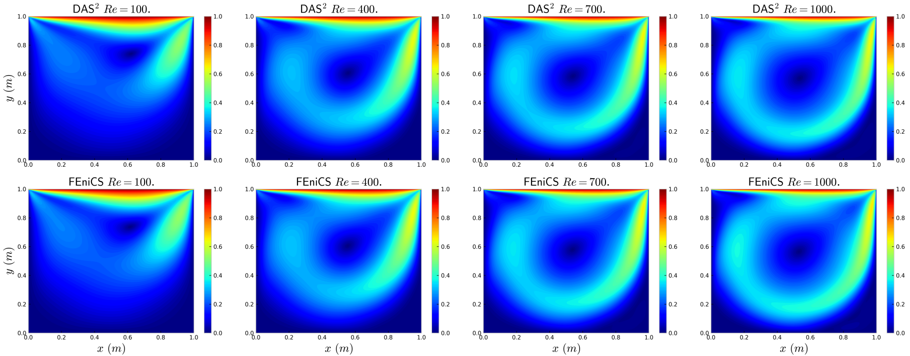


Figure: The visualization of $|\mathbf{u}| = \sqrt{u^2 + v^2}$ for surrogate modeling of parametric lid-driven cavity flow problems, $Re \in [100, 1000]$. The l_2 relative errors are $1.5\%, 1.1\%, 3.1\%, 4.8\%$ for $Re = 100, 400, 700, 1000$ respectively.

- Inference time of DAS^2: 0.02 seconds,
- The computation time of FEniCS: 309.94 seconds

Summary and outlook

summary

- illustrate that DAS^2 is necessary for parametric PDEs
- significantly improve the accuracy for low-regularity problems especially for high-dimensional or parametric problems

outlook

- incoporate tensor networks into deep adaptive sampling
- large scale problems
- more applications

Thank you for your attention
Q & A