Toward Efficient Kernel-Based Solvers for Nonlinear PDEs

Zhitong Xu[1]*, Da Long[1]*, Yiming Xu[2], Guang Yang[1], Shandian Zhe[1], Houman Owhadi[3]

[1] University of Utah [2] University of Kentucky [3] California Institute of Technology

KAHLERT SCHOOL OF COMPUTING
THE UNIVERSITY OF UTAH

ICML
International Conference On Machine Learning

Traditional Methods:

Complex Mesh design

Intricate numerical techniques

ML Based Methods:

No need for sophisticated numerical tricks

Simple and convenient implementation

ML Based PDE Solver: Setting

Given:

$$\mathcal{P}(u) = f(\mathbf{x})(\mathbf{x} \in \Omega), \quad \mathcal{B}(u) = g(\mathbf{x})(\mathbf{x} \in \partial\Omega)$$

Sample collocation points:

$$\mathcal{M} = \{\mathbf{x_1}, \ldots, \mathbf{x_{M_\Omega}} \in \boldsymbol{\Omega}, \mathbf{x_{M_\Omega+1}}, \ldots, \mathbf{x_M} \in \boldsymbol{\partial\Omega}\}$$

Goal: **Solve u(x)**

Why GP Based Solver:

- ***Solid mathematical foundation***
- ***High expressiveness***
- ***Uncertainty quantification***

Previous GP PDE Solver (Chen et al. 2021)

A Simple case: Non-linear operator

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

Their Approach: GP prior

$$\begin{cases} \underset{\mathbf{z}^{(1)} \in \mathbb{R}^M, \mathbf{z}^{(2)} \in \mathbb{R}^{M_\Omega}}{\text{minimize}} \begin{cases} \underset{u \in \mathcal{U}}{\text{minimize}} \ \|u\| \\ \text{s.t.} \quad u(\mathbf{x}_m) = z_m^{(1)} \text{ and } -\Delta u(\mathbf{x}_m) = z_m^{(2)}, \text{ for } m = 1, \ldots, M, \end{cases} \\ \text{s.t.} \quad z_m^{(2)} + \tau(z_m^{(1)}) = f(\mathbf{x}_m), \quad \text{for } m = 1, \ldots, M_\Omega, \\ \qquad z_m^{(1)} = g(\mathbf{x}_m), \quad \text{for } m = M_\Omega + 1, \ldots, M. \end{cases}$$

Previous Work: Computational Bottleneck

$$K(\phi,\phi) = \begin{bmatrix} K(x,x') & \Delta_x K(x,x') \\ * & \Delta_x \Delta_{x'} K(x,x') \end{bmatrix}$$

Their Final Objective:

$$\underset{\mathbf{z}_\Omega^{(1)} \in \mathbb{R}^{M_\Omega}}{\text{minimize}} \ (\mathbf{z}_\Omega^{(1)}, g(\mathbf{x}_{\partial\Omega}), f(\mathbf{x}_\Omega) - \tau(\mathbf{z}_\Omega^{(1)})) K(\boldsymbol{\phi}, \boldsymbol{\phi})^{-1} \begin{pmatrix} \mathbf{z}_\Omega^{(1)} \\ g(\mathbf{x}_{\partial\Omega}) \\ f(\mathbf{x}_\Omega) - \tau(\mathbf{z}_\Omega^{(1)}) \end{pmatrix}$$

Interior Points ($\mathbf{z}_\Omega^{(1)}$); Boundary Points ($g(\mathbf{x}_{\partial\Omega})$); *computational demanding* ($K(\boldsymbol{\phi}, \boldsymbol{\phi})^{-1}$)

Some Issues(Inverse of Gram Matrix): 1. Instability 2. Poor Scalability

Our Approach: Solution Approximation

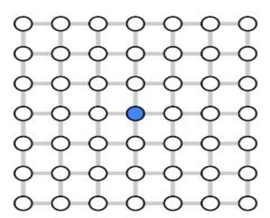
Let $u(\mathbf{x};\boldsymbol{\eta}) = \kappa(\mathbf{x}, \mathcal{M})\mathbf{K}_{MM}^{-1}\boldsymbol{\eta}$, eta is the estimated function value on collocation points,

Our objective:

$$\underset{\boldsymbol{\eta}}{\text{minimize}} \ \ \mathcal{L}(u(x;\boldsymbol{\eta}); \alpha, \beta) := \|u\|_{\mathcal{U}}^2 + \alpha \left[\frac{1}{M_\Omega}\sum_{m=1}^{M_\Omega}(\mathcal{P}(u)(\mathbf{x}_m) - f(\mathbf{x}_m))^2 - \epsilon/2\right] + \beta\left[\frac{1}{M - M_\Omega}\sum_{m=M_\Omega+1}^{M}(\mathcal{B}(u)(\mathbf{x}_m) - g(\mathbf{x}_m))^2 - \epsilon/2\right]$$

Key Ideas:

1. **Only estimate solution values on collocation points**
2. **Induce Kronecker Product structure**



Efficient Kernel Computation

With Product Kernel: $\kappa(\mathbf{x}, \mathbf{x}') = \prod_{j=1}^d \kappa_j(x_j, x'_j)$

$u(\mathbf{x};\boldsymbol{\eta}) = \kappa(\mathbf{x}, \mathcal{M})\mathbf{K}_{MM}^{-1}\boldsymbol{\eta}$ can be reduced to

$$[\kappa_1(x_1, \mathbf{s}^1) \otimes \ldots \otimes \kappa_d(x_d, \mathbf{s}^d)] [\mathbf{K}_1 \otimes \ldots \otimes \mathbf{K}_d]^{-1}\boldsymbol{\eta}$$

$$[\kappa_1(x_1, \mathbf{s}^1)\mathbf{K}_1^{-1} \otimes \ldots \otimes \kappa_d(x_d, \mathbf{s}^d)\mathbf{K}_d^{-1}]\ \boldsymbol{\eta}$$

Much Smaller ($\mathbf{K}_1^{-1}$)

Kronecker Product Property: $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$

Any derivatives maintain Kronecker Product structure:

$$\partial_{x_1 x_d} u(\mathbf{x};\boldsymbol{\eta}) = \mathcal{A} \times_1 [\partial_{x_1}\kappa_1(x_1, \mathbf{s}^1)\mathbf{K}_1^{-1}] \times_2 \ldots \times_d [\partial_{x_d}\kappa_d(x_d, \mathbf{s}^d)\mathbf{K}_d^{-1}]$$

Numerical Results:

Table 3: L^2 error of solving more challenging PDEs with a large number of collocation points.

(a) The Burgers' equation (14) with viscosity $\nu = 0.001$.

Method	43200 (360×120)	67500 (450×150)	97200 (540×180)	120000 (600×200
PINN	4.05E-03	6.01E-03	3.94E-03	4.13E-03
SKS	**3.90E-03**	**3.50E-03**	**2.60E-03**	**2.28E-03**

(b) The 2D Allen-Cahn equation (17) with $a = 15.0$

Method	6400 (80×80)	8100 (90×90)	22500 (150×150)	40000 (200×200)
PINN	5.03E0	5.30E0	4.21E0	5.86E0
SKS	**8.27E-05**	**3.41E-05**	**4.34E-06**	**4.44E-06**

(c) The 2D Allen-Cahn equation (17) with $a = 20.0$

Method	6400	8100	22500	40000
PINN	4.18E0	4.45E0	5.86E0	5.93E0
SKS	**3.98E-04**	**1.82E-04**	**4.00E-05**	**2.98E-05**

Table 1: L^2 error of solving less challenging PDEs, with a small number of collocation points. Inside the parenthesis of each top row indicates the grid used by SKS, which takes approximately the same number of collocation used by DAKS. Note that the Gram matrix of DAKS is larger than SKS.

(a) The Burgers' equation (14) with viscosity $\nu = 0.02$

Method	600 (25 × 25)	1200 (35 × 35)	2400 (49 × 49)	4800 (70 × 70)
DAKS	1.75E-02	7.90E-03	8.65E-04	**9.76E-05**
PINN	**2.68E-03**	**6.72E-04**	**3.60E-04**	3.73E-04
SKS	1.44E-02	5.40E-03	7.83E-04	3.21E-04

(b) Nonlinear elliptic PDE (15)

Method	300 (18 × 18)	600 (25 × 25)	1200 (35 × 35)	2400 (49 × 49)
DAKS	1.15E-01	1.15E-04	8.65E-04	**1.68E-07**
PINN	3.39E-01	1.93E-02	1.28E-03	3.20E-04
SKS	**1.26E-02**	**6.93E-05**	**6.80E-06**	1.83E-06

(c) Eikonal PDE (16)

Method	300	600	1200	2400
DAKS	1.01E-01	1.64E-02	2.27E-04	7.78E-05
PINN	2.95E-02	1.26E-02	4.53E-03	3.50E-03
SKS	**6.23E-04**	**2.68E-04**	**1.91E-04**	**2.51E-05**

Convergence Analysis:

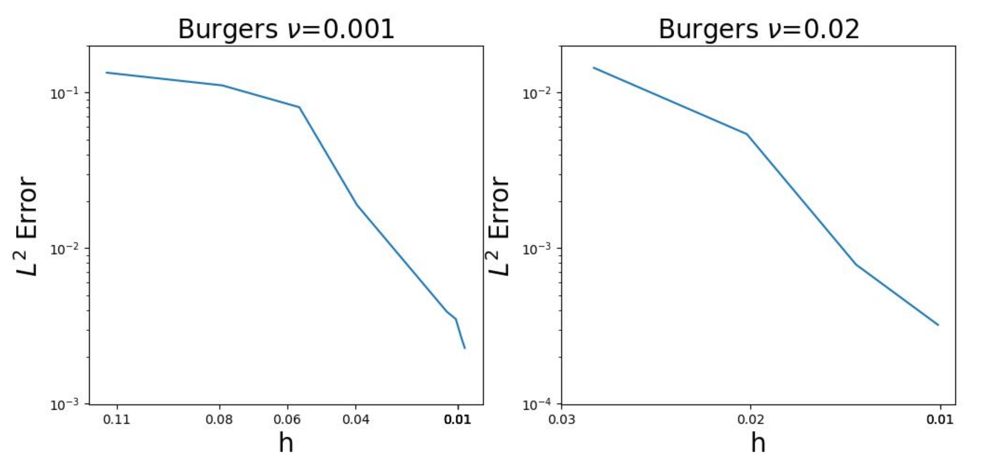
Lemma 4.2. *Let $u^* \in \mathcal{U}$ denote the unique strong solution of (1). Suppose Assumption 4.1 is satisfied, and a set of collocation points $\mathcal{M} \subset \overline{\Omega}$ is given, where $\mathcal{M}_\Omega \subset \mathcal{M}$ denotes the collocation points in the interior Ω and $\mathcal{M}_{\partial\Omega} \subset \mathcal{M}$ the collocation points on the boundary $\partial\Omega$. Assume the Voronoi diagram based on the collocation points has a uniformly bounded aspect ratio across all the cells 2. Define the fill-distances*

$$h_\Omega := \sup_{\mathbf{x} \in \Omega} \inf_{\mathbf{x}' \in \mathcal{M}_\Omega} |\mathbf{x} - \mathbf{x}'|,$$

$$h_{\partial\Omega} := \sup_{\mathbf{x} \in \partial\Omega} \inf_{\mathbf{x}' \in \mathcal{M}_{\partial\Omega}} \rho_{\partial\Omega}(\mathbf{x}, \mathbf{x}'), \quad (12)$$

where $|\cdot|$ is the Euclidean distance, and $\rho_{\partial\Omega}$ is a geodesic distance defined on $\partial\Omega$. Set $h = max(h_\Omega, h_{\partial\Omega})$. There is always a minimizer of (6) with the set of collocation points $\mathcal{M}$ and $\epsilon = C_0 h^{2\tau}$ where $C_0 > 0$ is a sufficiently large constant independent of h. Let $u^\dagger$ denote such a minimizer. When h is sufficiently small, at least $h \le C_1 M^{-\frac{1}{d}}$ where $C_1 > 0$ is a constant, then

$$\|u^\dagger - u^*\|_{H^l(\Omega)} \le C h^\rho \|u^*\|_{\mathcal{U}}, \quad (13)$$



Shows as fill-in distance decrease, minimizer of our objective will converge to the unique strong solution.