

Parallel Physics-Informed Neural Networks via Domain Decomposition

Raj Shukla

with A. Jagtap and G. Karniadakis

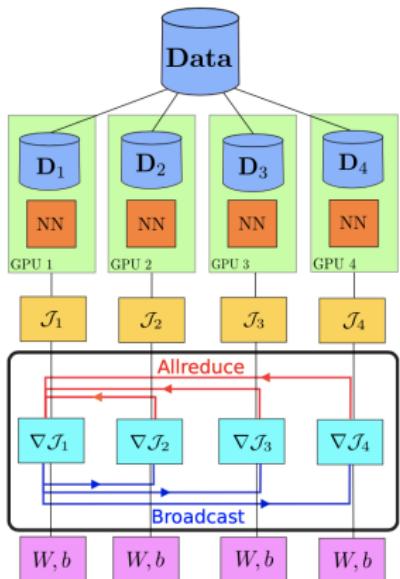
CRUNCH Group

Division of Applied Mathematics
Brown University

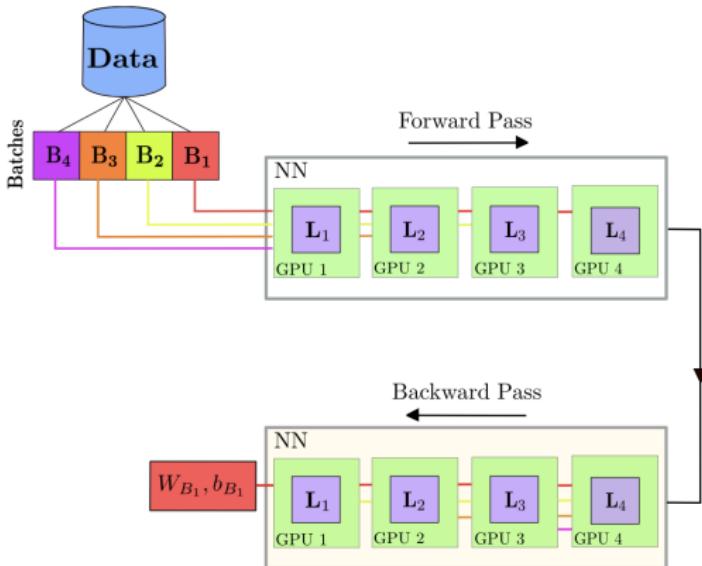
AMD Inc.

July 12, 2022

Motivation



Data parallel



Model parallel

Motivations

cPINN, XPINN: PINN + Domain Decomposition

cPINNs: A Jagtap, E Kharazmi, GE Karniadakis, CMAME 365
(2020) 113028

XPINNs: A Jagtap, GE Karniadakis,CiCP 28 (5), 2002-2041, 2020

Outline

- ① cPINNs
- ② XPINNs
- ③ Parallel Implementations

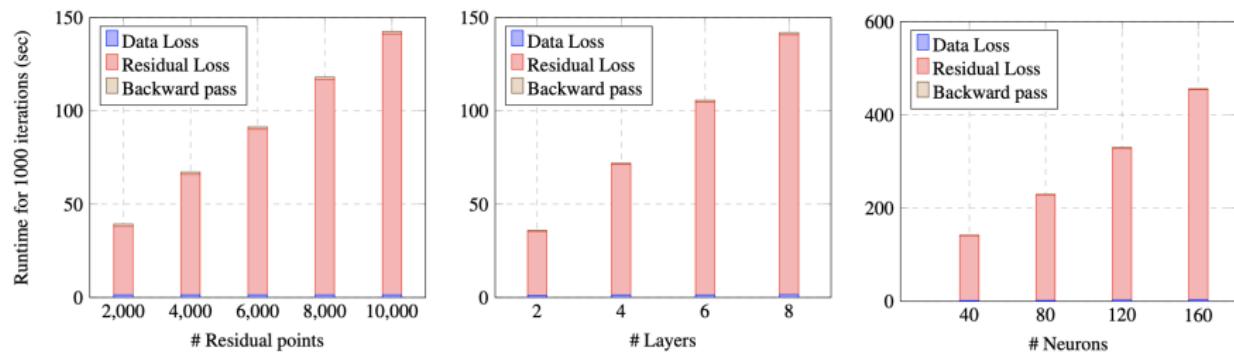
PINN Limitations:

- ① Large training time. (Domain decomposition)
- ② Due to high-dimensional non-convex optimization problem, the accuracy of the method suffers.

Physics-Informed Neural Networks: Profiling

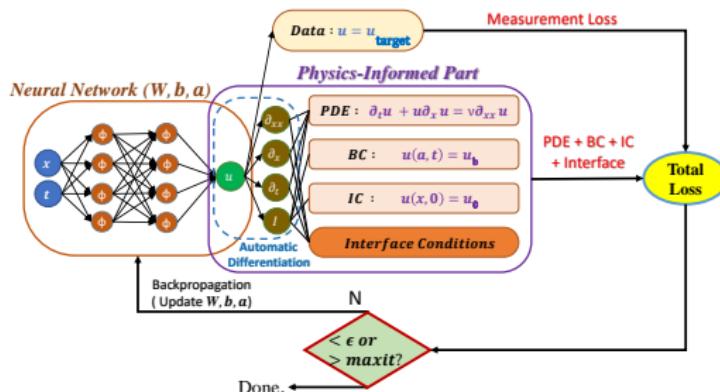
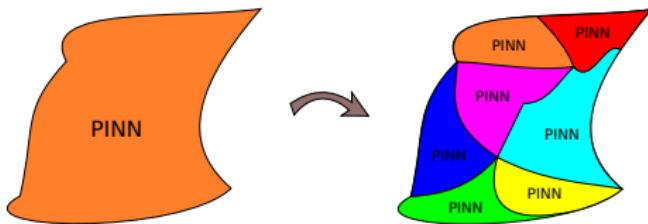
Viscous Burger's equation:

$u_t + uu_x = \nu u_{xx}$, $x \in \mathbb{R}$, $t > 0$ with IC $u(x, 0) = -\sin(\pi x)$ and BCs $u(t, 1) = u(t, -1) = 0$.



Reverse-mode AD: Graph Traversal: $\mathcal{O}(|E| + |V|)$

Domain Decomposition based PINNs



$$\mathcal{L}(\tilde{\Theta}) = \frac{1}{N_u} \sum_{i=1}^{N_u} |u_{\text{target}}^i - u_{\tilde{\Theta}}(x_i^u)|^2 + \frac{1}{N_f} \sum_{i=1}^{N_f} |\mathcal{F}_{\tilde{\Theta}}(x_i^f)|^2 + \text{Interface Loss}$$

Advantages

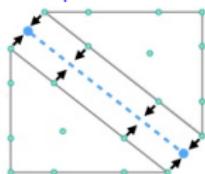
- ① **Parallelization capacity** : The partial independence of individual PINNs in decomposed domains can be further employed to implement cPINN in a parallelized algorithm.
- ② **Representation capacity** : Due to deployment of individual network in each sub-domain by the proposed cPINN method, the representation capacity of the network increases.
- ③ **Efficient hyper-parameter adjustment** : Based on prior (and sparse) knowledge of the solution regularity in each sub-domain, the hyper-parameter set of corresponding PINN is properly adjusted.
- ④ **Reduction of error propagation in the domain** : Individual networks in each sub-domain provide additional information about the solution using interface conditions, which results in reduction of error propagation in the neighbouring sub-domains as well as faster convergence.

Conservative PINNs (cPINNs) : Applications to conservation laws

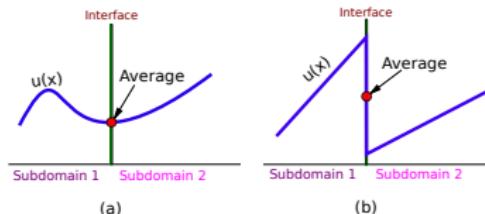
Conservation Laws:

$$u_t + \nabla \cdot (fu) = 0$$

Flux continuity = $\frac{1}{N_{Iq}} \sum_{i=1}^{N_{Iq}} \left| f_q(u(x_{Iq}^i)) \cdot \mathbf{n} - f_{q+}(u(x_{Iq}^i)) \cdot \mathbf{n} \right|^2$



Avg.Solution continuity = $\frac{1}{N_{Iq}} \sum_{i=1}^{N_{Iq}} \left| u_q(x_{Iq}^i) - \left\{ \left\{ u(x_{Iq}^i) \right\} \right\} \right|^2$



What Next?

- DD strategy for every PDE (not necessarily the conservation laws).
- Also, it will be more efficient if we can do DD in space-time domain.

eXtended PINNs (XPINNs)

XPINNs: Interface conditions Avg.Solution continuity =

$$\frac{1}{N_{I_q}} \sum_{i=1}^{N_{I_q}} \left| u_q(x_{I_q}^i) - \left\{ \left\{ u(x_{I_q}^i) \right\} \right\} \right|^2$$

$$\text{Residual continuity} = \frac{1}{N_{I_q}} \sum_{i=1}^{N_{I_q}} \left| \mathcal{R}_q(u(x_{I_q}^i)) - \mathcal{R}_{q^+}(u(x_{I_q}^i)) \right|^2$$

+

Additional continuity conditions

XPINNs: Interface conditions Avg.Solution continuity =

$$\frac{1}{N_{I_q}} \sum_{i=1}^{N_{I_q}} \left| u_q(x_{I_q}^i) - \left\{ \left\{ u(x_{I_q}^i) \right\} \right\} \right|^2$$

$$\text{Residual continuity} = \frac{1}{N_{I_q}} \sum_{i=1}^{N_{I_q}} \left| \mathcal{R}_q(u(x_{I_q}^i)) - \mathcal{R}_{q^+}(u(x_{I_q}^i)) \right|^2$$

+

Additional continuity conditions

Advantages

- ① Extension to any differential equation(s)
- ② Generalized space-time domain decomposition
- ③ Simple interface conditions

cPINN Loss

$$\mathcal{L} = \mathcal{L}_D + \mathcal{L}_F + \mathcal{L}_f + \mathcal{L}_c$$

XPINN Loss

$$\mathcal{L} = \mathcal{L}_D + \mathcal{L}_F + \mathcal{L}_{F_i} + \mathcal{L}_c$$

\mathcal{L}_D : Volume term - Concurrent evaluation

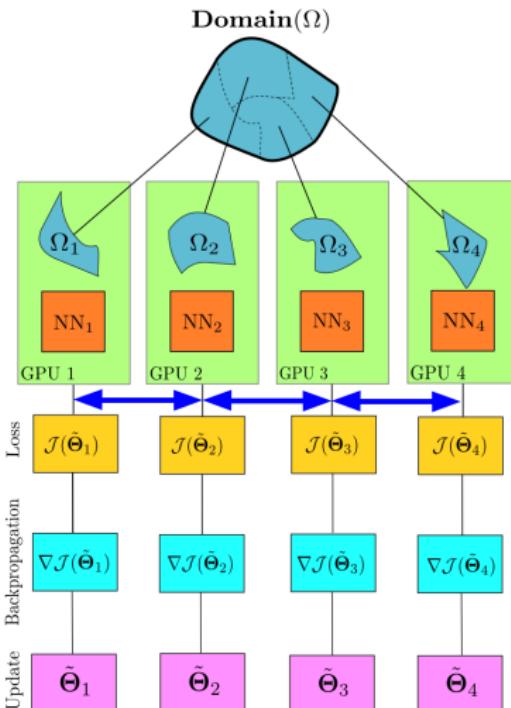
\mathcal{L}_F : Volume term - Concurrent evaluation

\mathcal{L}_f : Surface term - Communication bound

\mathcal{L}_{F_i} : Surface term - Communication bound

\mathcal{L}_c : Surface term - Communication bound

Parallel Implementation



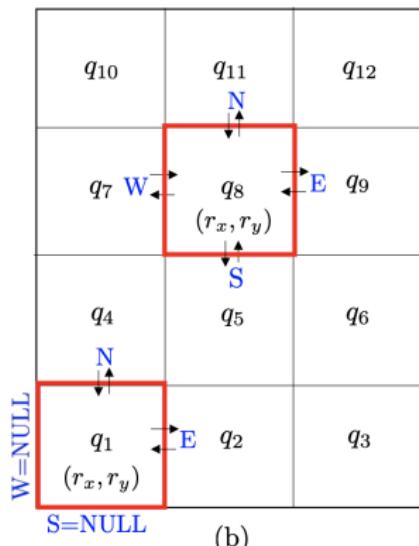
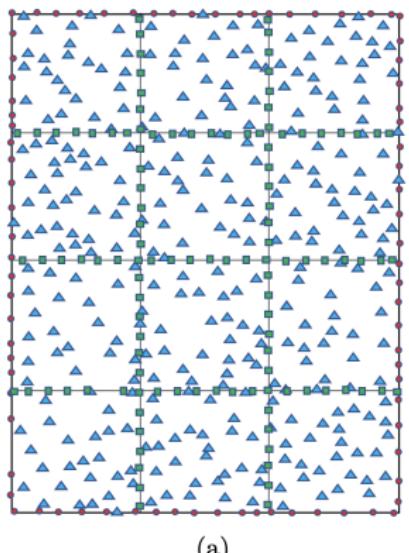
cPINN or XPINN parallel approach

Domain Partitioning: 2D Incompressible Navier-Stokes equations

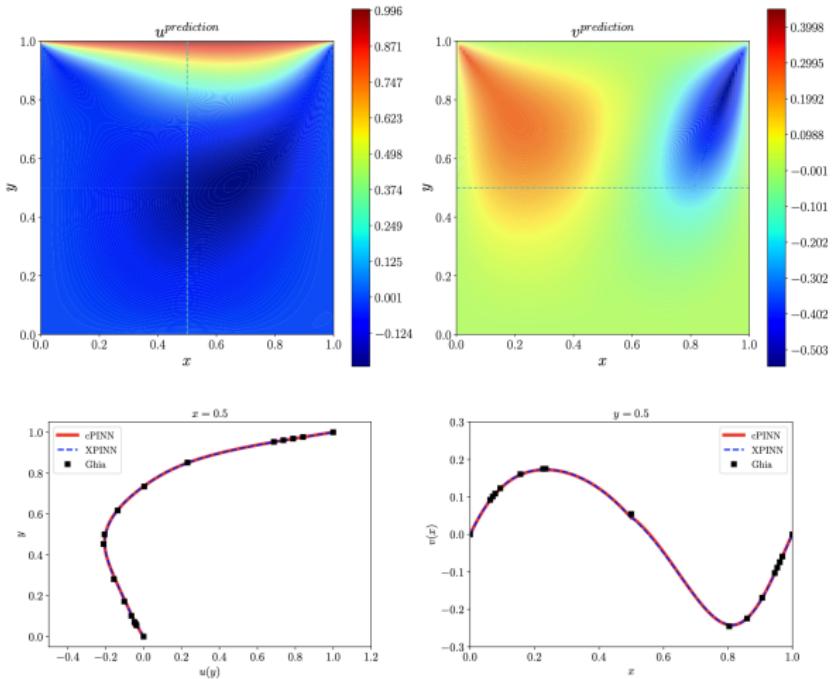
$$(u \cdot \nabla) u = -\nabla p + \frac{1}{Re} \nabla^2 u, \quad \text{in } \Omega$$

$$\nabla \cdot u = 0, \quad \text{in } \Omega$$

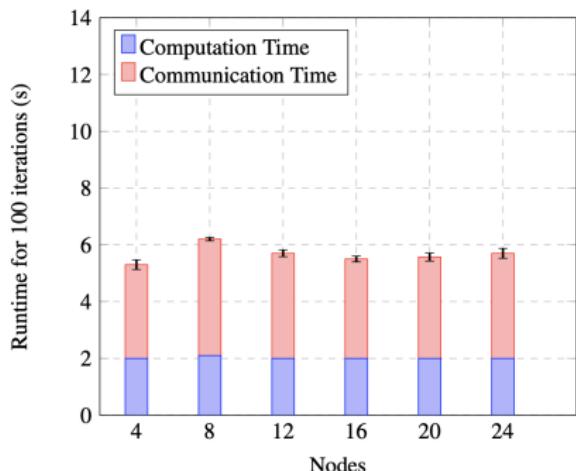
▲ Residual Points • Data Points ■ Interface Points



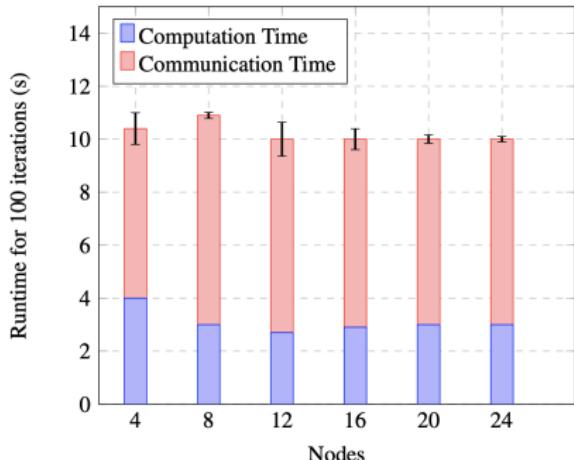
Accuracy of cPINN and XPINN



Computation vs Communication Time: CPUs



(a) Compute vs Communication for cPINN



(b) Compute vs Communication for xPINN

Figure 2: Computation and communication time for (a) cPINN and (b) xPINN with $N_f = 100$ and $N_{fi} = 20$

Computation vs Communication Time: GPUs

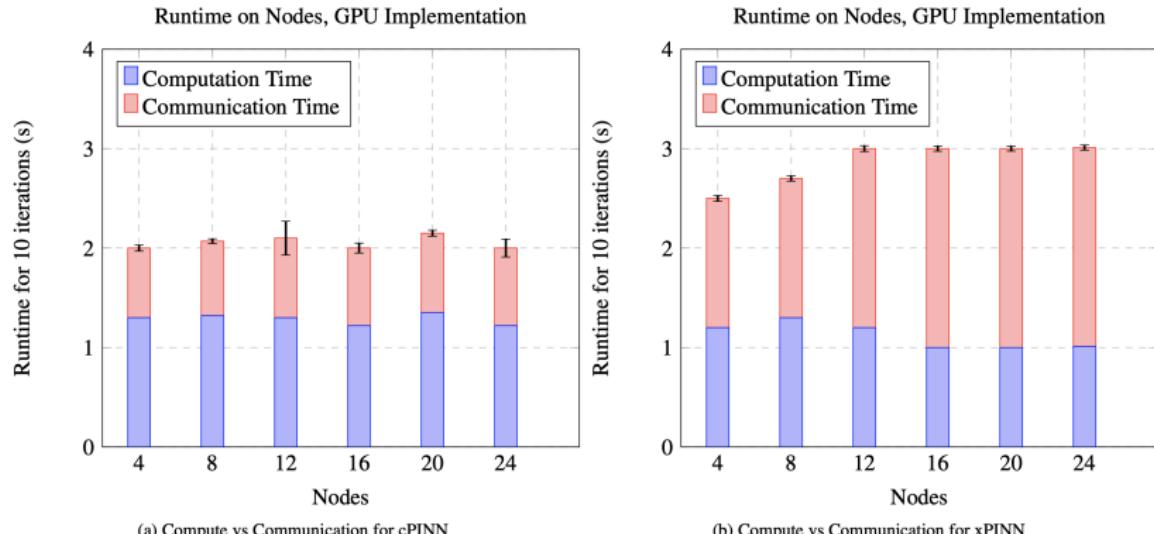


Figure 3: Computation and communication time for (a) cPINN and (b) xPINN with $N_f = 4000$ and $N_{fi} = 200$

Communication: XPINN > cPINN, Why?

NS Equation:

$$\begin{aligned}(\mathbf{u} \cdot \nabla \mathbf{u}) &= -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u}, && \text{in } \Omega \\ \nabla \cdot \mathbf{u} &= 0, && \text{in } \Omega\end{aligned}$$

cPINN Loss: : $\mathcal{F}(\mathbf{u}) + \{\mathbf{u}^+ - \mathbf{u}^-\}$

Flux	X-dir	Y-dir
Div*,†	u	v
Mom. X	$u^2 + p - \frac{1}{Re} \frac{\partial u}{\partial x}$	$uv - \frac{1}{Re} \frac{\partial u}{\partial y}$
Mom Y	$uv - \frac{1}{Re} \frac{\partial v}{\partial x}$	$v^2 + p - \frac{1}{Re} \frac{\partial v}{\partial y}$

XPINN Loss: : $\mathcal{F}(\mathbf{u}) + \{\mathbf{u}^+ - \mathbf{u}^-\}$

Weak Scaling: cPINN and XPINN

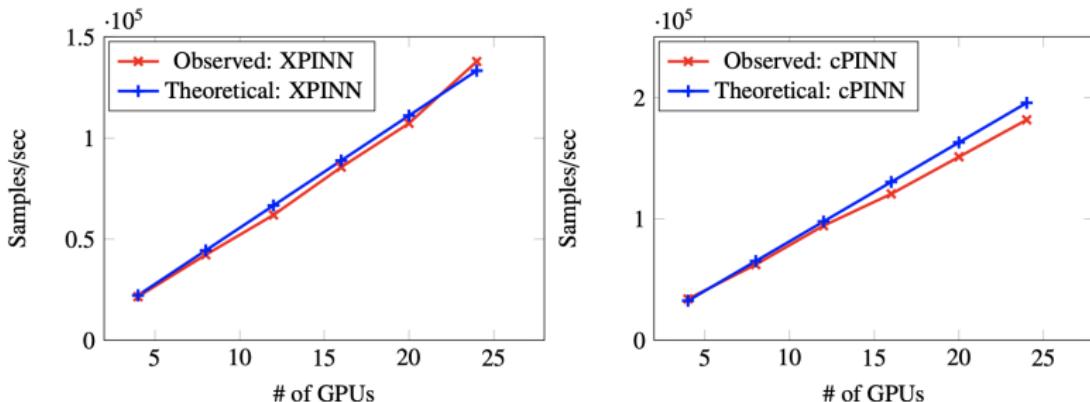
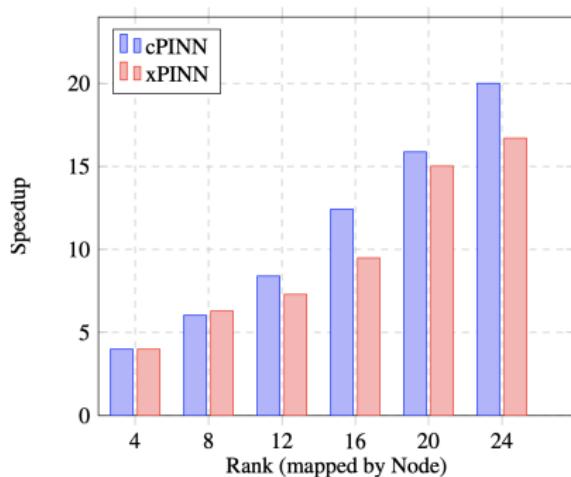
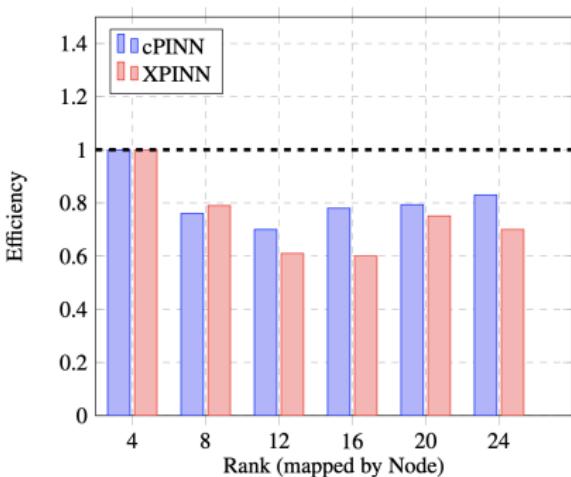


Figure 5: Weak scaling for the cPINN and the XPINN methods.

Strong Scaling: CPUs



(a) Strong scaling cPINN

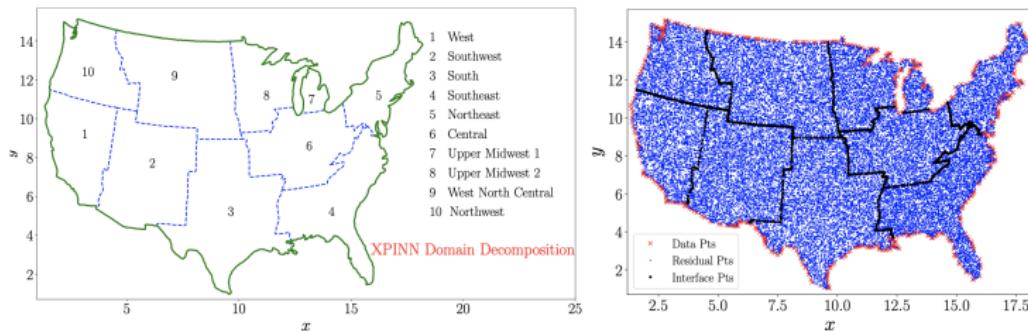


(b) Strong scaling for xPINN

Inverse problem: Steady state heat conduction with variable conductivity

$$\partial_x(K(x, y)T_x) + \partial_y(K(x, y)T_y) = f(x, y)$$

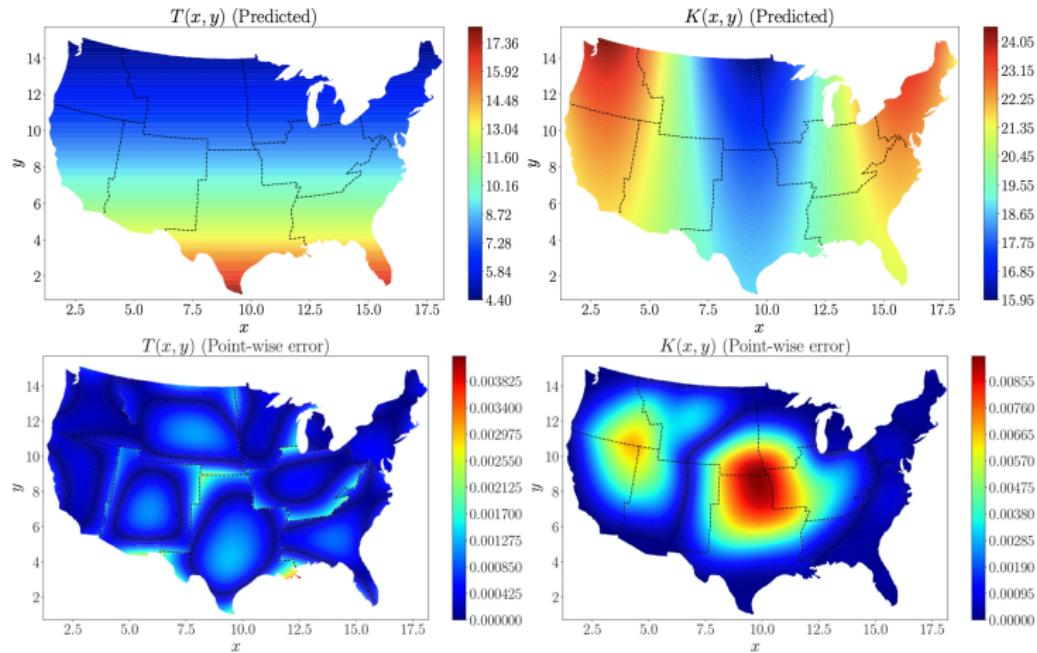
Domain partitioning



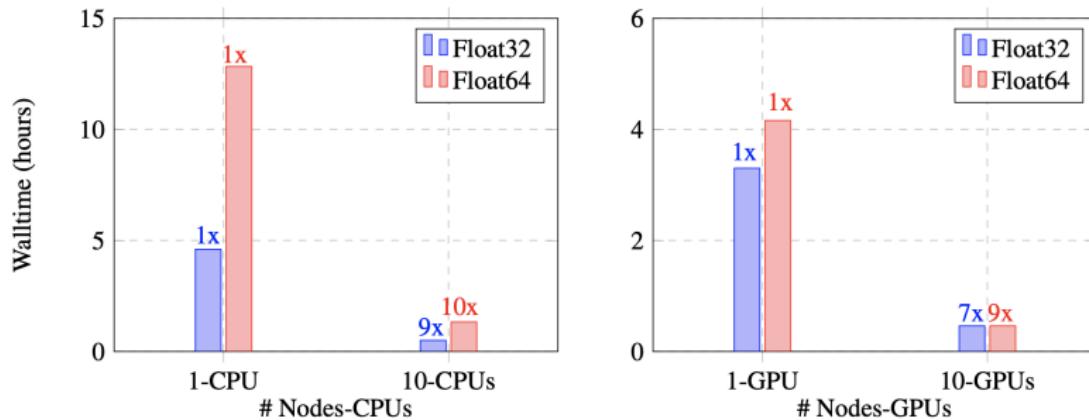
Hyperparameters

Subdomain number	1	2	3	4	5	6	7	8	9	10
# Residual points	3000	4000	5000	4000	3000	4000	800	3000	5000	4000
Adaptive Activation function	tanh	sin	cos	tanh	sin	cos	tanh	sin	cos	tanh

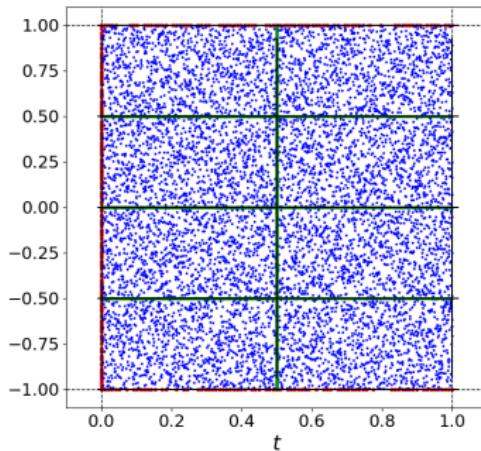
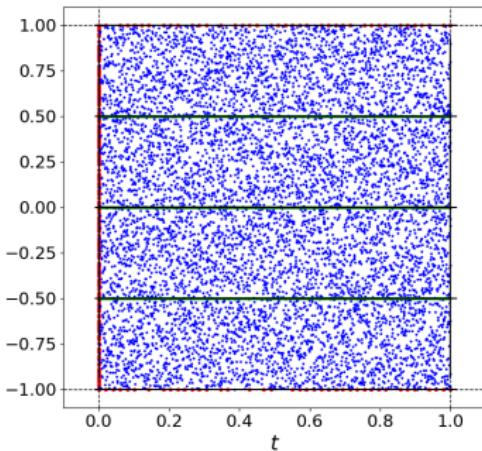
Temperature and Conductivity (T, K)



Scaling



Space-Time partitioning



cPINN vs XPINN partition

# x -partitions	# t partitions	cPINN time per iter. (s)	XPINN time per iter. (s)
4	1	0.14	-
4	2	-	0.060

Conclusions

- cPINN is only applicable for conservation laws. However, application of XPINN is independent of the nature of DEs.
- cPINN is more efficient than XPINN if decomposition is performed in space only.
- For transient problems, the communication overhead in XPINN (compared to cPINN) due to spatial decomposition will be compensated by partitioning the domain along the time axis as well.
- Weak scaling is achieved for $x \in \{CPUs, GPUs\}$.
- Strong scaling is achieved for CPUs.

Some development needs

- Message passing for tensors
- Better auto-differentiation algorithm or approach

Acknowledgment

- OSD/AFOSR MURI
- DARPA CompMods
- DOE PhILMs
- Oak Ridge National Laboratory (ORNL)
- CCV, Brown University

Thank You!