Extending the Physics Informed Neural Network Model to Multigroup Neutron Diffusion Problems

Mohamed H. Elhareef and Zeyun Wu

Department of Mechanical and Nuclear Engineering
Virginia Commonwealth University, Richmond VA, USA

VCU College of Engineering

ANS Student Conference 2022
University of Illinois at Urbana-Champaign, Urbana IL, April 15, 2022

Outline

- Introduction of PINN
- 1G Fixed-source Diffusion Model (M&C 2021)
- 2G Fixed-source Diffusion Model
  - One Dimensional (1D) Examples
  - Two Dimensional (2D) Examples
- Future Work and Conclusions

Physics-Informed Neural Networks (PINN)

- Provides a framework for integrating physics principles within data-driven models
- Solves two classes of problems:
    1. Data-driven solution of PDEs (Forward approach)
    2. Data-driven discovery of PDEs (Inverse approach)
- Successfully applied to various engineering problems:

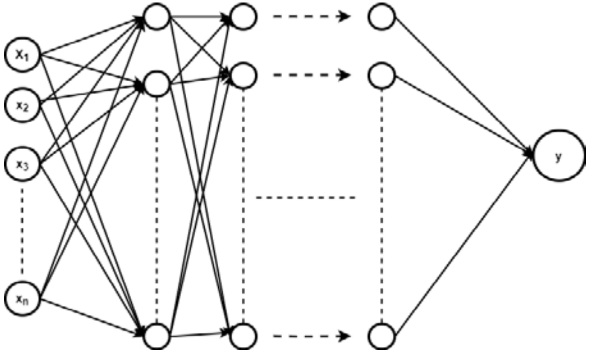
    Fluids, Quantum Mechanics, Power Systems, etc.

Ref.: M. Raissi et al., "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, **378**, pp. 686-707 (2019).

Standard Feed-forward Neural Networks

- Universal function approximators

- Automatic differentiation (AD)



Forward PINN Framework

The PINN approach uses the neural networks (NN) model to approximate the solution of PDEs:

- Considering a general non-linear differential operator:

$$F := \mathbb{N}\left(Y(x_1, x_2, \cdots, x_n)\right) = 0$$

- The solution can be approximated to a NN model:

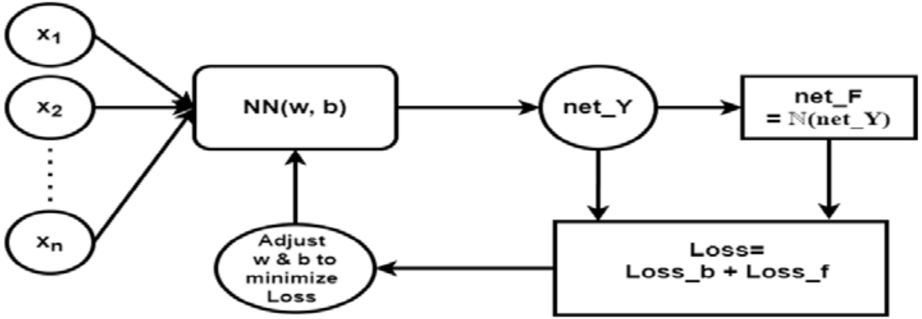
$$Y(x_1, x_2, \cdots, x_n) \cong net\_Y(x_1, x_2, \cdots, x_n)$$

- The PDE model can be constructed as:

$$net\_F := \mathbb{N}\left(net\_Y(x_1, x_2, \cdots, x_n)\right) \cong 0$$

- The shared learnable parameters can be learned by restricting the predictions of (***net_F***) to zero

Forward PINN Framework (Cont.)



PINN Training Scheme

$$\mathbf{Loss} = \underbrace{\frac{1}{N_b}\sum_{i=1}^{N_b}\left[net\_Y(x_1^i, x_2^i, \ldots x_n^i) - Y(x_1^i, x_2^i, \ldots x_n^i)\right]^2}_{\text{Boundary Conditions}} + \underbrace{\frac{1}{N_f}\sum_{i=1}^{N_f}\left[net\_F(x_1^i, x_2^i, \ldots x_n^i)\right]^2}_{\text{Neutron Balance (i.e., physics principle)}}$$

Boundary Conditions

Neutron Balance
(i.e., physics principle)

1G Fixed-source Diffusion Model

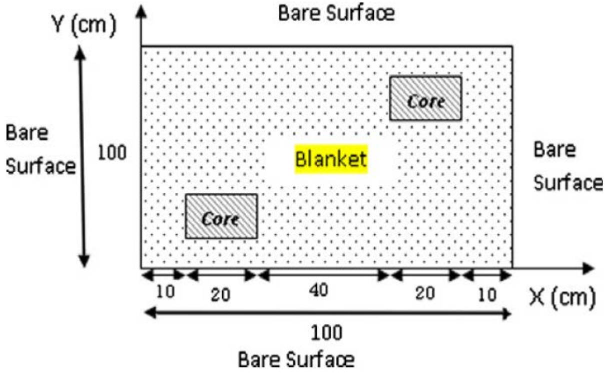
- **The 1G 2D steady state diffusion equation:**

$$\boldsymbol{F} \coloneqq -\nabla \cdot \left(D(x,y)\nabla\phi(x,y)\right) + \Sigma_a(x,y)\phi(x,y) - S(x,y) = 0$$

- **Zero-incoming fluxes are assumed for all boundary surfaces:**

For exmaple, at the surface $x = 0$:

$$\frac{1}{4}\phi(0,y) - \frac{1}{2}D\left.\frac{d\phi}{dx}\right|_{x=0} = 0$$



Geometry of one-group example

Region Material	Σ_a (cm^{-1})	Σ_s (cm^{-1})	S (n/cm^3)
Core	0.062158	0.089302	0.01048083
Blanket	0.064256	0.094853	0.00214231

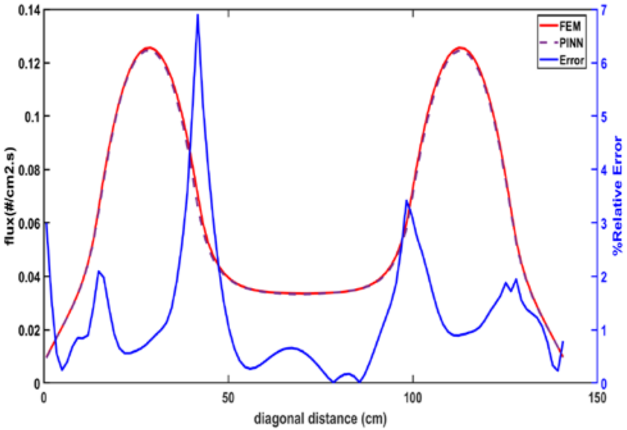
1G Model - Implementation Details

- LHS strategy was used to generate training points
- Adam optimizer was used to minimize the loss function
- A high-order FEM solution was used as a reference solution
- PINN optimum hyperparameters (after optimization)

# of hidden layers	# of neurons/layer	N_f	N_b/surface
8	40	10,000	100

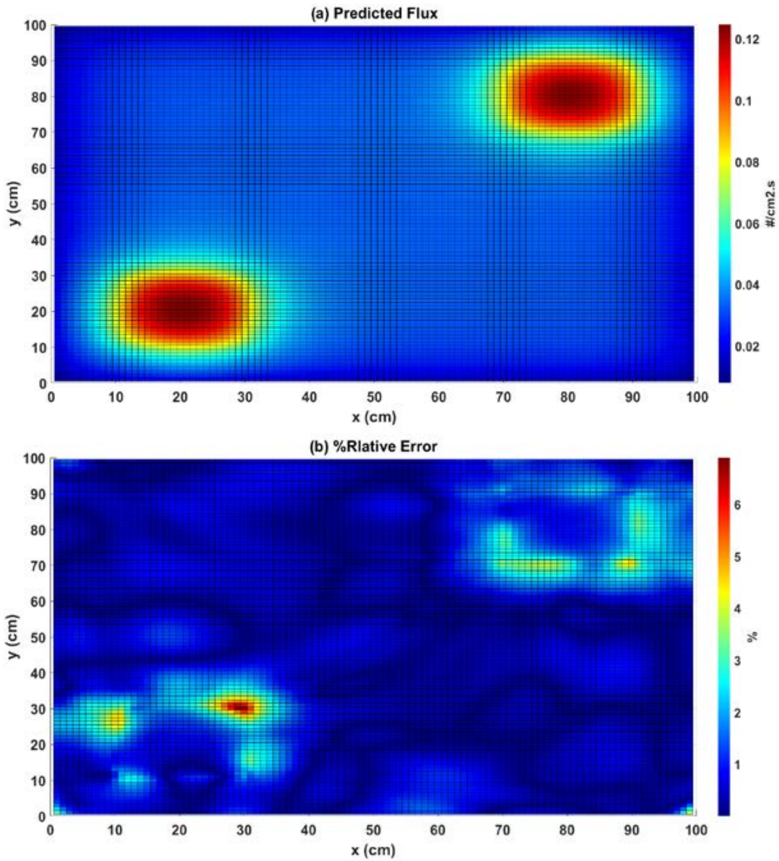
The optimum hyperparameters are shown below with mean percentage relative error of 0.69% and maximum error of 6.9% in flux solution.

1G Model - Results



Predicted flux and percentage relative error along the diagonal line.

Relative percentage error	
Mean	0.69%
Std.	0.74
Max	6.9%



PINN predicted flux distribution (a) and relative percentage error distribution compared to the FEM solution (b).

2G Fixed-source Diffusion Model

$$\begin{cases} f_1 := -\nabla(D_1 \nabla\phi_1) + \Sigma_{r,1}\phi_1 - \nu\Sigma_{f,2}\phi_2 - Q_1 = 0 \\ f_2 := -\nabla(D_2 \nabla\phi_2) + \Sigma_{a,2}\phi_2 - \Sigma_{s,1\to 2}\phi_1 \quad = 0 \end{cases}$$

- **Why it's different?**
  - **Joint learning task**
  - **Generally multi-scale optimization problem**

$$\begin{bmatrix} \phi_1(x,y) \\ \phi_2(x,y) \end{bmatrix} = NN(x,y)$$

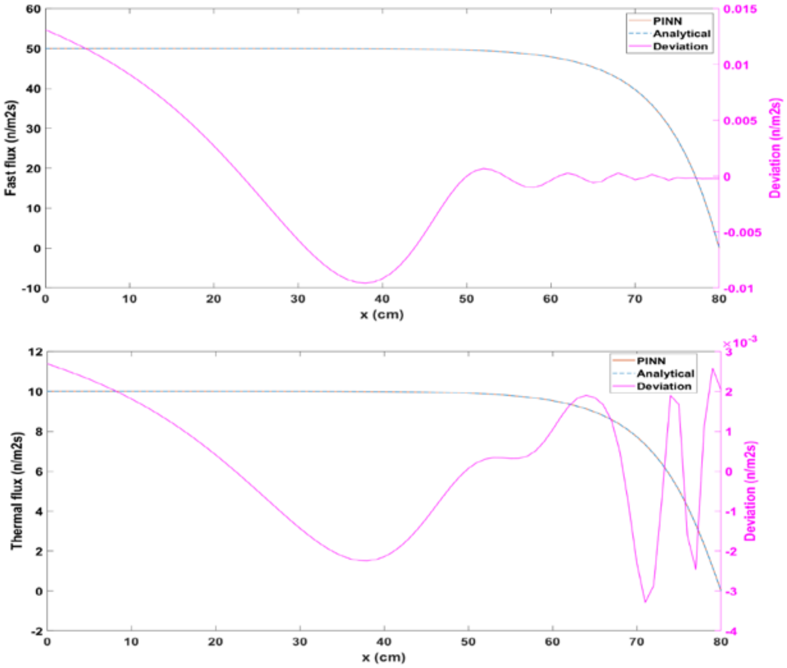
$$\text{Loss} = \underbrace{\sum_{j=1}^{N_b} \left| \begin{matrix} f_{T1}(x_j^T, y_j^T) \\ f_{T2}(x_j^T y_j^T, \end{matrix} \right|^2}_{\text{Boundary Conditions}} + \underbrace{\sum_{i=1}^{N_f} \left| \begin{matrix} f_1(x_i^f, y_i^f) \\ f_2(x_i^f, y_i^f) \end{matrix} \right|^2}_{\text{Neutron Balance}} + \cdots$$

2G1D Exemple - Homogenous Slab

- A **80-cm** thick slab is composed of a single material.
- BCs:
  - Reflective BC on left
  - Zero-flux BC on right

D_1 (cm)	D_2 (cm)	$\Sigma_{r,1}$ (cm^{-1})	$\Sigma_{a,2}$ (cm^{-1})	$\Sigma_{s,1\to2}$ (cm^{-1})	Q_1 $(\mathrm{cm}^{-3}\mathrm{s}^{-1})$
1.2	0.4	0.03	0.1	0.02	1.5

- MAE results:
  - Fast flux: **4.3E-3**
  - Thermal flux: **1.4E-3**



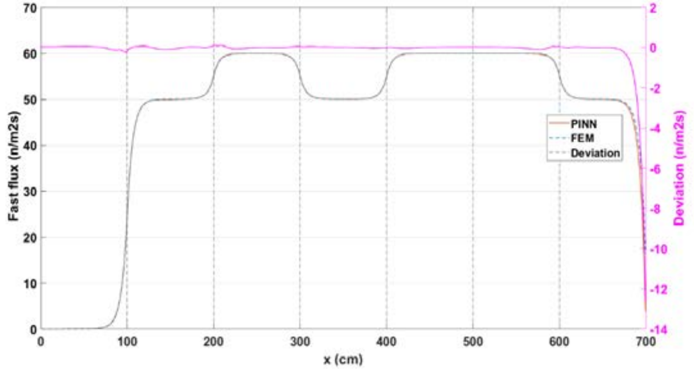
VCU | Virginia Commonwealth University
Department of Mechanical & Nuclear Engineering

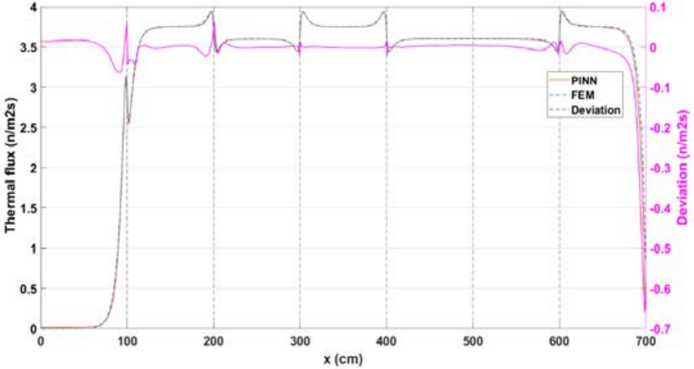
2G1D Exemple – Heterogeneous Slab

- Seven **100-cm** thick assemblies
- Material arrangement: **1-2-3-2-3-3-2**
- BCs:
  - Zero-flux BC for both sides

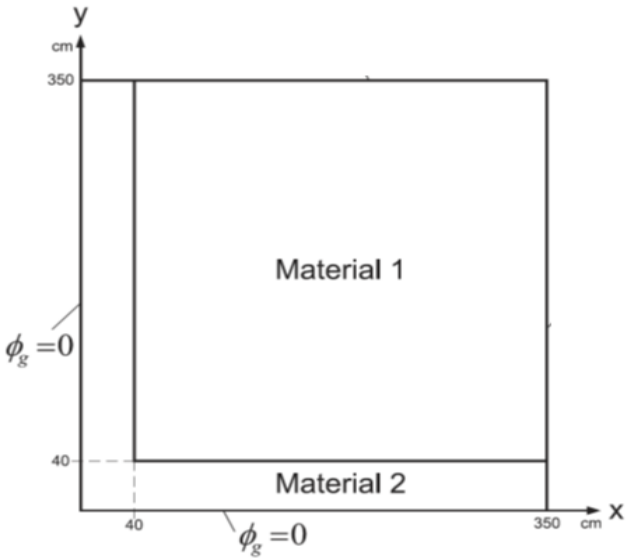
Materials	D_1 (cm)	D_2 (cm)	$\Sigma_{r,1}$ (cm^{-1})	$\Sigma_{a,2}$ (cm^{-1})	$\Sigma_{s,1\to 2}$ (cm^{-1})	Q_1 $(\mathrm{cm}^{-3}\mathrm{s}^{-1})$
1	1.2	0.4	0.03	0.1	0.02	0
2	1.2	0.4	0.03	0.2	0.015	1.5
3	1.2	0.4	0.03	0.25	0.015	1.8

- MAE results:
  - Fast flux: **1.54E-1**
  - Thermal flux: **1.52E-2**





2G2D Exemple – Problem Description

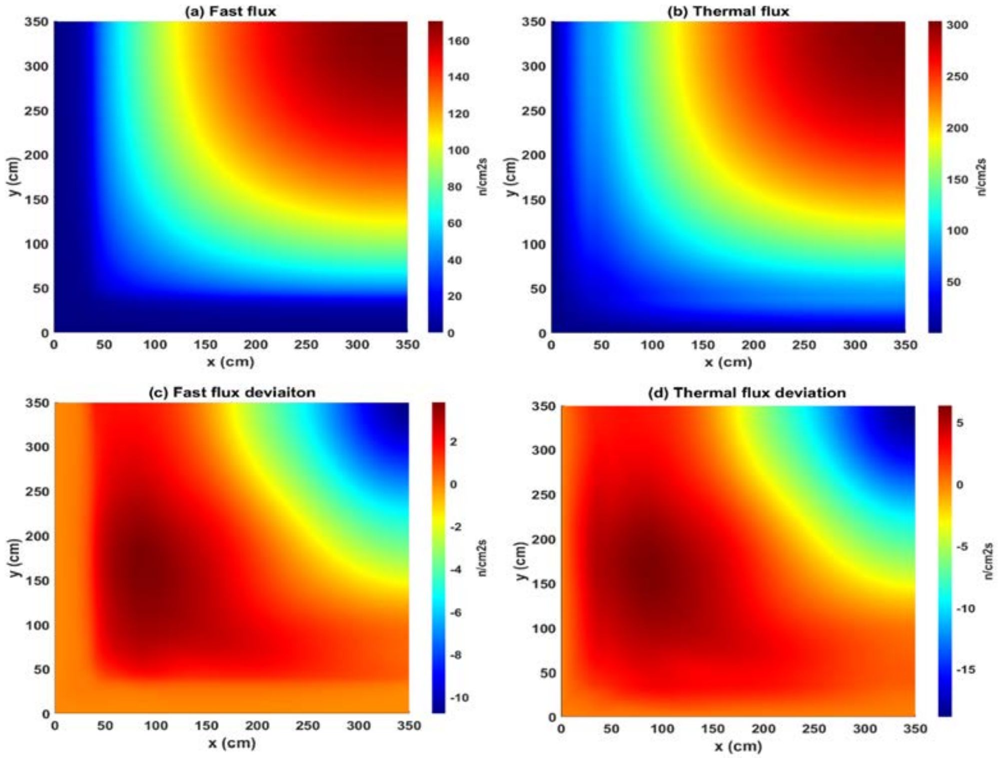


	Material 1	Material 2
D_1 [cm]	1.269	1.31
D_2 [cm]	0.9328	0.8695
$\Sigma_{a,1}$ $[cm^{-1}]$	7.86E-4	0
$\Sigma_{a,2}$ $[cm^{-1}]$	4.1E-3	2.117E-4
$\Sigma_{s,1\to2}$ $[cm^{-1}]$	7.368E-3	1.018E-2
$\nu\Sigma_{f,2}$ $[cm^{-1}]$	4.562E-3	0
Q_1 $[cm^{-3}s^{-1}]$	0.01	0

BCs:

- Reflective BC on right & top
- Zero-flux BC on left & bottom

2G2D Exemple – Flux Distribution

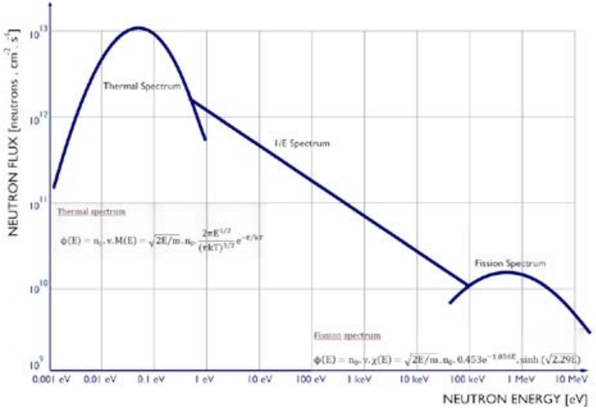


- MAE results:
  - Fast flux: **2.13**
  - Thermal flux: **3.98**

- Considering the average flux value for the fast and thermal groups, the relative error values are within the **3%** and **2%** range, respectively

Future Work

- **Multigroup (G>2) Challenges**
  - *Operator stiffness*
  - *Computational cost*



- ***k*-eigenvalue Problem Challenges**

$$F := \frac{1}{k}\nu\Sigma_f(x,y)\phi(x,y) + \left[\frac{\partial}{\partial x}\left(D\frac{\partial\phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(D\frac{\partial\phi}{\partial y}\right)\right] - \Sigma_a(x,y)\phi(x,y) = 0$$

  - *Parametric equation (unknown **k**)*
  - *Homogenous (Direct minimization of **F** results in $\phi(x,y) = 0$)*

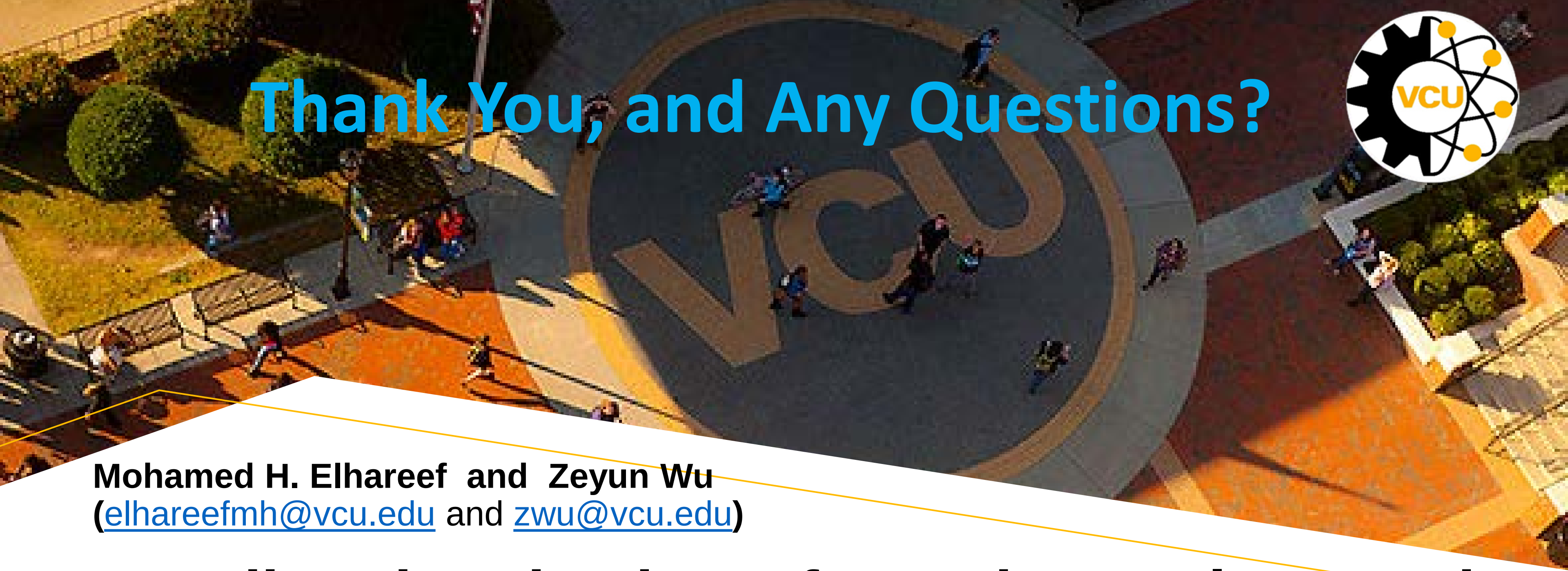
Conclusions

- **Advantages:**
  1. Obtain **mesh-free** solutions
  2. **No large amount of training data needed ahead**
  3. Achieve the same level of accuracy as conventional methods.
  4. Manpower efforts for the PINN can be significantly reduced.
- **Challenges:**
  1. Computational complexity
  2. Applications to higher dimensionality problems
  3. Multi-scale optimization

References:

1. M. RAISSI et al., "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, 378, (2019).
2. A. GRIEWANK and A. WALTHER, *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*, 2nd Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA (2008).
3. M. H. ELHAREEF, Z. WU, and Y. MA, "Physics-Informed Deep Learning Neural Network Solution to the Neutron Diffusion Model," *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2021)*, Raleigh, North Carolina (2021).
4. M. H. ELHAREEF and Z. WU, "Extension of the PINN Diffusion Model to k-eigenvalue Problems", submitted to the PHYSOR 2022 Conference, May 15-20, Pittsburg, PA (2022).
5. J. P. SENECAL and W. JI, "Characterization of the proper generalized decomposition method for fixed-source diffusion problems", *Annals of Nuclear Energy*, 126, (2019)
6. M. ABADI et al., "Tensorflow: large-scale machine learning on heterogeneous distributed systems," *arXiv:1603.04467* (2016).
7. D. LIU and J. NOCEDAL, "On the limited memory BFGS method for large scale optimization," *Math. Program*, 45, (1989).

Thank You, and Any Questions?



Mohamed H. Elhareef and Zeyun Wu
(elhareefmh@vcu.edu and zwu@vcu.edu)

Extending the Physics Informed Neural Network Model to Multigroup Neutron Diffusion Problems

ANS Student Conference, April 14-16, 2022

VCU College of Engineering
Mechanical and Nuclear Engineering