# Adaptive activation functions accelerate convergence in deep and physics-informed neural networks

Ameya D. Jagtap, George Em Karniadakis

Journal of Computational Physics, 2020

# PINN

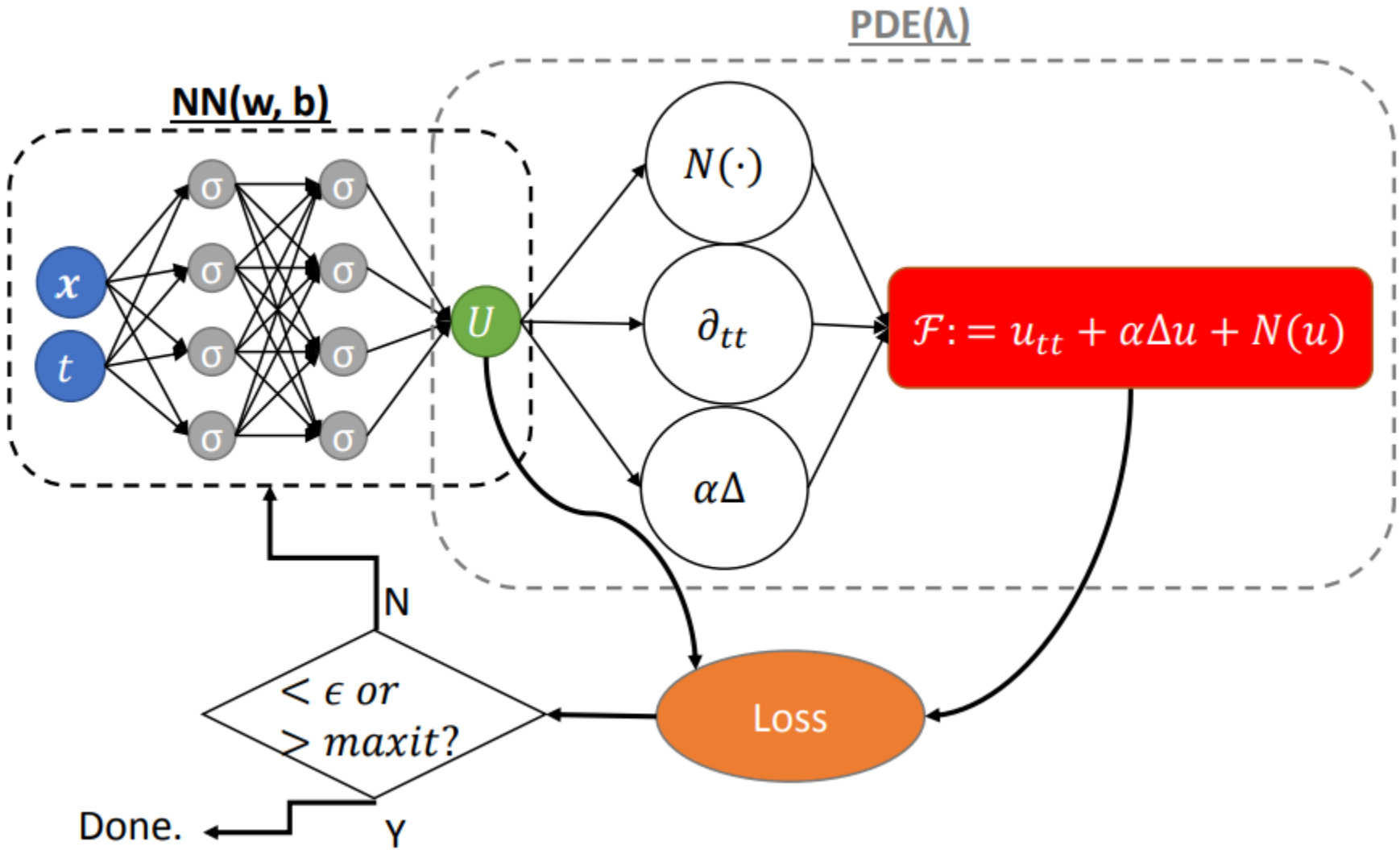
$$f := u_t + \mathcal{N}[u],$$

Burgers equation (fluid mechanics, nonlinear acoustics, gas dynamics, traffic flow) :

$$\begin{cases} u_t + uu_x - (0.01/\pi)u_{xx} = 0, \quad x \in [-1, 1], \quad t \in [0, 1], \\ u(0, x) = -\sin(\pi x), \\ u(t, -1) = u(t, 1) = 0. \end{cases}$$

$$f := u_t + uu_x - (0.01/\pi)u_{xx}, \quad MSE_u = \frac{1}{N_u}\sum_{i=1}^{N_u} |u(t_u^i, x_u^i) - u^i|^2, \quad MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f} |f(t_f^i, x_f^i)|^2.$$

$$MSE = MSE_u + MSE_f,$$




Schematic of PINN for the Klein-Gordon equation

# Adaptive activation functions

- 1. activation function supposed to be nonlinear
- 2. activation function must be differentiable
- 3. activation function better be less prone to the vanishing and the exploding gradient problem

$$\mathcal{L}_k(x^{k-1}) := w^k x^{k-1} + b^k, \quad u_{\mathbf{\Theta}}(x) = (\mathcal{L}_k \circ \sigma \circ \mathcal{L}_{k-1} \circ \ldots \circ \sigma \circ \mathcal{L}_1)(x),$$

$$J(\mathbf{\Theta}) = MSE_{\mathcal{F}} + MSE_u$$

$$w^* = \arg\min_{w \in \mathbf{\Theta}} (J(w)); \qquad b^* = \arg\min_{b \in \mathbf{\Theta}} (J(b)).$$

introduce the hyper-parameter a

$$\sigma(a\, \mathcal{L}_k(x^{k-1})), \quad a^* = \arg\min_{a \in \mathbb{R}^+ \setminus \{0\}} (J(a)).$$

such hyper-parameter can change the slope of the activation function

$$\text{Sigmoid}: \frac{1}{1+e^{-ax}},$$

$$\text{Hyperbolic tangent}: \frac{e^{ax}-e^{-ax}}{e^{ax}+e^{-ax}},$$

$$\text{ReLU}: \max(0, ax),$$

$$\text{Leaky ReLU}: \max(0, ax) - \nu\, max(0, -ax).$$

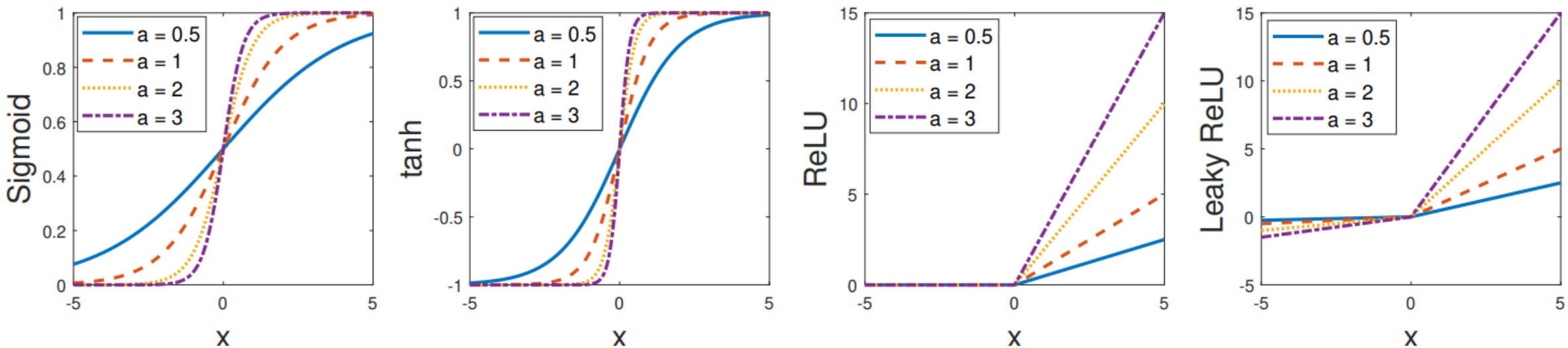



Figure 2: (Left to right) Sigmoid or logistic, tanh, ReLU and Leaky-ReLU activation functions for various values of $a$.

In order to accelerate convergence towards global minima, thinking a scale factor n ⩾ 1 multiplied by a:

$$\sigma(na\ \mathcal{L}_k(x^{k-1})).$$

$$u_{\tilde{\Theta}}(x) = (\mathcal{L}_k \circ \sigma \circ na\mathcal{L}_{k-1} \circ \sigma \circ na\mathcal{L}_{k-2} \circ \ldots \circ \sigma \circ na\mathcal{L}_1)(x).$$

**Algorithm 1:** Adaptive activation function based PINN algorithm

**Step 1** : Specification of training set

Training data : $u_{NN}$ network $\{x_u^i, y_u^i, t_u^i\}_{i=1}^{N_u}$.

Residual training points : $\mathcal{F}$ network $\{x_f^i, y_f^i, t_f^i\}_{i=1}^{N_f}$.

**Step 2** : Construct neural network $u_{NN}(\tilde{\mathbf{\Theta}})$ with random initialization of parameters $\tilde{\mathbf{\Theta}}$.

**Step 3** : Construct the residual neural network $\mathcal{F}$ by substituting surrogate $u_{NN}$ into the governing equations using automatic differentiation [5] and other arithmetic operations.

**Step 4**: Specification of loss function:

$$J(\tilde{\mathbf{\Theta}}) = \frac{1}{N_f}\sum_{i=1}^{N_f}|\mathcal{F}(x_f^i, y_f^i, t_f^i)|^2 + \frac{1}{N_u}\sum_{i=1}^{N_u}|u^i - u(x_u^i, y_u^i, t_u^i)|^2.$$

**Step 5**: Find the best parameters using suitable optimization method for minimizing the loss function

$$\tilde{\mathbf{\Theta}}^* = \arg\min(\tilde{\mathbf{\Theta}}).$$

# Neural network approximation of nonlinear smooth and discontinuous functions

- neural network learns simple pattern first before memorizing (found by Arpit, et al.)
- neural network learns low frequencies first (found by Rahaman, et al.)
- the amplitude of each frequency component of the network output is controlled by the spectral norm of the small-size network: longer training time allows the network to learn complex functions by allowing it to capture the high frequencies components in the solution
- In this case:

  activation function: tanh

  the number of hidden layers: 4

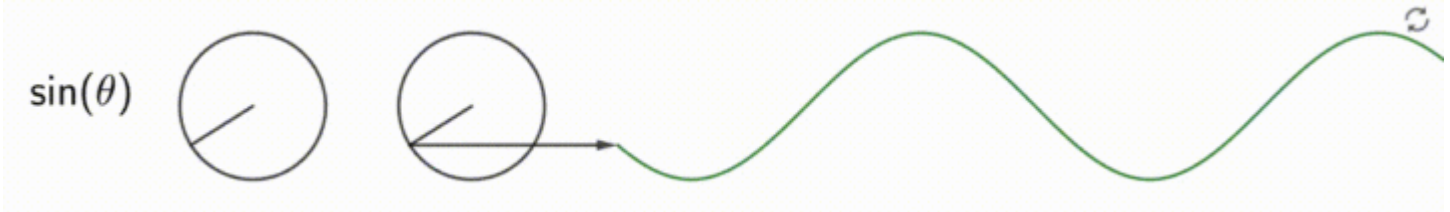
  each layer: 50 neurons

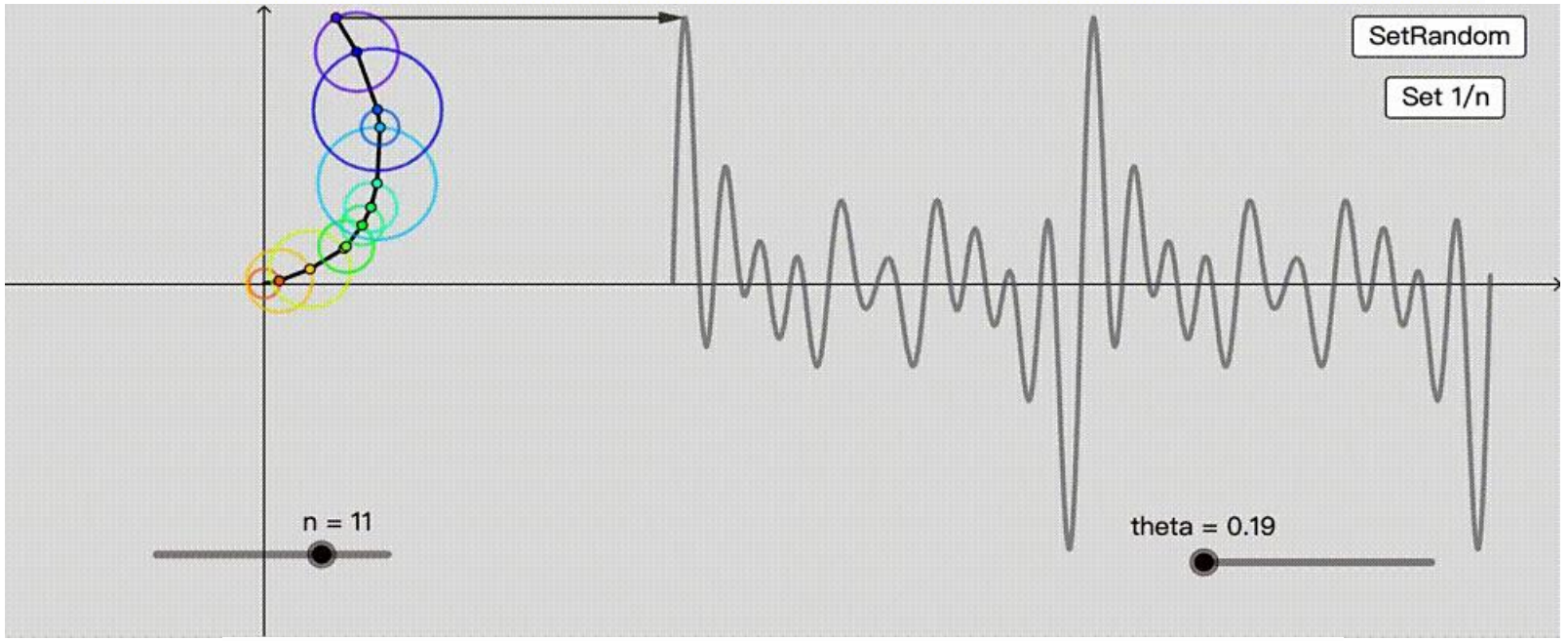
  loss function:

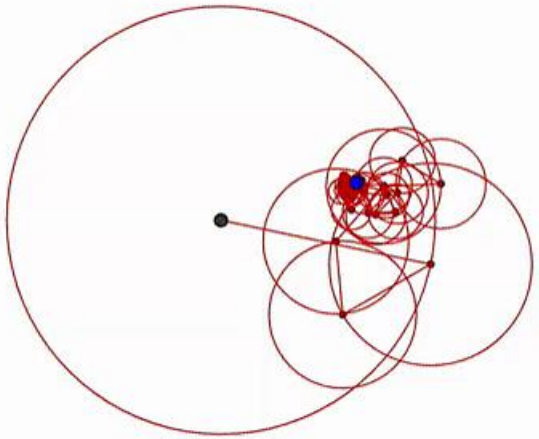
$$J(\tilde{\Theta}) = \frac{1}{N_u} \sum_{i=1}^{N_u} |u^i - u(x_u^i, y_u^i, t_u^i)|^2.$$

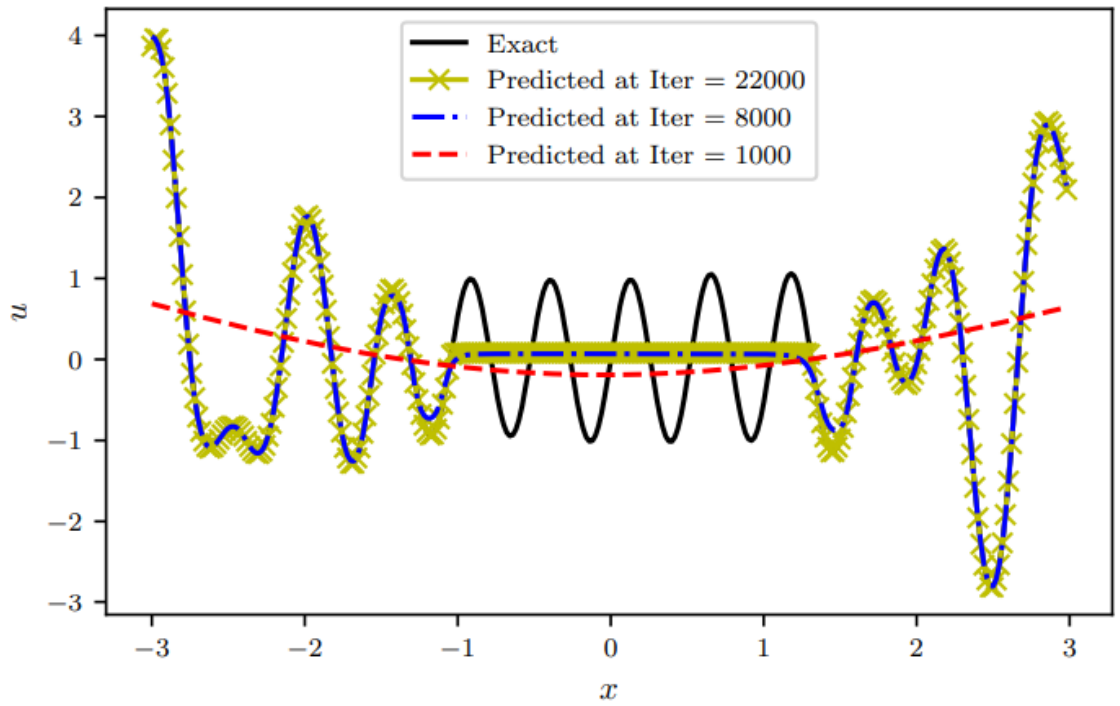
Fourier Transform

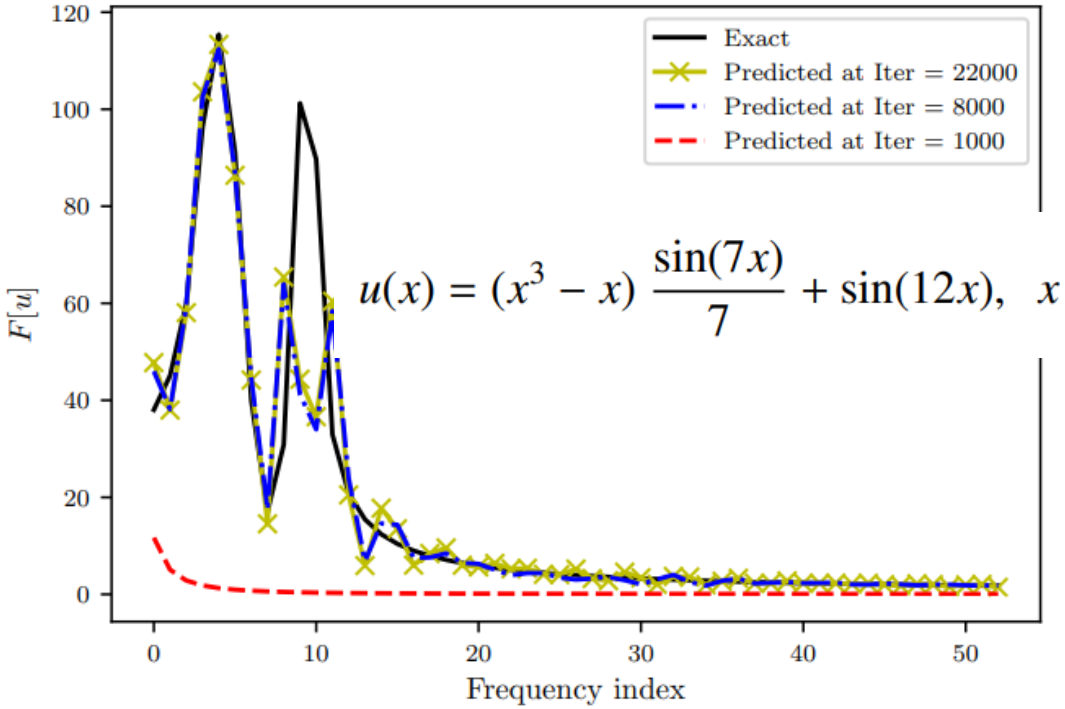
$$F(\omega) = \mathcal{F}[f(t)] = \int_{-\infty}^{\infty} f(t) e^{-iwt}\, dt$$



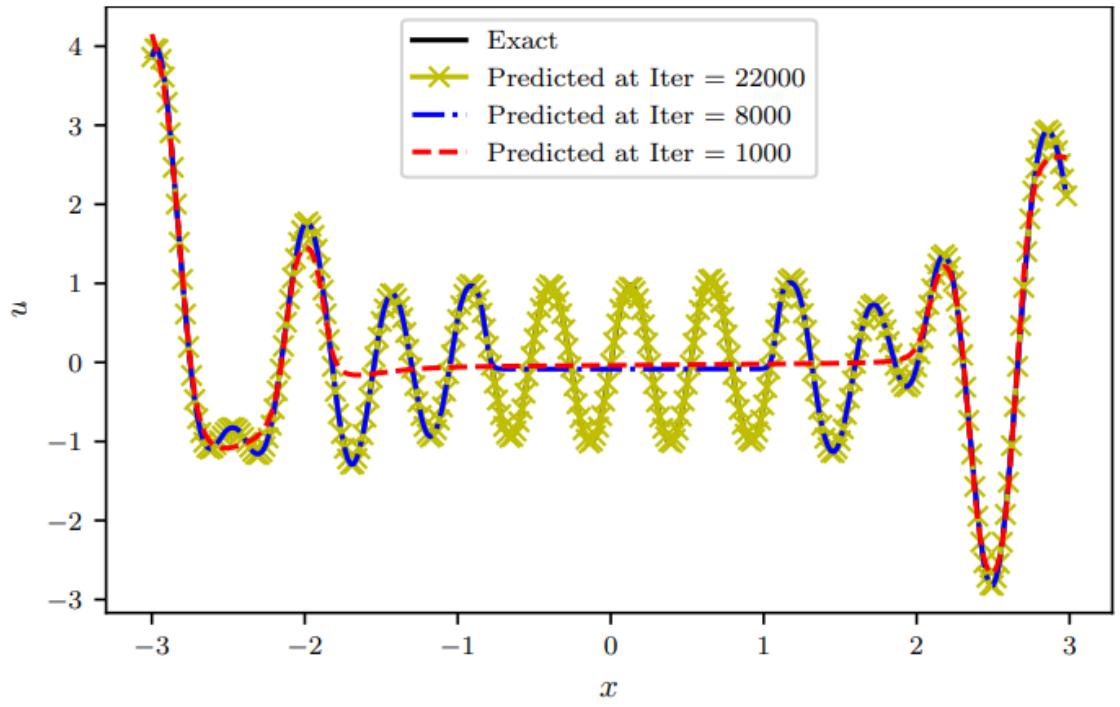



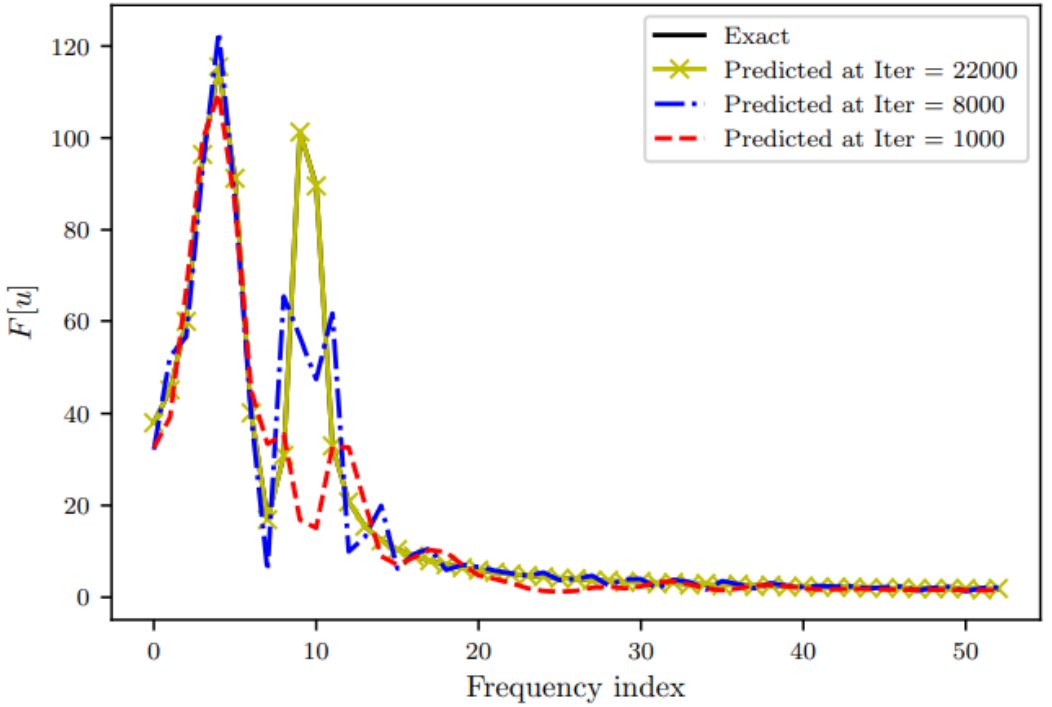






$$u(x) = (x^3 - x)\,\frac{\sin(7x)}{7} + \sin(12x), \quad x \in [-3,\ 3]$$

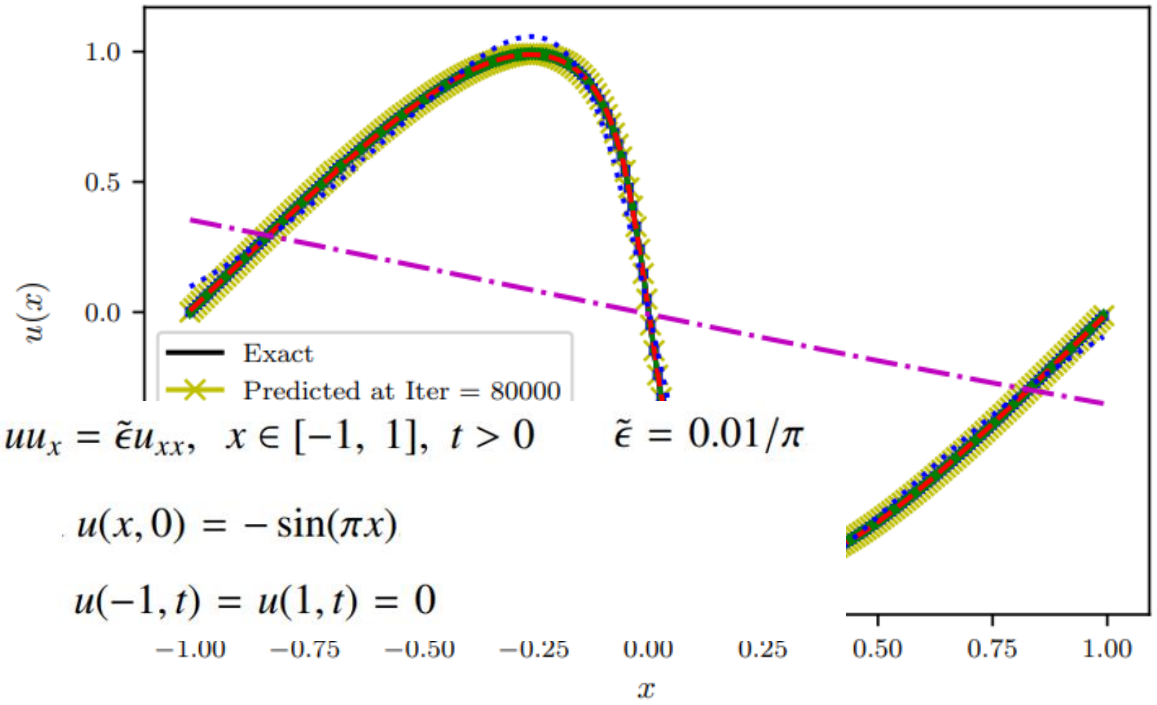


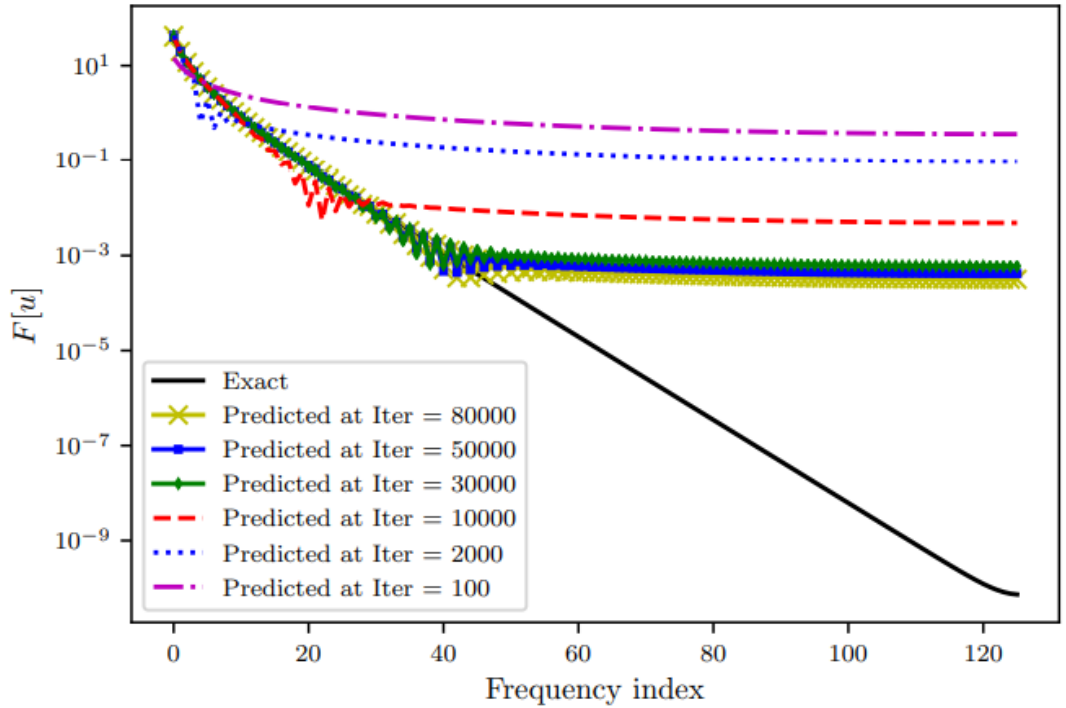



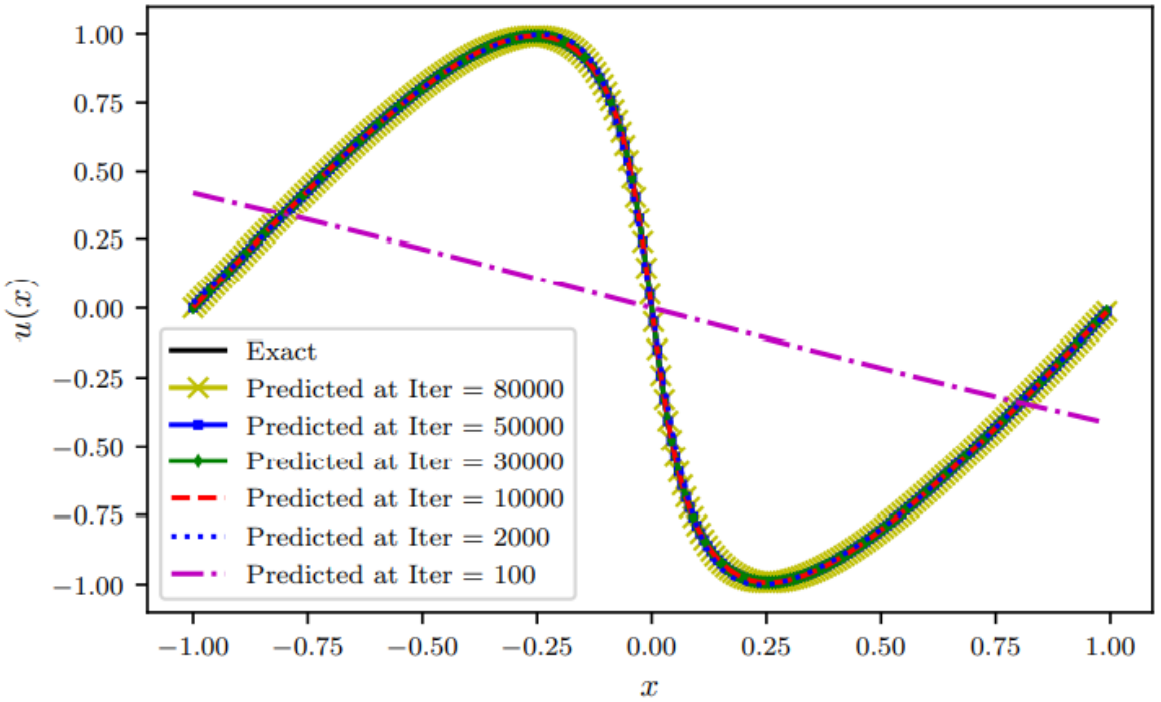
$$u_t + uu_x = \tilde{\epsilon} u_{xx}, \quad x \in [-1,\ 1],\ t > 0 \qquad \tilde{\epsilon} = 0.01/\pi$$

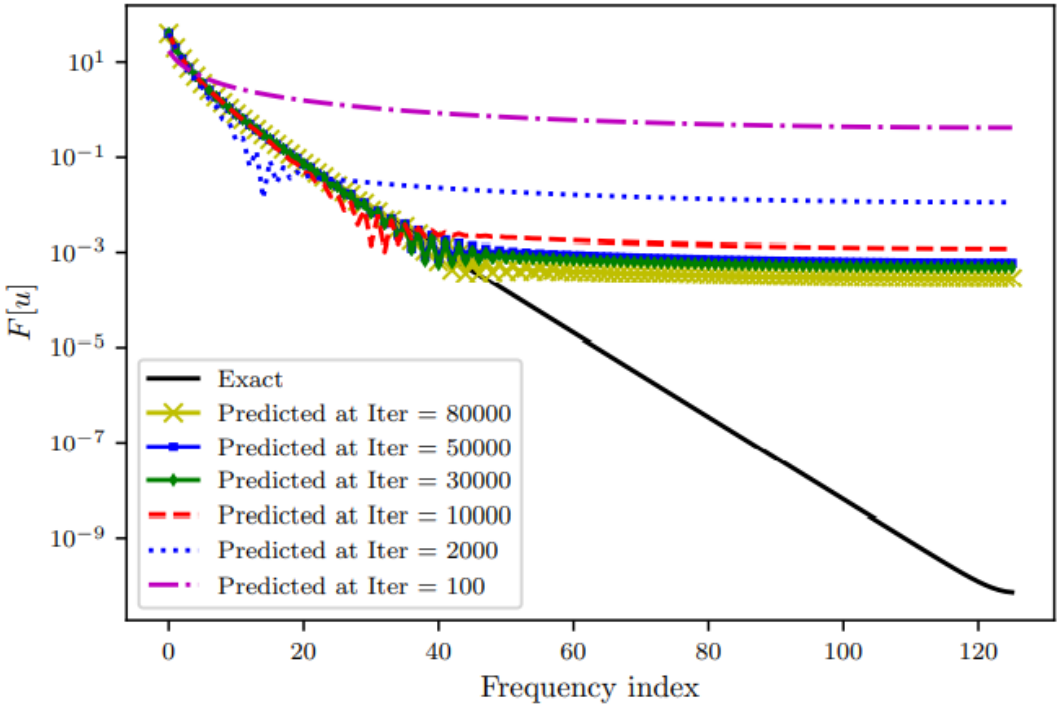
$$u(x, 0) = -\sin(\pi x)$$

$$u(-1, t) = u(1, t) = 0$$

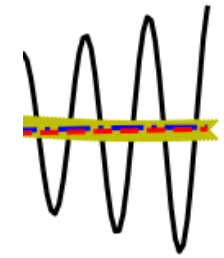


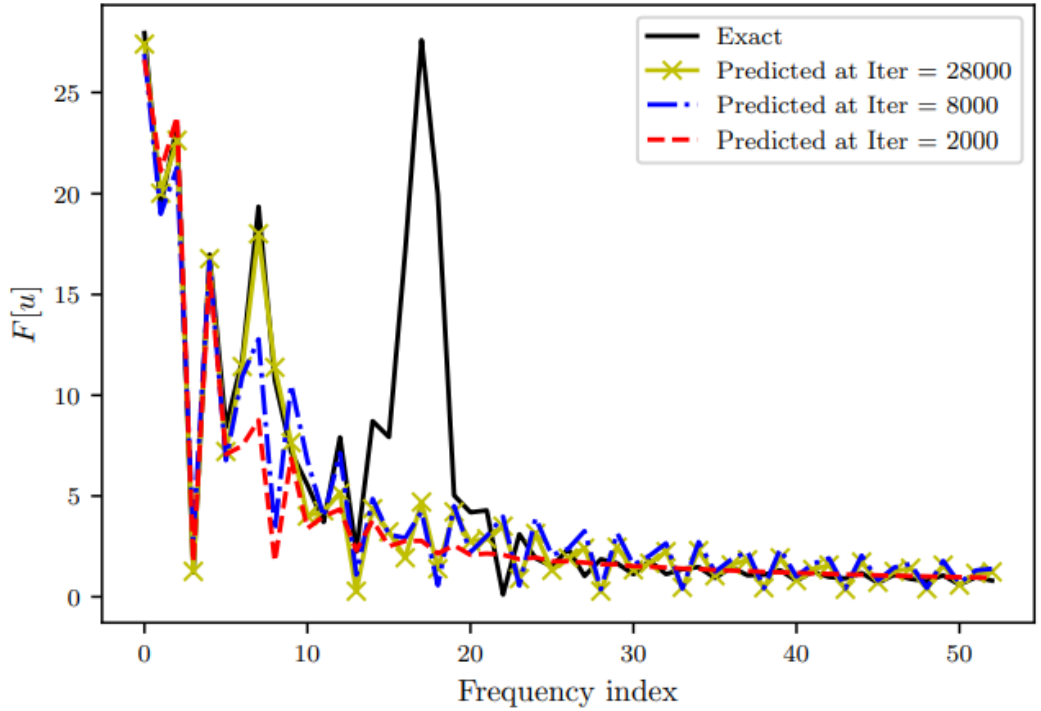



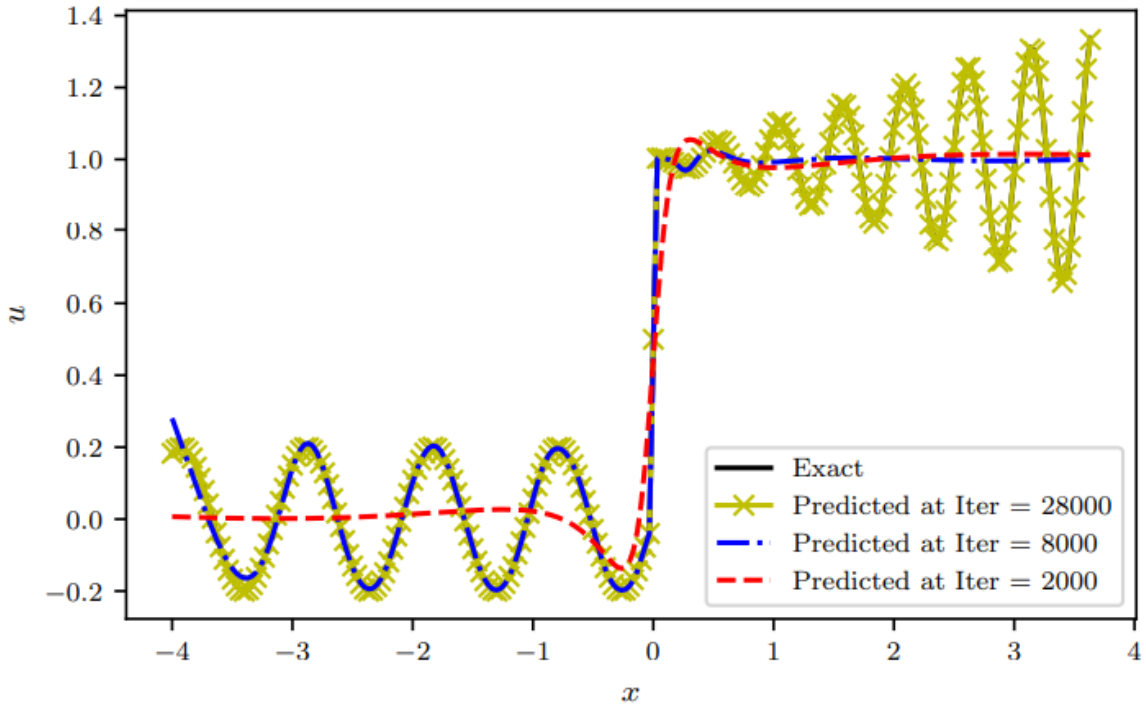



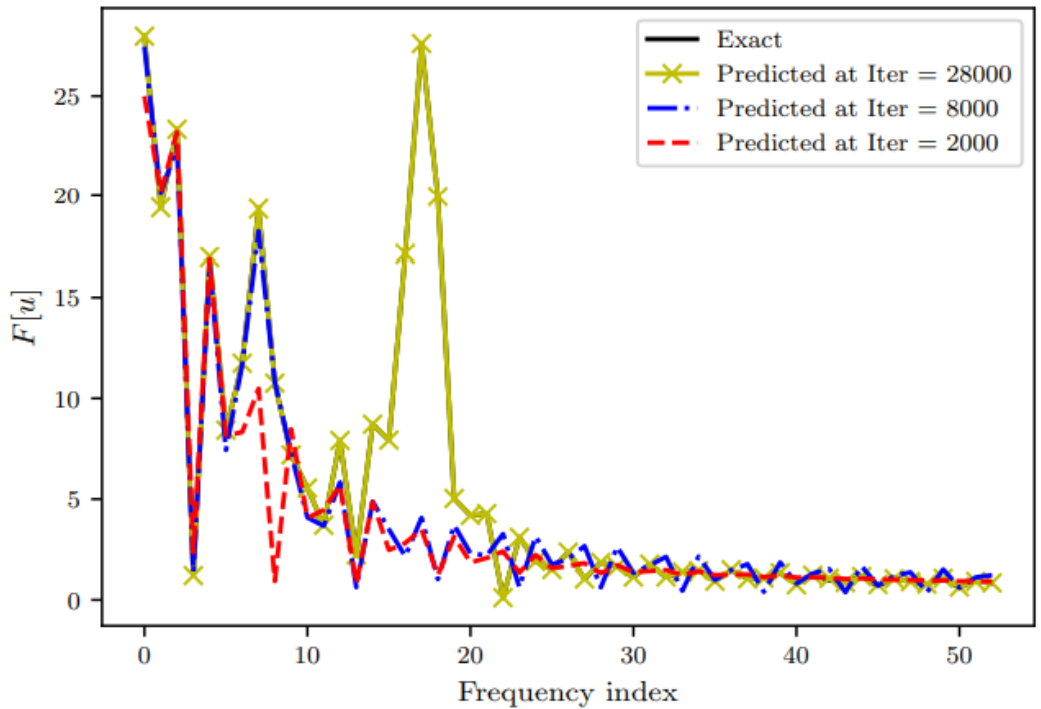
$$u(x) = \begin{cases} 0.2\ \sin(6x) & \text{If } x \le 0 \\ 1 + 0.1\ x\cos(12x) & \text{Otherwise.} \end{cases}$$

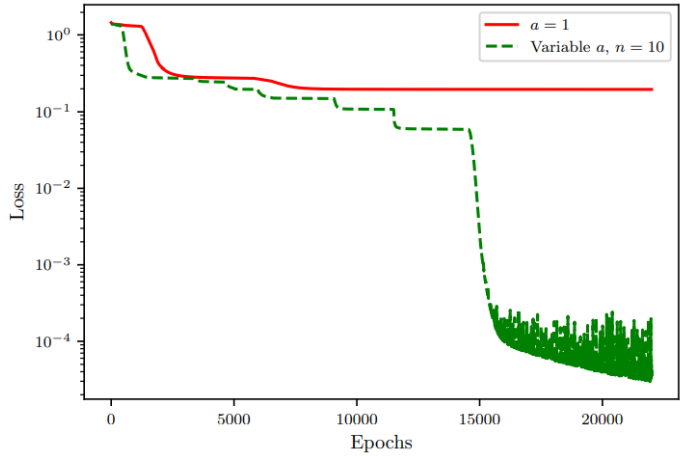


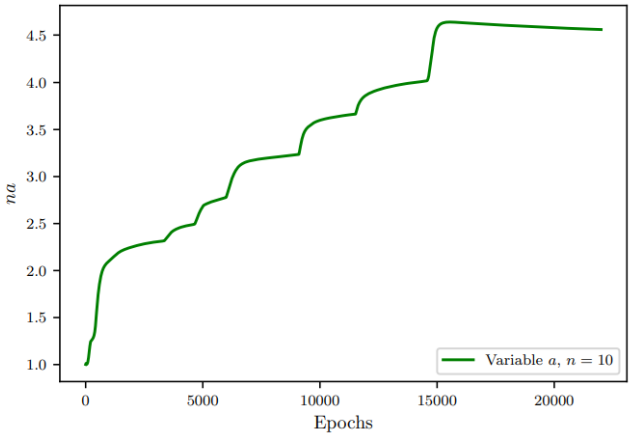


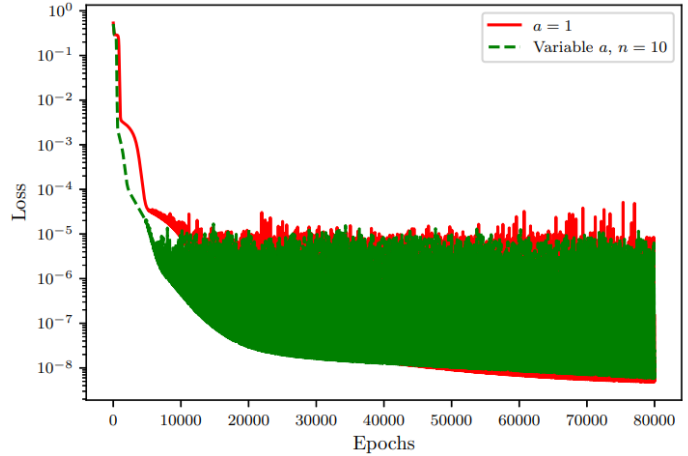



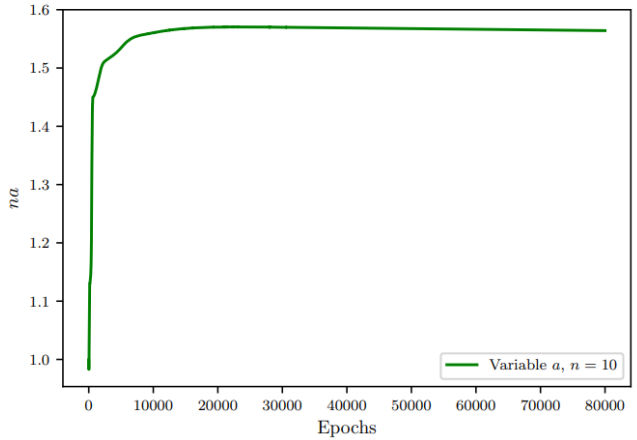



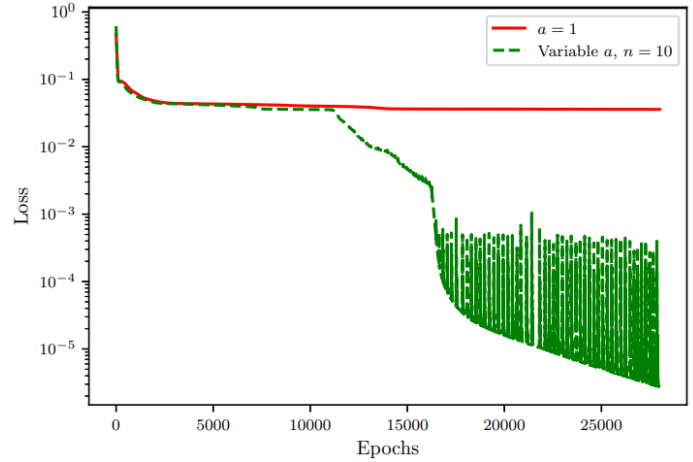



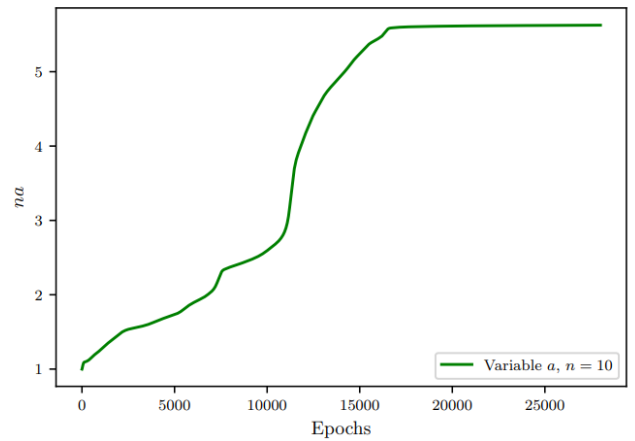

$a = 1$
Variable $a$, $n = 10$
Loss
Epochs


Variable $a$, $n = 10$
$na$
Epochs


$a = 1$
Variable $a$, $n = 10$
Loss
Epochs


Variable $a$, $n = 10$
$na$
Epochs


$a = 1$
Variable $a$, $n = 10$
Loss
Epochs


Variable $a$, $n = 10$
$na$
Epochs

## Burgers equation

$$\begin{cases} u_t + uu_x = \tilde{\epsilon} u_{xx}, \quad x \in [-1,\ 1],\ t > 0 \qquad \tilde{\epsilon} = 0.01/\pi \\ u(x,0) = -\sin(\pi x) \\ u(-1,t) = u(1,t) = 0 \end{cases}$$

$$\mathcal{F} := (u_{NN})_t + u_{NN}(u_{NN})_x - \tilde{\epsilon}(u_{NN})_{xx},$$

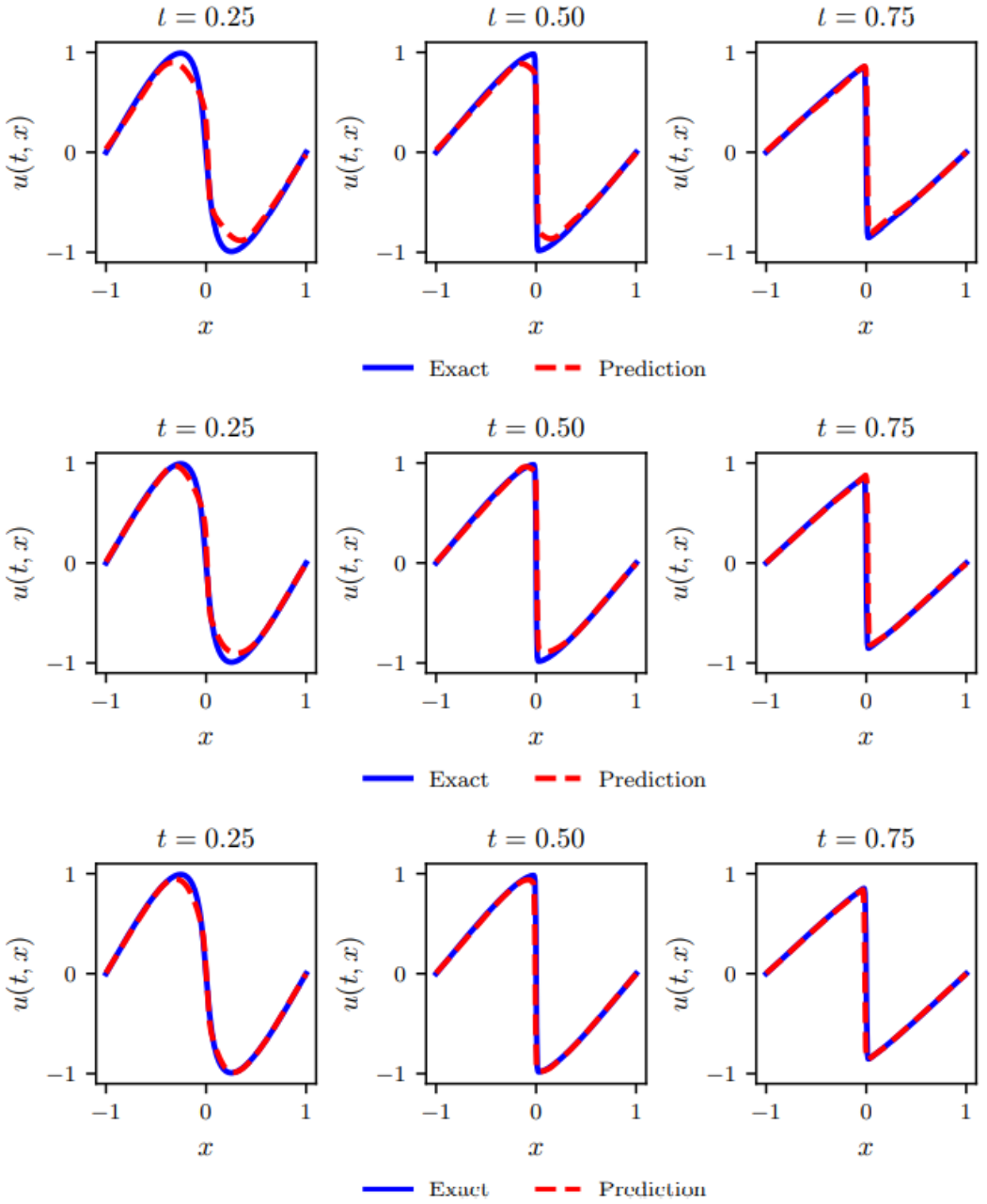



Figure 7: Burgers equation: Comparison of the exact solution with the solution given by PINN using $a = 1$ (top), variable $a, n = 1$ (middle) and variable $a, n = 5$ (bottom) obtained after 2000 iterations.

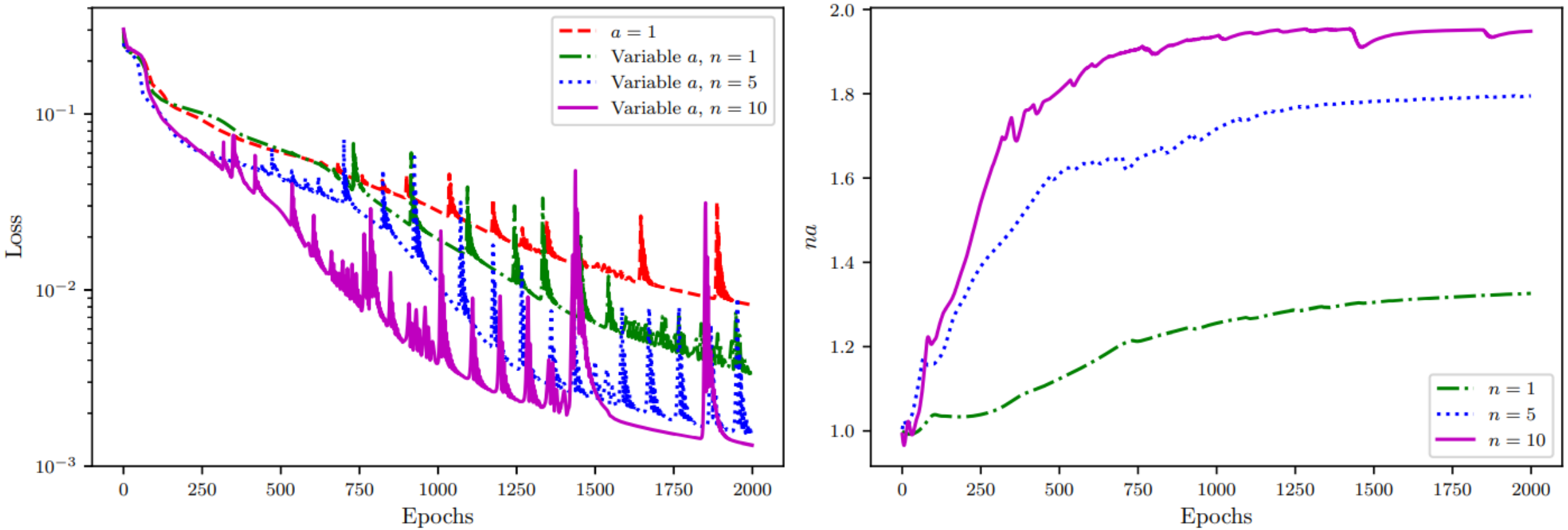



Figure 8: Burgers equation: Loss vs. epochs for fixed and variable $a$ with different values of $n$ (left) and corresponding variation in $na$ with epochs (right).

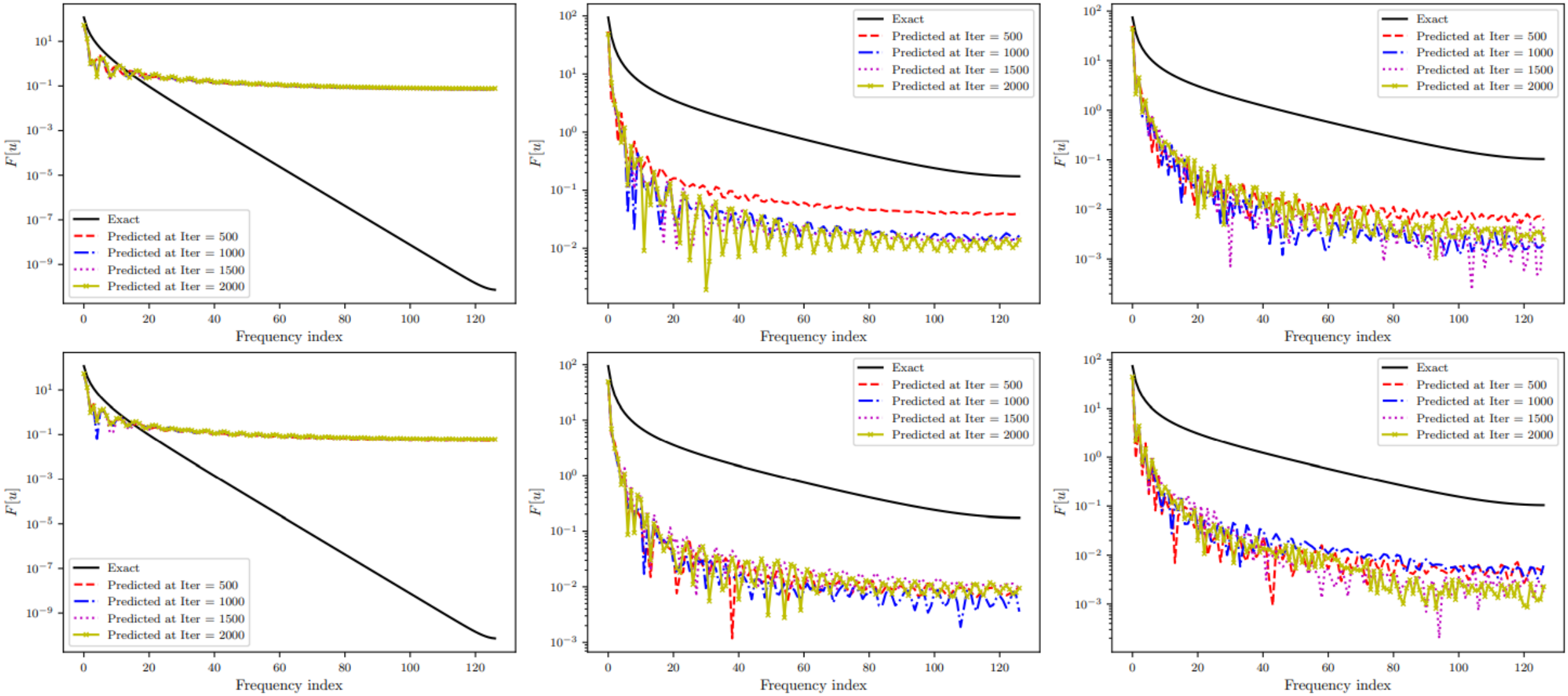



Figure 9: ReLU activation: Comparison of solution of Burgers equation in frequency domain with fixed (1st row) and variable $a, n = 5$ (2nd row) *'ReLU'* activation function. Columns (left to right) represent the solution in frequency domain at $t = 0.25, 0.5$ and $0.75$, respectively.

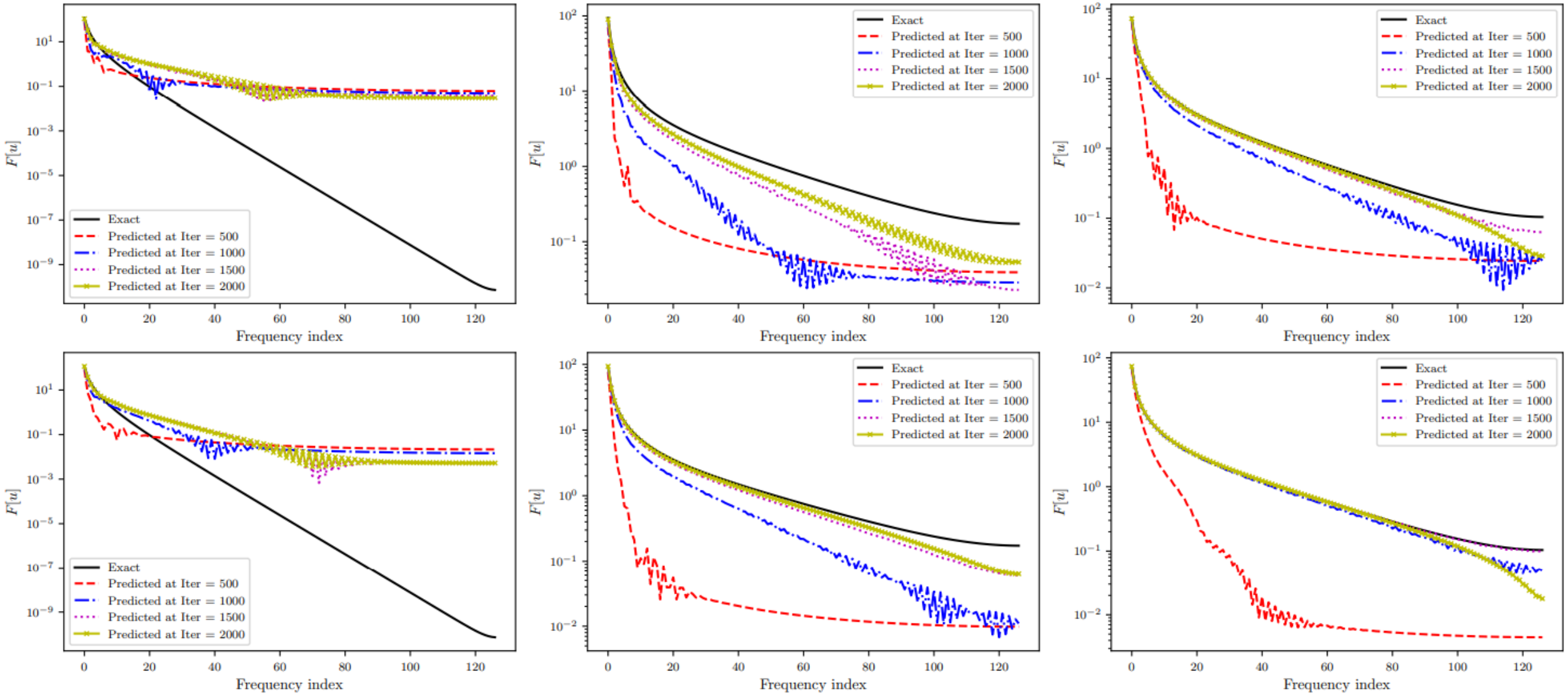



Figure 10: *'Tanh'* activation: Comparison of solution of Burgers equation in frequency domain with fixed (1st row) and variable $a, n = 5$ (2nd row) *'tanh'* activation function. Columns (left to right) represent the solution in frequency domain at $t = 0.25, 0.5$ and $0.75$, respectively.

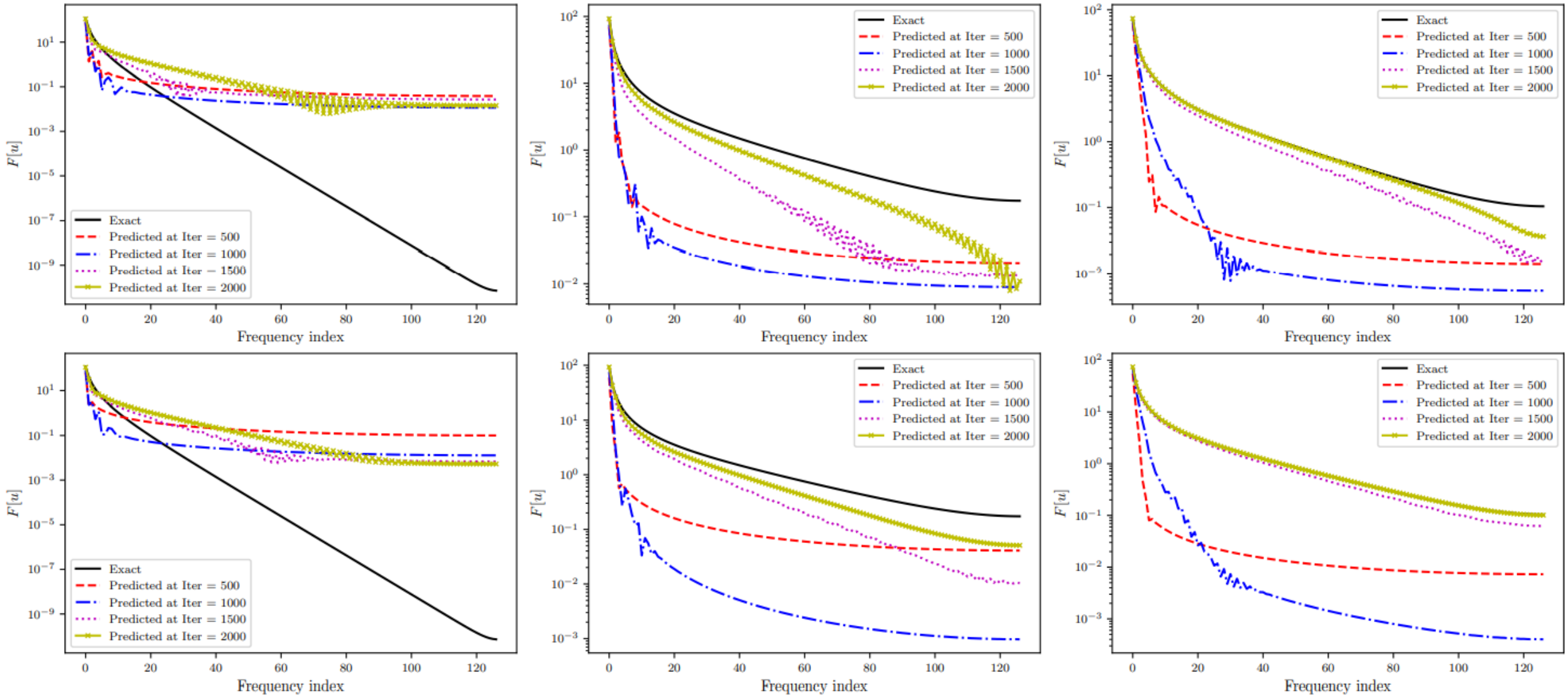



Figure 11: *'Sin'* activation: Comparison of solution of Burgers equation in frequency domain with fixed (1st row) and variable $a, n = 5$ (2nd row) *'sin'* activation function. Columns (left to right) represent the solution in frequency domain at $t = 0.25, 0.5$ and $0.75$, respectively.

# Klein-Gordon equation

$$u_{tt} + \alpha \Delta u + N(u) = h(x,t), \quad x \in [-1,\ 1],\ t > 0,$$

$$N(u) = \beta u + \gamma u^k$$

$$h(x,t) = -x\cos(t) + x^2\cos^2(t).$$

$$\alpha = -1, \beta = 0, \gamma = 1,\ k = 2$$

$$f(x) = x,\ g(x) = 0$$

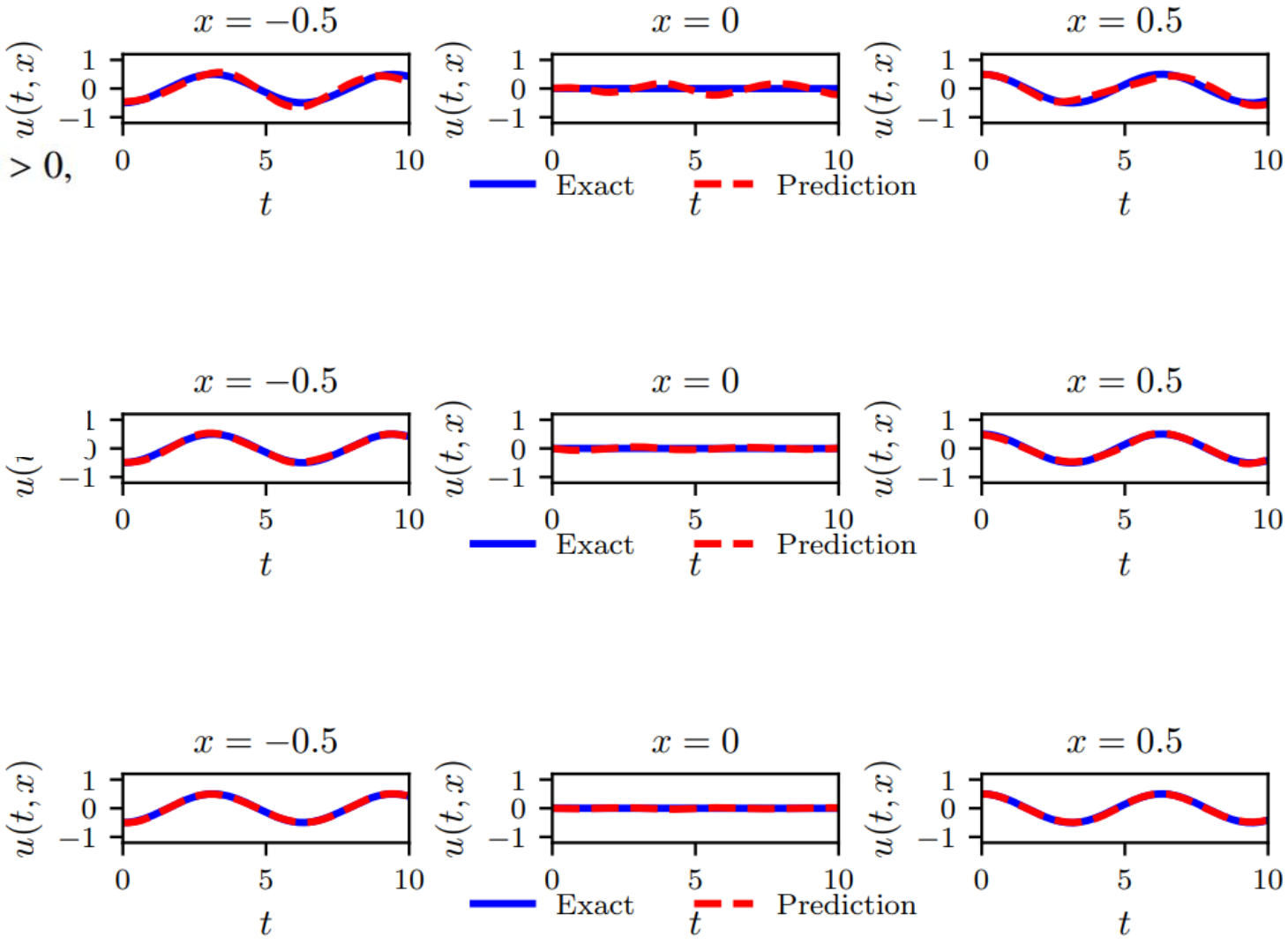



Figure 14: Klein-Gordon equation: Comparison of the exact solution with the solution given by PINN using $a = 1$ (top), variable $a, n = 1$ (middle) and variable $a, n = 5$ (bottom) obtained after 1400 iterations.

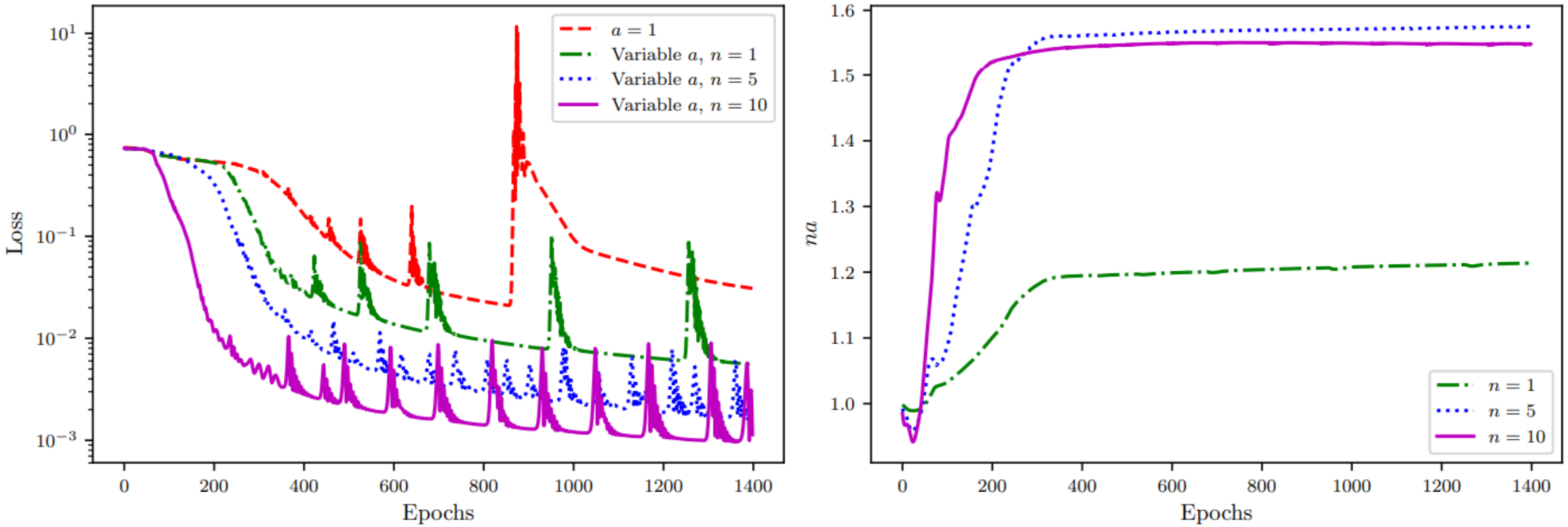



Figure 15: Klein-Gordon equation: Loss vs. epochs for the fixed and variable $a$ with different values of $n$ (left) and corresponding variation in $na$ with epochs (right).

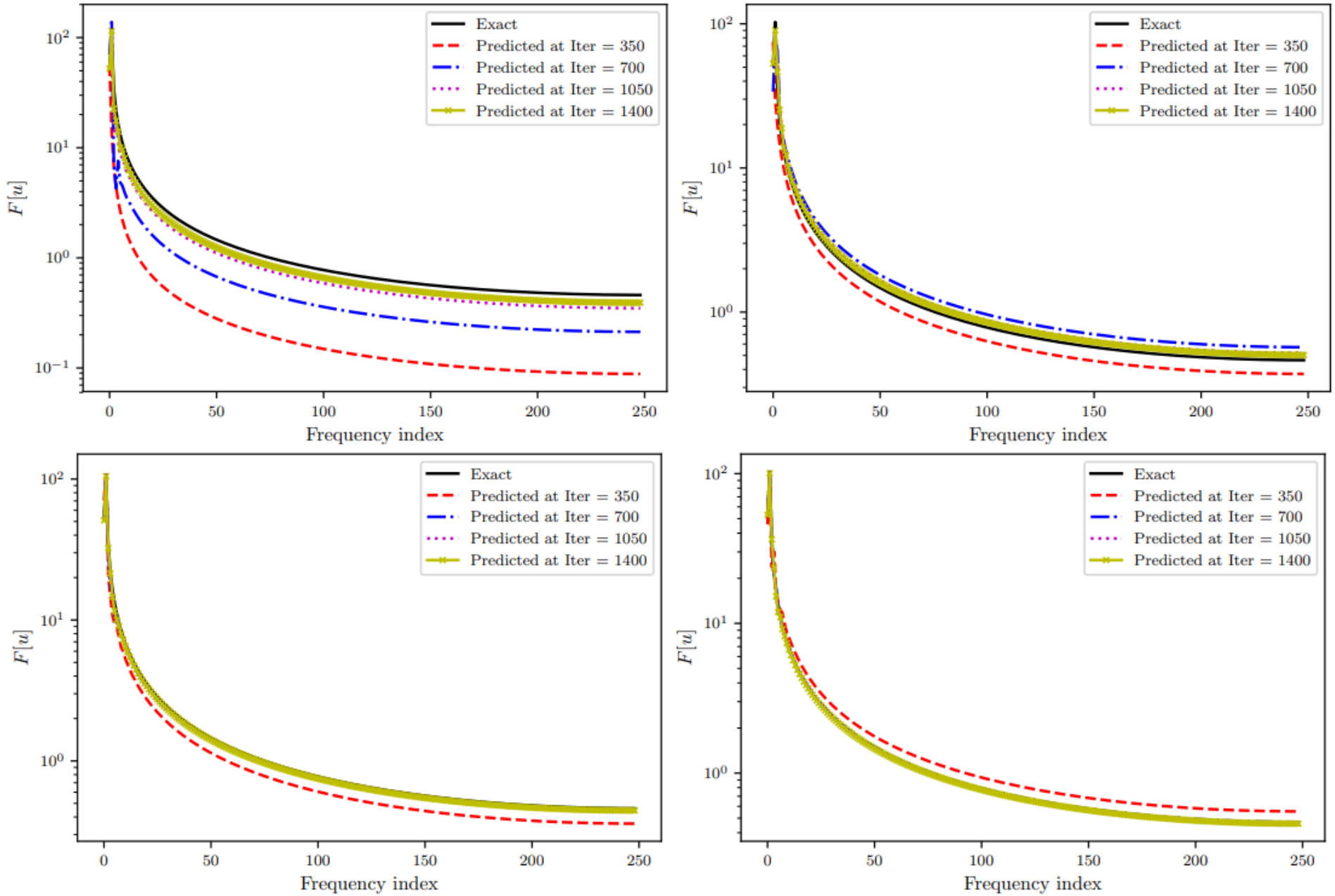



Figure 16: Comparison of solution of the Klein-Gordon equation in frequency domain with the fixed (1st row) and variable $a, n = 5$ (2nd row) *'tanh'* activation function. First column shows the frequencies in the solution at $x = -0.5$ whereas second column shows at $x = 0.5$.

## Helmholtz equation

$$\Delta u + k^2(u) = q(x,y), \quad (x,y) \in [-1,\ 1]^2$$

$$q(x,y) = 2\pi \cos(\pi\, y)\sin(\pi\, x) + 2\pi \cos(\pi\, x)\sin(\pi\, y) + \ (x+y)\sin(\pi\, x)\sin(\pi\, y) - 2\pi^2(x+y)\sin(\pi\, x)\sin(\pi\, y)$$

$$k = 1 \qquad u(x,y) = (x+y)\sin(\pi\, x)\sin(\pi\, y).$$

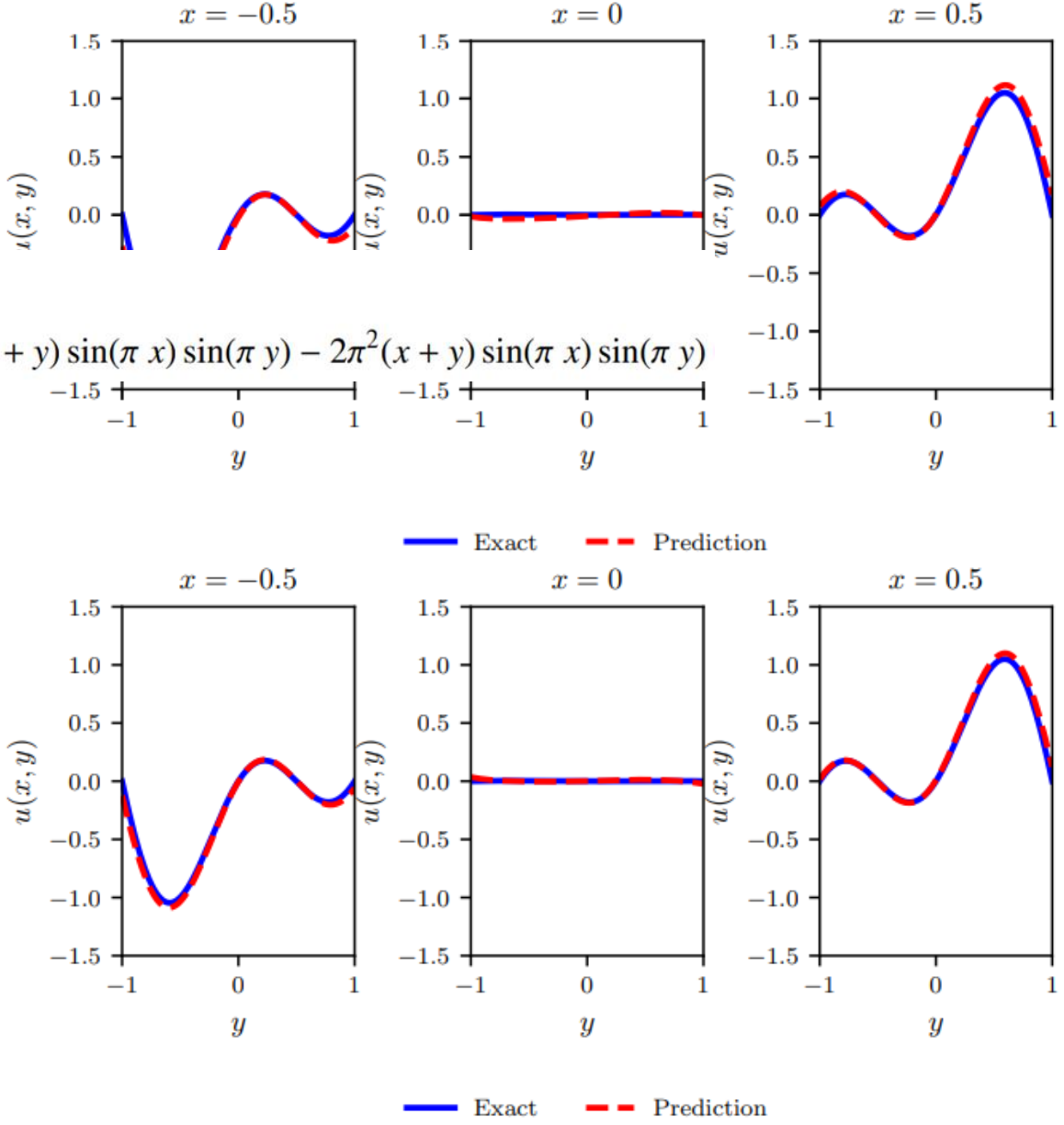



Figure 17: Contour plot (top) shows the solution of Helmholtz equation using the adaptive activation function. The middle and bottom rows compare the PINN solution with exact solution using the fixed (middle) and variable $a, n = 10$ (bottom) 'tanh' activation function, respectively after 3600 iterations.

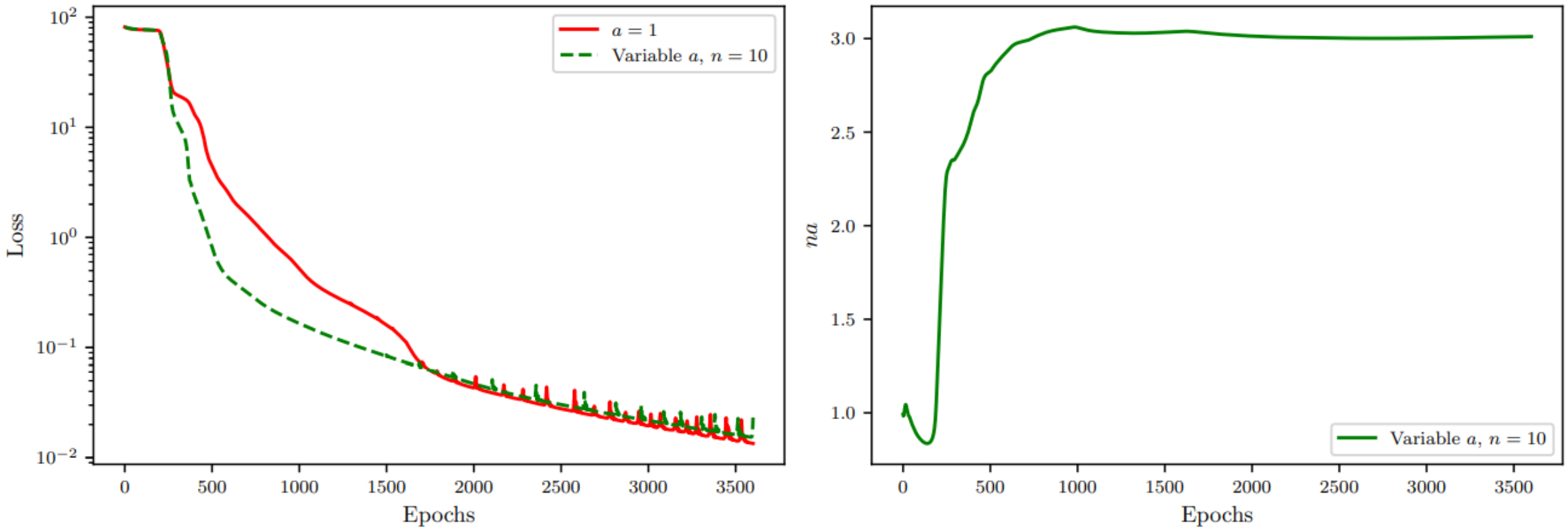



Figure 19: Helmholtz equation: (Left) loss function variation with epochs for fixed ($a = 1$) and (right) adaptive activation ($n = 10$) functions with variation in $a$.

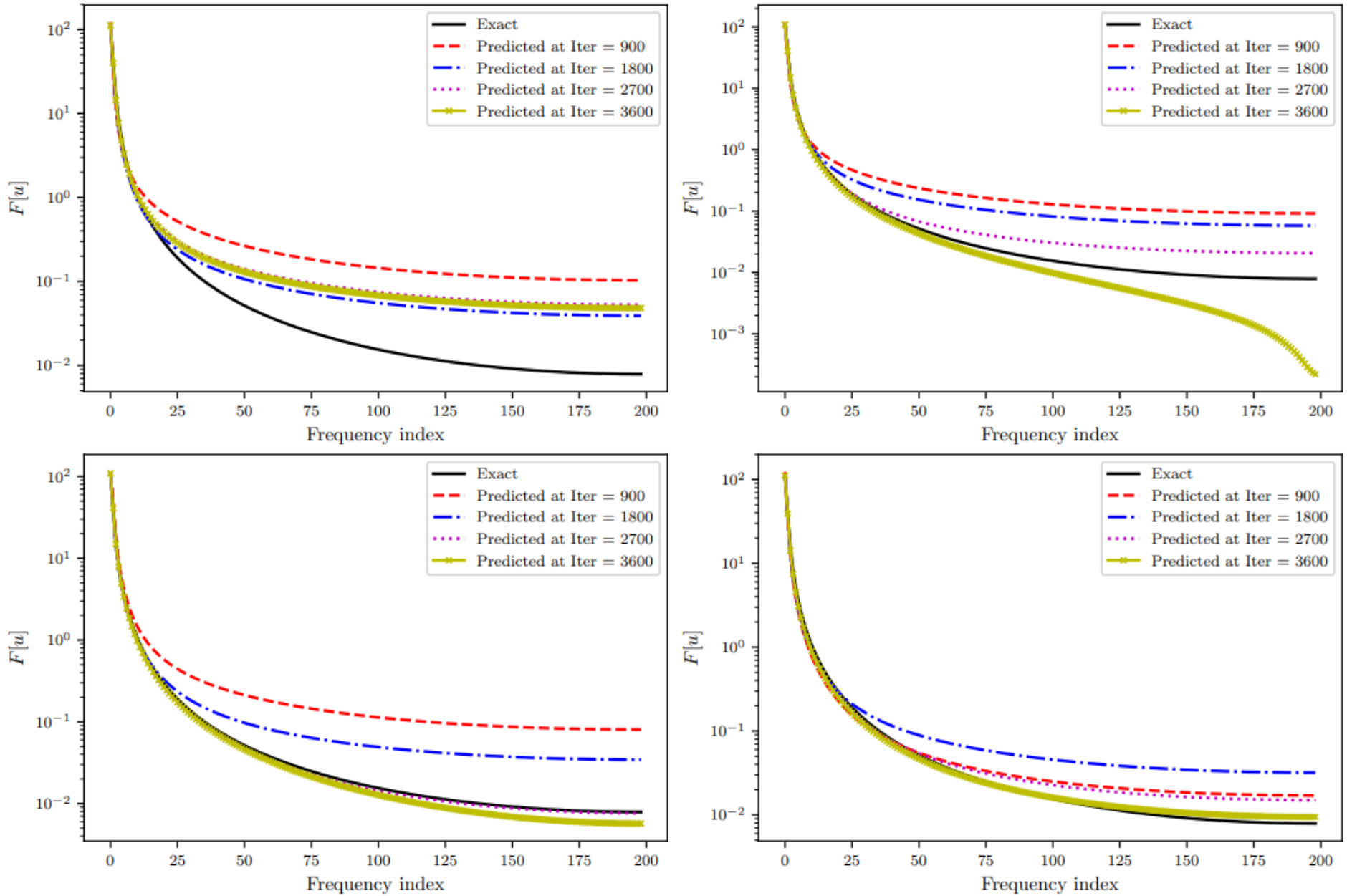



Figure 18: Comparison of solution of Helmholtz equation in frequency domain using fixed (1st row) and variable $a, n = 10$ (2nd row) 'tanh' activation function. First column shows the frequencies in the solution at $x = -0.5$ location whereas second column shows at $x = 0.5$ location.