PINNS Heat Transfer in 2D

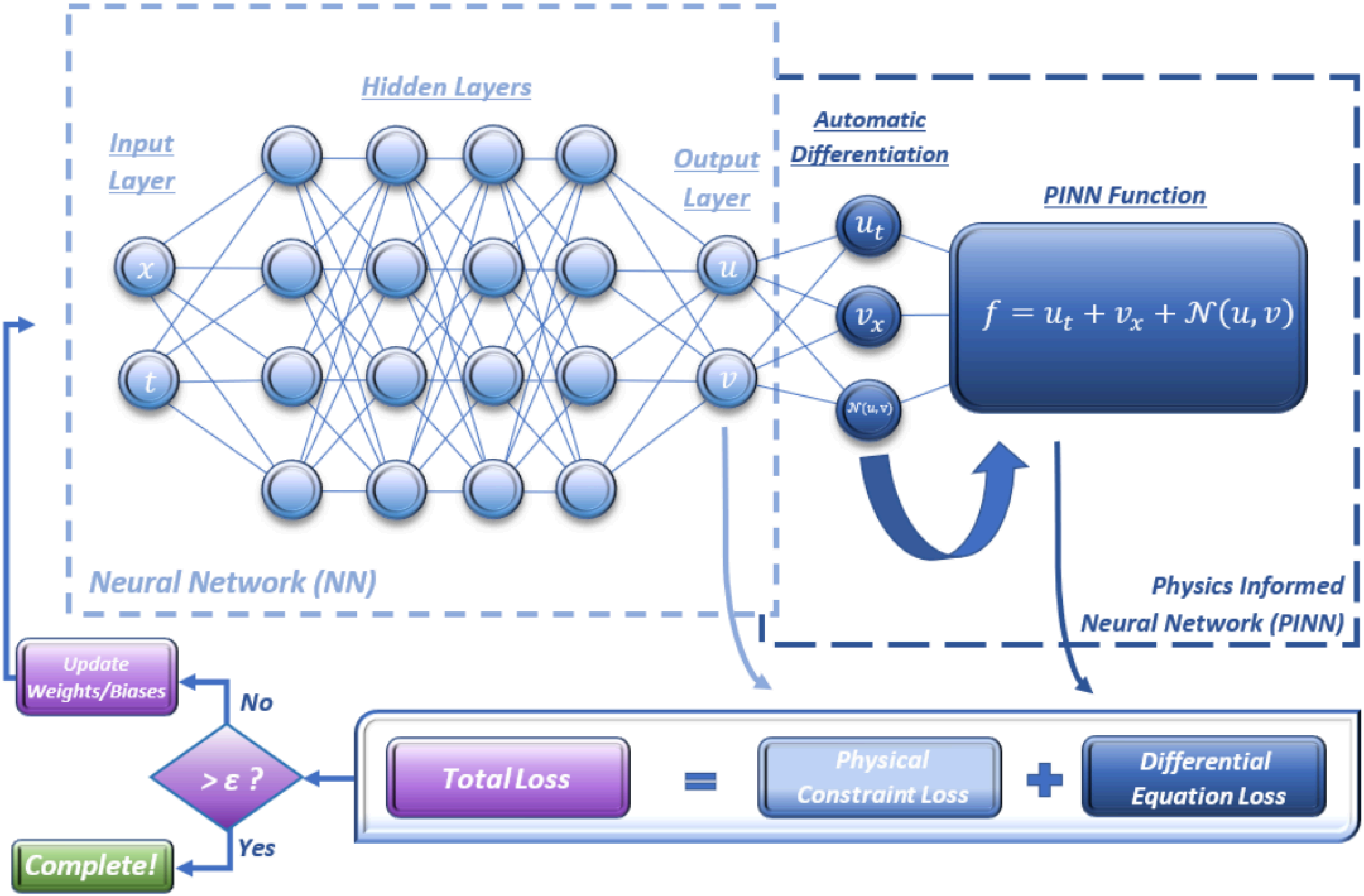
Physics 5300 final project

Content

- Background/motivation for PINNS
- Heat-transfer equation and solution
- PINNS for heat transfer
- Results and discussion

Background/motivation for PINNS?

- Physics informed neural networks (PINNS) is a type of neural network that aims to solve differential equation is a small data regime:
  - Deep learning algorithm aims to accurately identify a nonlinear map from a few – potentially very high-dimensional – input and output data pairs seems at best naive.
  - Encoding structured information (principled physical laws) into a learning algorithm results in amplifying the information content of the data that the algorithm sees, enabling it to quickly steer itself towards the right solution and generalize well even when only a few training examples are available.



Hidden Layers
Input Layer
Output Layer
Automatic Differentiation
PINN Function
x
t
u
v
u_t
v_x
$\mathcal{N}(u,v)$
$f = u_t + v_x + \mathcal{N}(u, v)$
Neural Network (NN)
Physics Informed Neural Network (PINN)
Update Weights/Biases
No
$> \varepsilon$?
Yes
Complete!
Total Loss
=
Physical Constraint Loss
+
Differential Equation Loss

Heat-transfer equation

Heat equation: $\frac{\partial u}{\partial t} = a \cdot \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \quad a = const$

Boundary conditions: $u(x, y, t) = f(x, y, t), \ (x, y) \in \partial\Omega$

Initial conditions: $u(x, y, 0) = g(x, y), \ (x, y) \in \Omega$

Solution to equation: $u(x, y, t)$

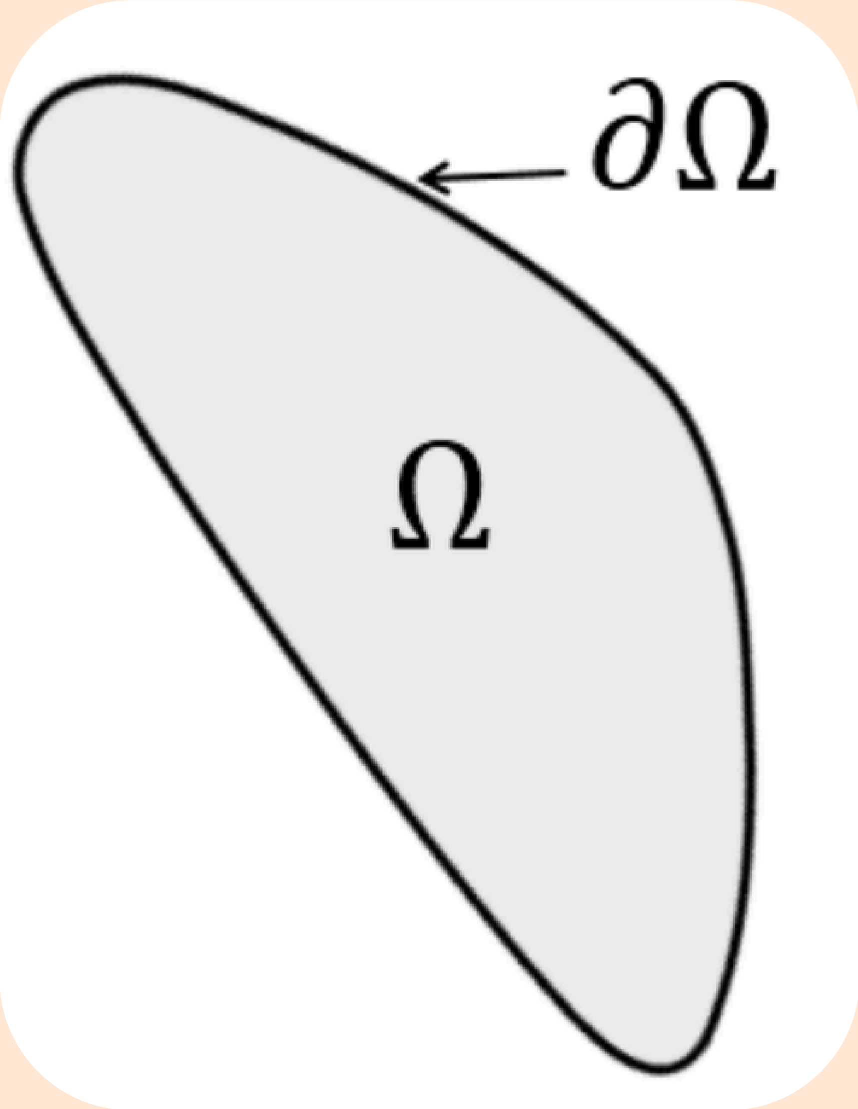
Heat-transfer equation

Heat equation: $$\frac{\partial u}{\partial t} = a \cdot \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \quad a = const$$

Boundary conditions: $u(x, y, t) = f(x, y, t), \ (x, y) \in \partial\Omega$

Initial conditions: $u(x, y, 0) = g(x, y), \ (x, y) \in \Omega$

Solution to equation: $u(x, y, t)$

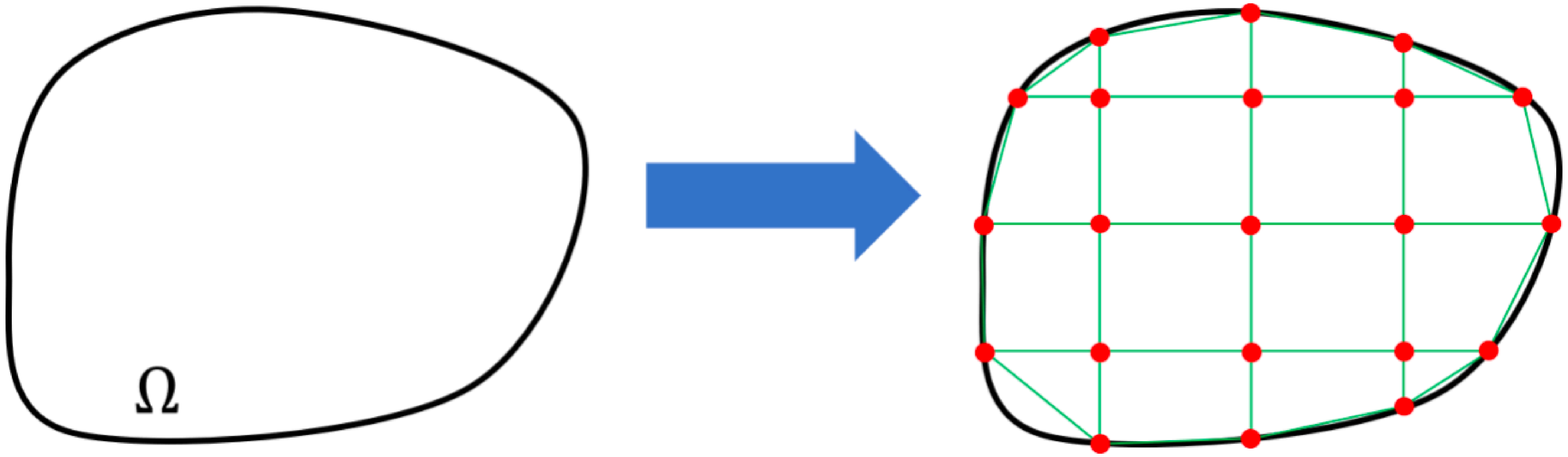


Finite Difference Method

- Discretize the domain and determine the time step
- Approximate (discretize) the partial derivatives

$$u_{i,j}^{k+1} = \alpha \frac{\Delta t}{\Delta x^2}(u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k - 4u_{i,j}^k) + u_{i,j}^k$$

Stability condition: $\Delta t \leq \frac{\Delta x^2}{4\alpha}$



PINNS for heat transfer

Boundary Training sets:

$$\begin{cases} \text{Left bound: } & u(x_{min}, y, t) \\ \text{Right bound: } & u(x_{max}, y, t) \\ \text{Upper bound: } & u(x, y_{max}, t) \\ \text{Lower bound: } & u(x, y_{min}, t) \end{cases}$$

Loss function BC:

$$MSE_u = \frac{1}{N_u} \sum_{i=1}^{N_u} |u(x_u^i, y_u^i, t_u^i) - NN(x_u^i, y_u^i, t_u^i)|^2$$

Solution to equation:

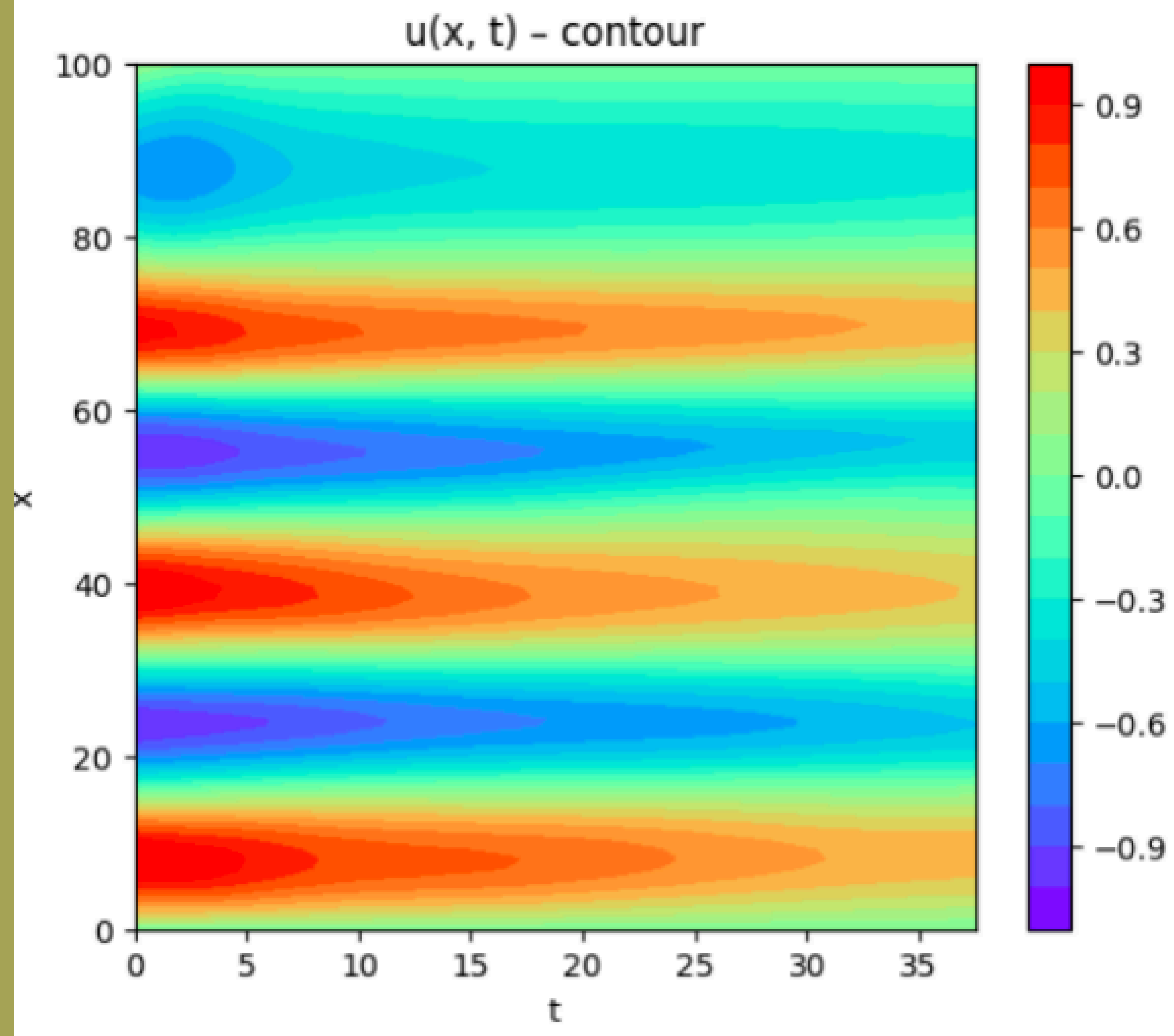
$$f(x, y, t) = \frac{\partial NN}{\partial t} - \alpha \left(\frac{\partial^2 NN}{\partial x^2} + \frac{\partial^2 NN}{\partial y^2}\right)$$

We evaluate our PDE in a certain number of "collocation points" (N_f) inside our domain (x, y, t). Then we iteratively minimize a loss function related to f:

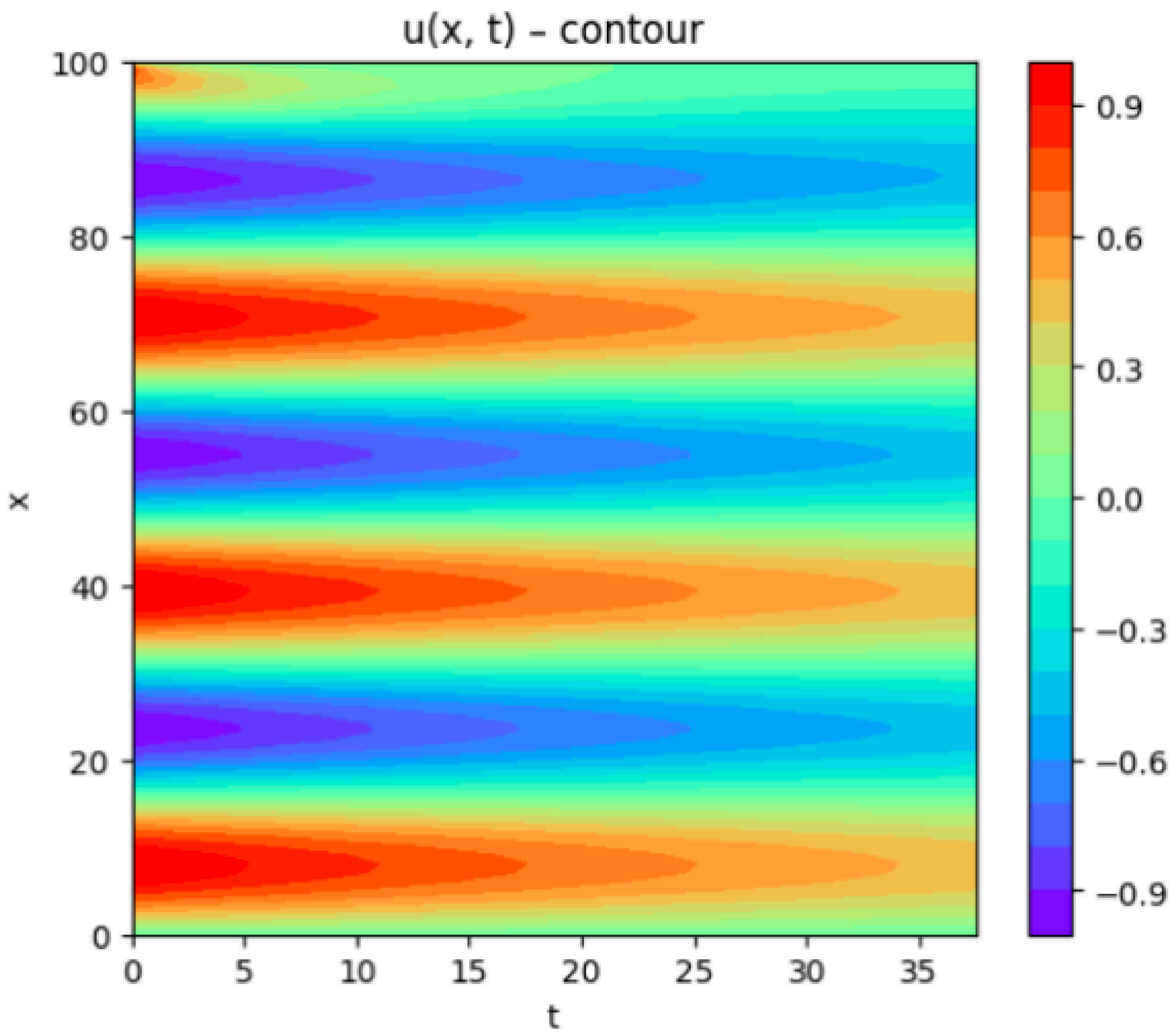
Loss function PDE:

$$MSE_f = \frac{1}{N_f} \sum_{i=1}^{N_f} |f(x_u^i, y_u^i, t_u^i)|^2$$

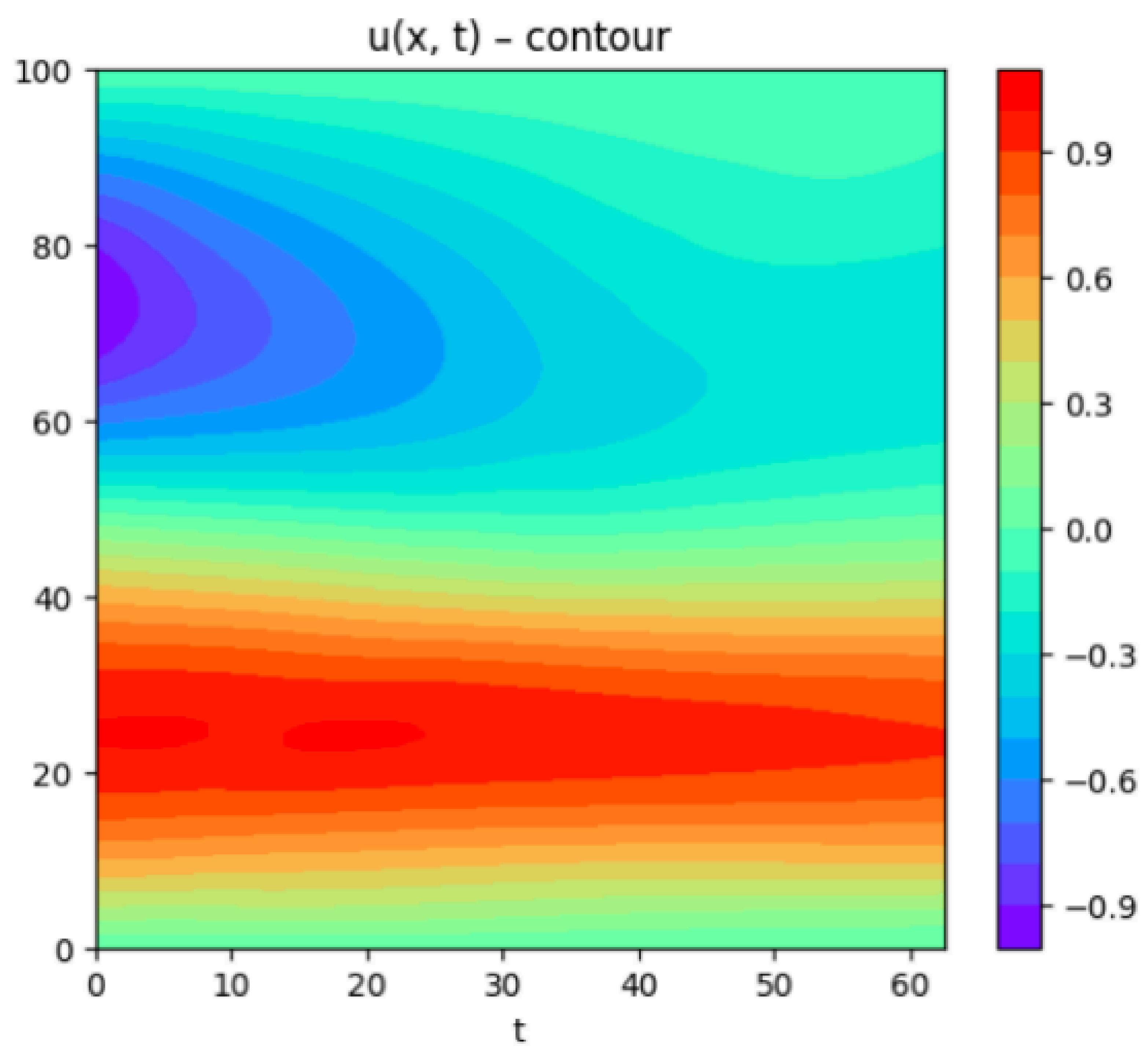
PINNS


u(x, t) – contour
x
t
100
80
60
40
20
0
0
5
10
15
20
25
30
35
0.9
0.6
0.3
0.0
−0.3
−0.6
−0.9

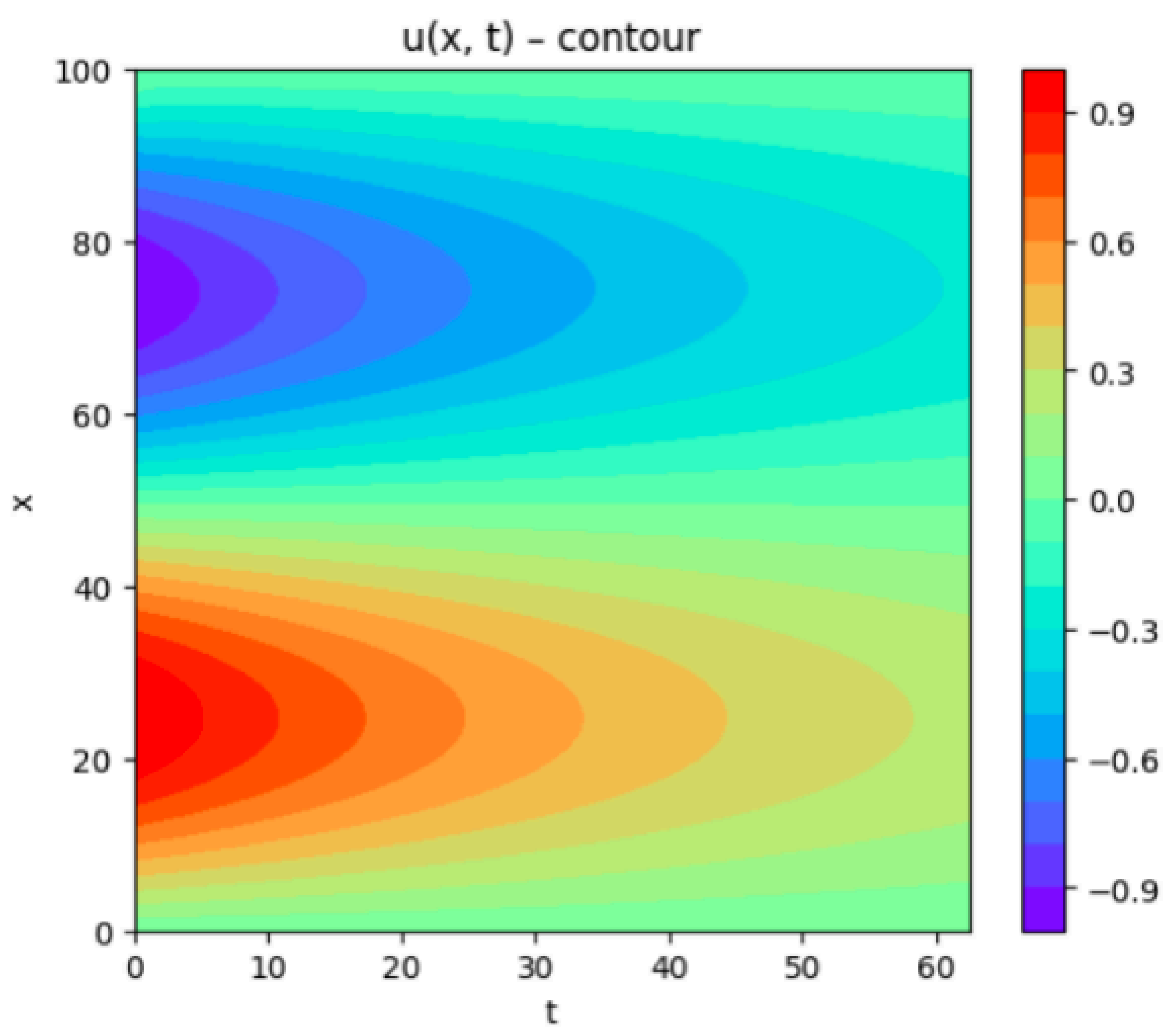
Real


u(x, t) – contour
x
t
100
80
60
40
20
0
0
5
10
15
20
25
30
35
0.9
0.6
0.3
0.0
−0.3
−0.6
−0.9

PINNS


u(x, t) - contour
100
80
60
40
20
0
0
10
20
30
40
50
60
t
0.9
0.6
0.3
0.0
−0.3
−0.6
−0.9

Real


u(x, t) - contour
100
80
60
40
20
0
x
0
10
20
30
40
50
60
t
0.9
0.6
0.3
0.0
−0.3
−0.6
−0.9

Thank You