A novel technique for minimizing energy functional using neural networks

Sanjeeb Poudel[1], **Xiaoqiang Wang**[1], **Sanghyun Lee**[2]

[1] Department of Scientific Computing
[2] Department of Mathematics
Florida State University, Tallahassee, FL

Abstract

An energy functional describes the equilibrium state of a system. In this work, we present a novel technique, Functional Optimization using Neural Networks (FONN), for minimizing the system's energy. FONN utilizes neural networks to process information at discrete grid points, considering their interactions with neighboring grid points, to update the state of the system. The training process involves formulating a loss function based on the system's energy, and with the help of multiple fine-tuning steps, the method employs a progressive energy reduction technique that decreases the energy in multiple steps. FONN's effectiveness is demonstrated across various problems, including the minimization of the heat and Lyapunov energy.

Introduction

0.1 Energy Functional

It takes a function and maps the function to a non-negative real number. It is denoted as

$$E(\phi) = \int_\Omega f\left(\phi(\mathbf{x}), \nabla\phi(\mathbf{x})\right) d\mathbf{x} \geq 0, \qquad (1)$$

where $\mathbf{x} : \Omega \subset \mathbb{R}^d, \phi : \Omega \to \mathbb{R}$.
For example,

$$f(\phi, \nabla\phi) = \frac{1}{2}|\nabla\phi|^2 \qquad (2)$$

We may solve the minimization problem by changing the energy functional into a PDE, i.e.,

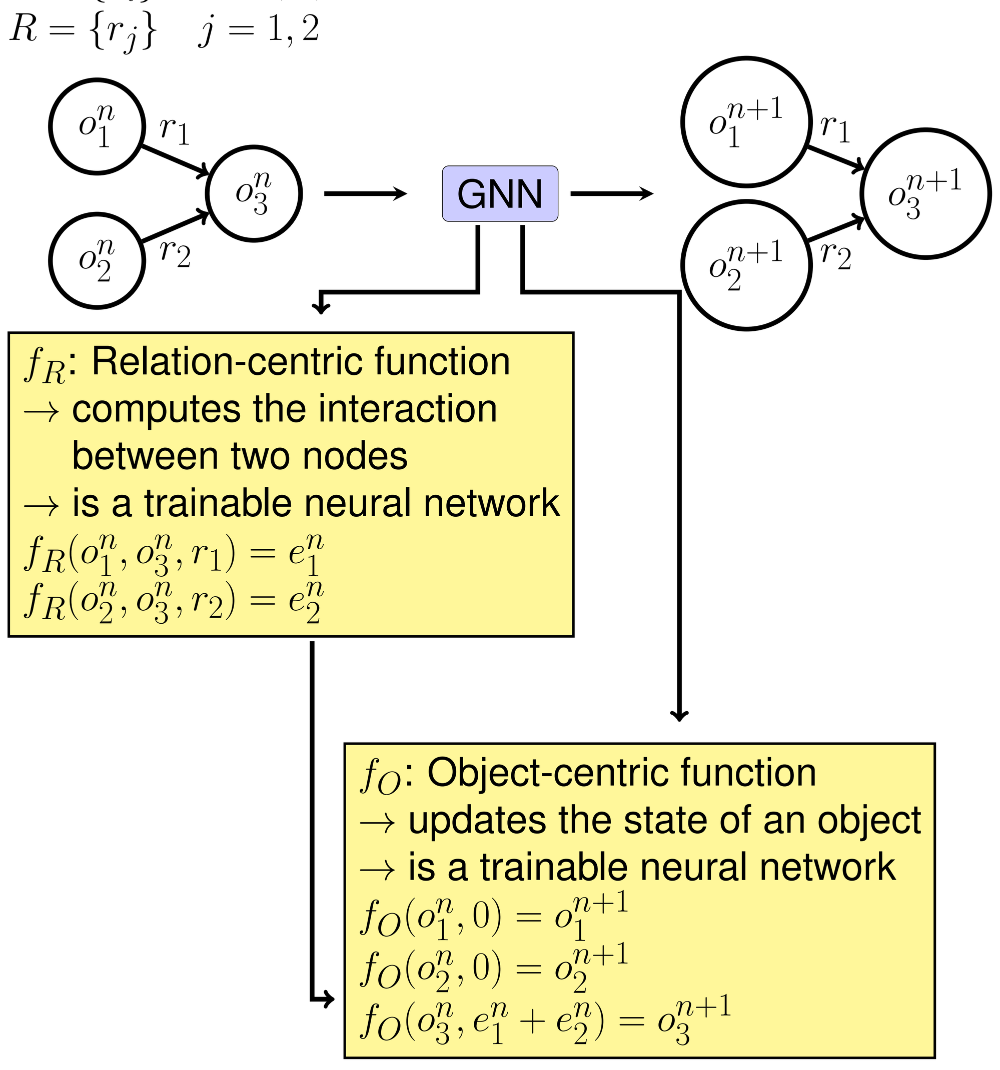
$$\frac{\partial\phi}{\partial t} = \Delta\phi \qquad (3)$$

0.2 Graph Neural Networks

For a graph $G = \langle O, R\rangle$, where O represents the nodes and R represents their relations,

$$G^{n+1} = GNN(G^n), \qquad (4)$$

where n denotes the state of the graph. For example,
$O = \{o_i\} \quad i = 1, 2, 3$
$R = \{r_j\} \quad j = 1, 2$



In general,

- $f_R(o_{a_j}^n, o_{b_j}^n, r_j) = e_j^n, \quad j = 1, 2, \ldots |R|$
  → $o_{a_j}^n$ and $o_{b_j}^n$ are the sender and the receiver of the relation r_j
  → e_j^n is the effect of interaction between the two objects.
- $f_O(o_i^n, \sum_{k\in\mathcal{N}_k} e_k^n) = o_i^{n+1}, \quad i = 1, 2, \ldots |O|$
  → $\mathcal{N}_k$ denotes the total number of the relations for which the node o_i^n is the receiver

Functional Optimization using Neural Networks

0.3 Discretization

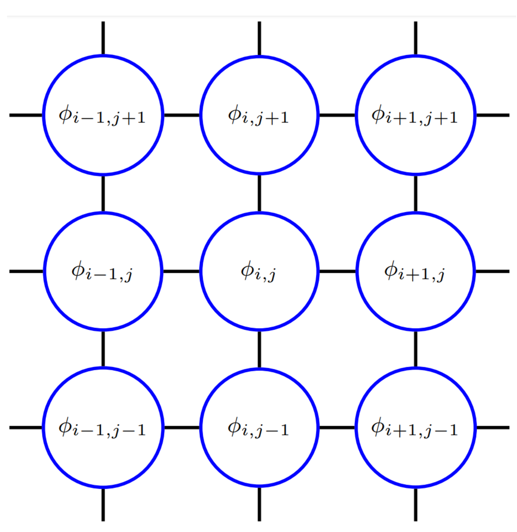


Figura 1: *Regular Grid*

We employ the idea of GNN to the regular grid. The object-centric and the relation-centric function can be formulated as

$$f_R(\phi_{i,j}^n, \phi_{i-1,j}^n, \psi_{i,j}^{m-1}, \psi_{i-1,j}^{m-1}) = e_{i,j,i-1,j}^m, \qquad (5)$$
$$f_R(\phi_{i+1,j}^n, \phi_{i,j}^n, \psi_{i+1,j}^{m-1}, \psi_{i,j}^{m-1}) = e_{i+1,j,i,j}^m, \qquad (6)$$
$$f_R(\phi_{i,j}^n, \phi_{i,j-1}^n, \psi_{i,j}^{m-1}, \psi_{i,j-1}^{m-1}) = e_{i,j,i,j-1}^m, \qquad (7)$$
$$f_R(\phi_{i,j+1}^n, \phi_{i,j}^n, \psi_{i,j+1}^{m-1}, \psi_{i,j}^{m-1}) = e_{i,j+1,i,j}^m. \qquad (8)$$

The combined interacting effect is given by

$$c_{i,j} = e_{i,j,i-1,j}^m - e_{i+1,j,i,j}^m + e_{i,j,i,j-1}^m - e_{i,j+1,i,j}^m. \qquad (9)$$

Finally,

$$f_O(\phi_{i,j}^n, c_{i,j}) = \psi_{i,j}^m. \qquad (10)$$

With $\psi_{i,j}^m$, we can implement a recursive algorithm that allows the information to pass beyond the pairwise interaction.

0.4 Loss Function and Training

The training process involves a loss function defined using the energy functional and the change between the input and the output states.

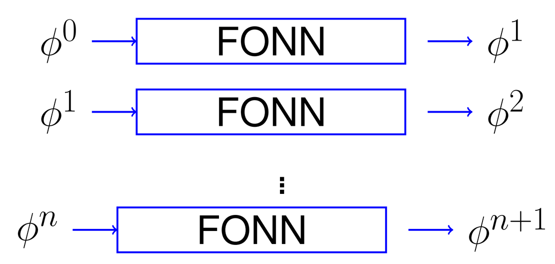
$$\mathcal{L}(\phi^{in}, \phi^{out}) = \left[\frac{1}{N_x N_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}(\phi_{i,j}^{out} - \phi_{i,j}^{in})^2\right] + \omega\frac{E^{out}}{E^{in}}, \qquad (11)$$

→ N_x is the number of grid points along the x-axis
→ N_y is the number of grid points along the y-axis
→ ϕ^{in} and ϕ^{out} are input and output of the network
→ E denotes the energy
→ ω is the weight parameter

Initially, we train the networks f_R and f_O to learn the mapping of the initial state of the system ($\phi^0 \to \phi^0$) using the mean squared error. During this initial training, the Gaussian noise (X) with mean $(\mu) = 0$ and a small standard deviation (σ) is added to the initial state at each iteration, enhancing the robustness of the initial mapping. Once the initial training is complete, the networks are fine-tuned for a few iterations (K) with the given loss function $\mathcal{L}$ (11). This process helps the networks to learn a mapping that not only keeps the output near the input but also achieves a lower energy state. With this fine-tuned network we obtain an updated state of the system. Using this new state, the network is again fine-tuned to get a subsequent state of the system. We continue this iterative process of energy reduction until the termination criterion is met.

- Initial Training $(\omega = 0)$

- Subsequent Training $(\omega \neq 0)$



Numerical Experiments

0.5 Example 1: Heat Equation

Energy Functional:

$$E(\phi) = \frac{1}{2}\int_\Omega |\nabla\phi|^2 d\mathbf{x} \quad \text{on} \quad (0,1)^2$$

Boundary Conditions:

- $\phi(x, 0) = 0$
- $\phi(x, 1) = -x + 1$
- $\phi(0, y) = sin(5\pi y/2)$
- $\phi(1, y) = -4(y - 1/2)^2 + 1$

In the experiment, we have noticed that using a single step for the optimization does not yield satisfactory results.

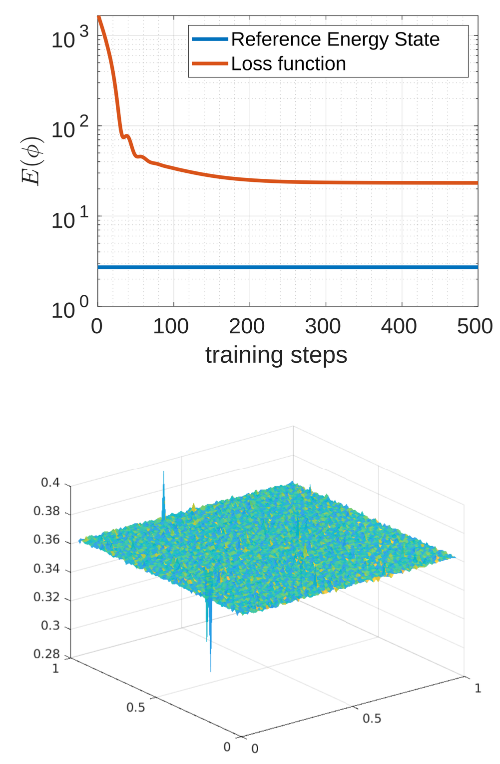


Figura 3: *Optimization of the energy functional with a single step.*

Using our method, the system gets to the minimum of the energy functional. Moreover, it incorporates the boundary condition into the domain.

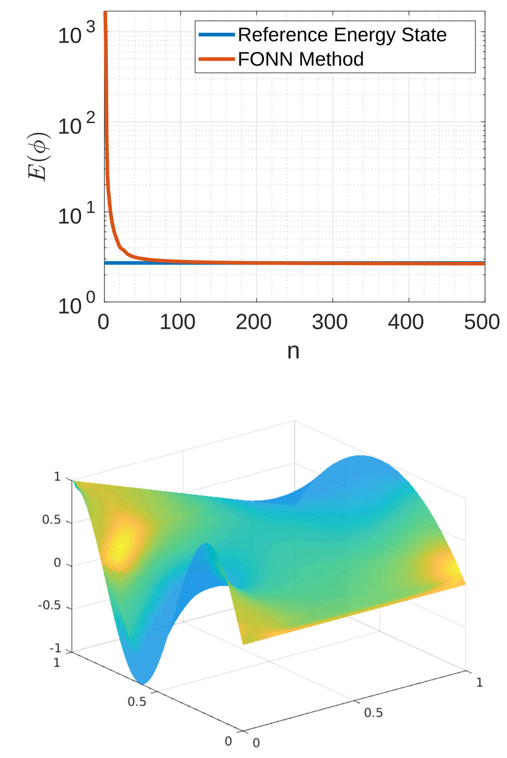


Figura 5: *Optimization of the energy functional using FONN.*

0.6 Example 2: Lyapunov Energy

Energy Functional:

$$E(\phi) = \int_\Omega \left(\frac{1}{2}|\nabla\phi|^2 + \frac{1}{\epsilon^2}F(\phi)\right) d\mathbf{x}, \qquad (12)$$

where $F(\phi) = \frac{1}{4}(\phi^2 - 1)^2$.

Boundary condition: periodic

Initial state: $\sin(4\pi x)cos(4\pi y)$ on $[0,1]^2$

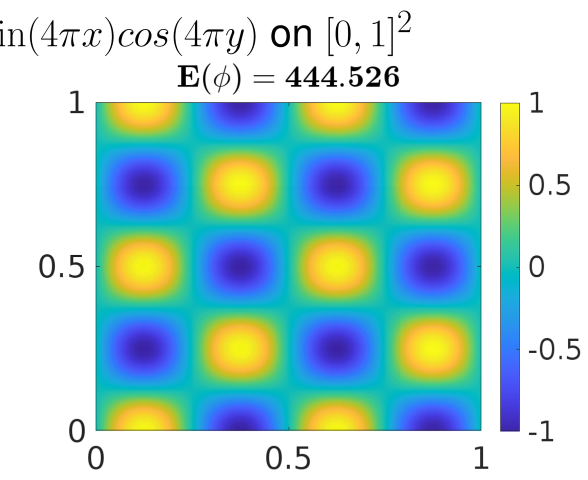


Figura 6: *Initial Condition*

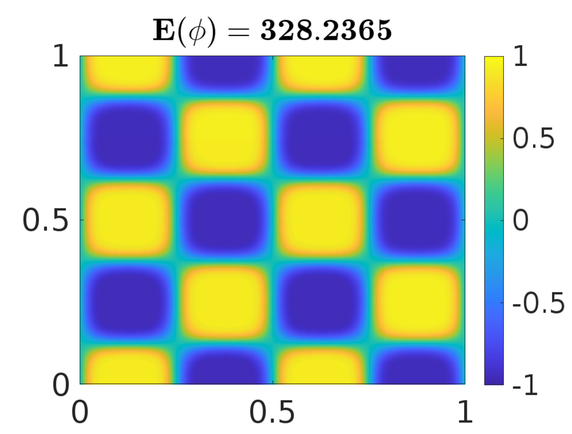


Figura 7: *Minimum of the energy functional*

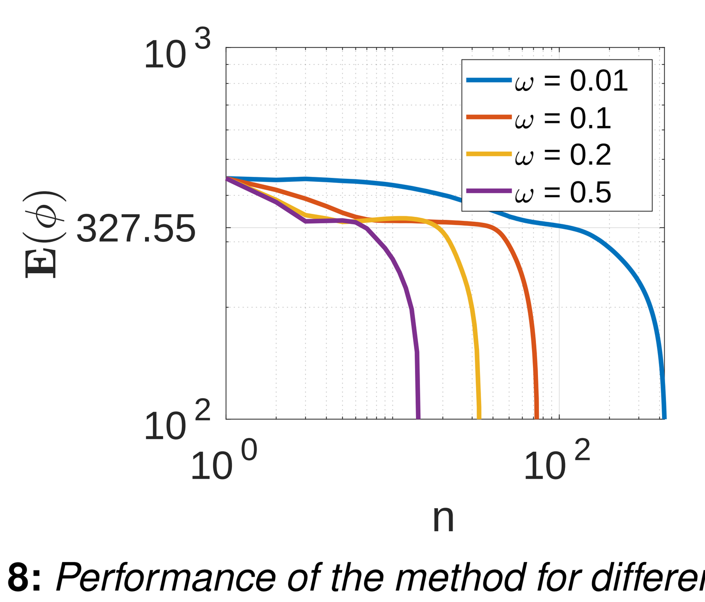


Figura 8: *Performance of the method for different values of the weight parameter, ω*

Conclusion

In this paper, we developed a novel FONN method to minimize energy functional using neural networks. The proposed method is described with a network architecture to work on a regular grid and a loss function based on the energy functional. In addition, a progressive energy reduction technique to decrease energy in multiple steps is established. Several numerical experiments for a wide variety of problems demonstrated the ability of the FONN method for energy minimization.

Reference

Poudel, S., Wang, X., & Lee, S. (2024). A novel technique for minimizing energy functional using neural networks. Engineering Applications of Artificial Intelligence, 133, 108313. https://doi.org/10.1016/j.engappai.2024.108313