# Evolutionary Neural Networks for Option Pricing:

Yang Li [1,2], Zelin Wu [1], Feiyang Ye [1,3]

[1]Southern University of Science and Technology

[2]National University of Singapore

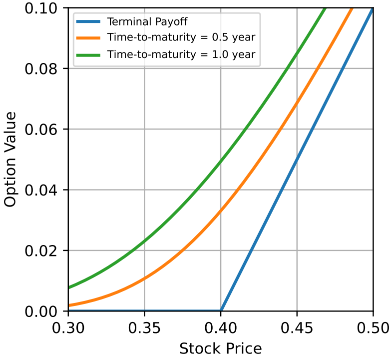
[3]University of Technology Sydney

AAAI 2023 Summer Symposium

$18^{th}$ July 2023

# Table of Contents

# Introduction




**Vanilla European Call Option**

$$\frac{\partial v(\mathbf{s}, t)}{\partial t} + \sigma s \frac{\partial^2 v}{\partial s^2} + rs \frac{\partial v}{\partial s} = rv.$$

**Analytical solution of BS formula** (Black and Scholes 1973):

$$v(s, t) = sN(d_1) - Ke^{-r\tau} N(d_2),$$

$$d_1 = \frac{\log(s/K) + \left(r + 0.5\sigma^2\right)\tau}{\sigma\sqrt{\tau}},$$
$$d_2 = d_1 - \sigma\sqrt{\tau}, \ \tau := T - t.$$

# Introduction

PDE-based option pricing

- Multi-assets version of BS

$$\frac{\partial v(\mathbf{s},t)}{\partial t} + \frac{1}{2}\sum_{i,j=1}^{n} \sigma_i \sigma_j \rho_{ij} s_i s_j \frac{\partial^2 v}{\partial s_i \partial s_j} + r\sum_{i=1}^{n} s_i \frac{\partial v}{\partial s_i} = rv, \tag{1}$$

where $s_i$ is the $i^{th}$ underlying stock price, $v$ is the option value, $r$ represents the risk-free interest rate and $\sigma$ stands for volatility.

- Different payoff functions

$$v(\mathbf{s},T) = v_T(\mathbf{s};K), \tag{2}$$

where T is the time to maturity and K stands for parameters.

# Literature Review: PDE Solvers

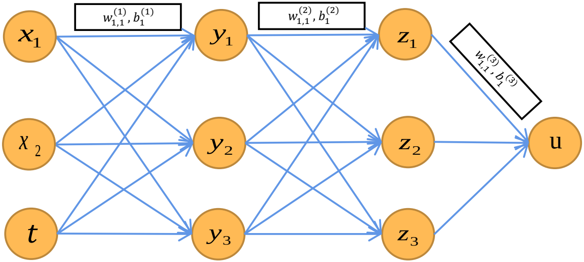
Classical approaches:

- Finite Difference Method(Kim et al. 2020)
- Finite Element Method(Zhang, Zhang, and Song 2015)

NN-based methods:

- DeepBSDE(E, Han, and Jentzen 2017)
- PINNs(Raissi, Perdikaris, and Karniadakis 2019)

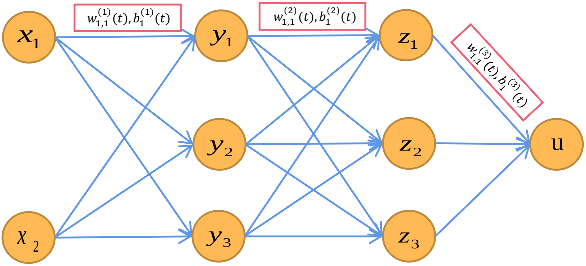
# PINNs for Option Pricing




$$u(x_1, x_2, t) = NN(x_1, x_2, \boxed{t; \theta})$$

$\min_\theta L(\theta)=$ PDE Loss + Boundary Loss + Initial Loss

# Evolutionary Neural Network




$$u(x_1, x_2, t) = NN(x_1, x_2; \theta(t))$$

Control the dynamics: $\mathbf{C}(\theta(t))\dot{\theta} = \mathbf{b}(\theta(t))$

# Advantage of ENN

- **Reliability**: By separating time and space, it effectively reduces PDE to ODE using NNs in a **deterministic** manner.
- **Variability**: Applicable to various PDE-based option pricing problems with different payoff functions.
- **Scalability**: Capable to handle multi-assets option pricing problems.

# Problem Statement

Generally, consider a spatial domain $\mathcal{S} \subseteq \mathcal{R}^d$, and we want to find the solution $v : \mathcal{S} \times [0, T] \to \mathcal{R}$, governed by:

$$\frac{\partial v(\mathbf{s}, t)}{\partial t} = f(t, \mathbf{s}, v, v_{s_i}, v_{s_i s_j}), \quad (\mathbf{s}, t) \in \mathcal{S} \times [0, T], \tag{3}$$

subject to the terminal condition

$$v(\mathbf{s}, T) = v_T(\mathbf{s}; K), \tag{4}$$

where $v_T$ belongs to a function space $\mathcal{V}$.

# Control the Dynamics

Use $V(\mathbf{s}, \theta(t))$ to approximate $v(s, t)$.
Define the residual function

$$r_t(\theta, \eta, \mathbf{s}) := |\nabla_\theta V(\mathbf{s}, \theta(t)) \cdot \eta - f(t, \mathbf{s}, V)|^2, \tag{5}$$

where $\eta$ serves the purpose of $\dot{\theta}(t)$.
To control the dynamics, $\eta$ satisfies:

$$\dot{\theta}(t) \in \arg\min_{\eta} L_t(\theta(t), \eta), \tag{6}$$

where $L_t(\theta, \eta)$ is the objective function defined as:

$$L_t(\theta, \eta) = \int_{\mathcal{S}} r_t(\theta, \eta, s) d\mathbf{s}. \tag{7}$$

# Explicit Solution to the Minimization Problem

Consider the first-order optimization condition:

$$\nabla_\eta L_t(\theta(t), \eta) = 0. \tag{8}$$

The explicit solution is given by:

$$\mathbf{C}(\theta(t))\dot{\theta} = \mathbf{b}(\theta(t)), \quad \theta(T) = \theta_T, \tag{9}$$

where we have:

$$\mathbf{C}(\theta) = \int_{\mathcal{S}} \nabla_\theta V(\mathbf{s}, \theta(t)) \otimes \nabla_\theta V(\mathbf{s}, \theta(t)) d\mathbf{s}, \tag{10}$$

$$\mathbf{b}(\theta) = \int_{\mathcal{S}} \nabla_\theta V(\mathbf{s}, \theta(t)) f(\mathbf{s}, V(\mathbf{s}, \theta(t))) d\mathbf{s}. \tag{11}$$

# Two Tricks in Practice

- We may add $\lambda I$ to ensure the invertibility

$$(\mathbf{C}(\theta(t)) + \lambda I)\dot{\theta}(t) = \mathbf{b}(\theta(t)). \tag{12}$$

- Enforce the Dirichlet boundary conditions
  Suppose the boundary condition is specified as

$$v(s,t)|_{\partial S} = g(s)|_{\partial S}, \quad \forall t \in [0, T]$$

then we can design a suitable function h satisfies $h(s)|_{\partial S} = 0$.
Thus the final output is

$$V(s, \theta(t)) = h(s) * ENN(s, \theta(t)) + g(s).$$

# Algorithm

**Algorithm** ENN-PDE Solver

**Require:** The PDE , $v_T(s_T, K)$ and $\lambda$.

Set $t_k$, $k = 0, \ldots, N$, $t_0 = 0, t_N = T$, $\theta_k = \theta(t_k)$.
Train $\theta_N$.
**for** $k = N - 1$ to 0 **do**
    Calculate the coefficient $\mathbf{C}(\theta_{k+1})$ and $\mathbf{b}(\theta_{k+1})$.
    Update $\theta_k$ explicitly by equation

$$(\mathbf{C}(\theta_{k+1}) + \lambda I)\frac{\theta_{k+1} - \theta_k}{\tau_k} = \mathbf{b}(\theta_{k+1}).$$

**end for**
**return** Discretized solution $V(\mathbf{s}, \theta_k) \approx v(s, t_k), k = 0, ..., N$.

# Vanilla European Call Option

BS with terminal payoff $v(s, T) = (s - K)^+$

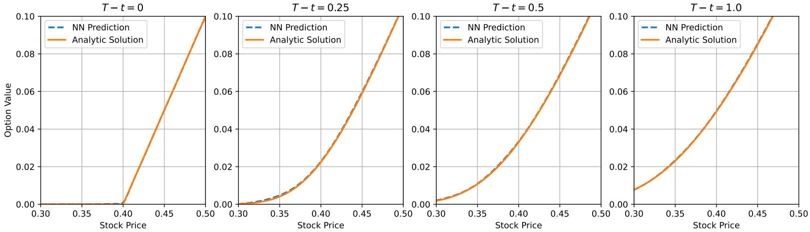



Figure: NN predictions and analytic solutions $\tau = 10^{-3}, \lambda = 10^{-7}$

# Call Option with Transaction Cost

BS with terminal payoff $v(s, T) = (s - K)^+$. (Leland 1985) The volatility is modified as

$$\hat{\sigma}^2 = \sigma^2 \left(1 + \sqrt{\frac{2}{\pi}} \frac{\kappa}{\sigma\sqrt{\delta_t}} sign(v_{ss})\right), \quad (13)$$

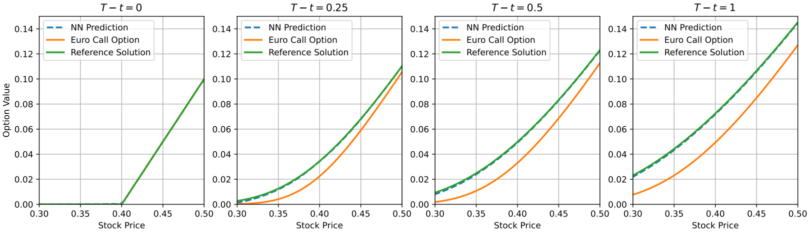



Figure: NN predictions and finite difference method. $\tau = 10^{-3}, \lambda = 10^{-7}$

# Barrier Option: Down-and-out Call

BS with terminal payoff $v(s, T) = (s - K)^+$.
But with an additional condition requires $S_t \geq B, \quad \forall t \leq T$

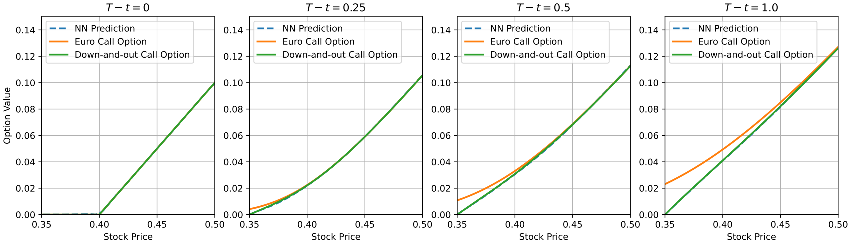



Figure: NN predictions and analytic solutions $\tau = 10^{-3}, \lambda = 10^{-7}$

# Multi-assets option: Exchange Option

BS with terminal payoff $v(s_1, s_2, T) = (s_1 - s_2)^+$.

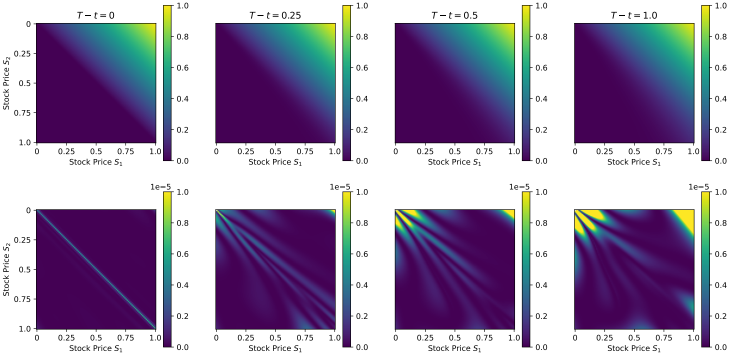



Figure: NN predictions and squared error. $\tau = 10^{-3}, \lambda = 10^{-8}$

# Conclusion

Through our numerical experiments, we demonstrate that our proposed ENN framework can solve **various** PDE-based option pricing problems in a **deterministic** manner effectively and accurately and can be extended to **multi-assets** option pricing problems.

# Reference I

Black, Fischer and Myron Scholes (1973). "The Pricing of Options and Corporate Liabilities". In: *Journal of Political Economy* 81.3, p. 637.

E, Weinan, Jiequn Han, and Arnulf Jentzen (Dec. 2017). "Deep Learning-Based Numerical Methods for High-Dimensional Parabolic Partial Differential Equations and Backward Stochastic Differential Equations". In: *Communications in Mathematics and Statistics* 5.4, pp. 349–380.

Kim, Sangkwon et al. (Mar. 2020). "Finite Difference Method for the Multi-Asset Black–Scholes Equations". In: *Mathematics* 8.3, p. 391.

# Reference II

Leland, Hayne (1985). "Option Pricing and Replication with Transactions Costs". In: *Journal of Finance* 40.5, pp. 1283–1301.

Raissi, M., P. Perdikaris, and G.E. Karniadakis (Feb. 2019). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378, pp. 686–707.

Zhang, Ran, Qi Zhang, and Haiming Song (2015). "An efficient finite element method for pricing American multi-asset put options". In: *Communications in Nonlinear Science and Numerical Simulation* 29.1, pp. 25–36.