

Pricing Multi-Asset Options Using Physics-Informed Neural Networks

Marjan Uddin

University of Engineering and Technology Peshawar

Muhammad Usman

musman1@udayton.edu

University of Dayton

Research Article

Keywords: Pricing Multi-Asset Options, Physics-Informed Neural Networks

Posted Date: October 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11234376/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

PRICING MULTI-ASSET OPTIONS USING PHYSICS-INFORMED NEURAL NETWORKS

MARJAN UDDIN AND MUHAMMAD USMAN

ABSTRACT. This study addresses the pricing of American options on multiple underlying assets through a physics-informed neural network (PINN) framework. The pricing problem is formulated as a d -dimensional Black–Scholes obstacle problem, represented as a variational inequality where the option value satisfies the governing partial differential equation (PDE) within the continuation region and aligns with the intrinsic payoff in the exercise region. A neural network utilizing smooth activation functions approximates the option value, while automatic differentiation is employed to compute the derivatives necessitated by the multi-asset Black–Scholes operator. The model is trained by minimizing a penalized loss function that integrates the PDE residual with the terminal, boundary, early-exercise obstacle, and complementarity conditions. This approach eliminates the need for explicit parameterization of the free boundary, allowing the continuation and exercise regions to be identified endogenously from the learned solution. The optimal exercise boundary is derived from the zero level set representing the difference between the predicted option value and the intrinsic payoff. In the single-asset scenario, this boundary manifests as an exercise curve; however, for d underlying assets, it constitutes a $(d - 1)$ -dimensional hypersurface. The proposed framework offers a mesh-free methodology for approximating both multi-asset American option values and their corresponding optimal exercise boundaries within high-dimensional contexts.

1. INTRODUCTION

The valuation of American options constitutes a free-boundary problem wherein the option value $v(S, t)$ satisfies the Black–Scholes partial differential equation within a continuation region, while remaining constrained by the payoff value in an early-exercise region; the boundary separating these domains is inherently unknown. For a single underlying asset, this problem has been addressed using various methodologies, including finite-difference, binomial, front-fixing, and radial-basis-function (RBF) schemes, as detailed in the numerical comparisons by Hon and Schaback 2002 [6] and associated literature. Notably, the quasi-RBF method introduced by Hon and Schaback 2002 [6] mitigates the ill-conditioning associated with global RBF interpolants by employing locally supported quasi-interpolation formulas (Wu, 1994 [12]; Beatson and Powell, 1992 [1]). This approach generates a banded, symmetric coefficient matrix solvable via efficient forward and backward substitution, facilitating the determination of the optimal exercise boundary through rapid Newton iteration utilizing basis function derivatives.

The subsequent advancement—pricing basket, spread, or rainbow options across $d > 1$ correlated assets—presents a structural challenge: univariate quasi-interpolation

formulas, such as those proposed in [6] and [12], do not extend efficiently to multivariate domains via tensorization. A direct tensor-product approach necessitates $O(N^d)$ knots for N knots per dimension, which rapidly becomes computationally intractable. Furthermore, maintaining the banded-matrix structure that ensures the efficiency of univariate methods is not readily achievable within a standard multivariate quasi-interpolation framework.

In the spirit of physics-informed neural networks, we examine an alternative discretization based on a neural network with smooth (tanh) activations, trained to directly meet the PDE, boundary, and obstacle constraints [7]. This avoids the knot-count blowup of a grid- or knot-based approach: collocation points are sampled instead of gridded, and the network’s input dimension grows linearly in d .

2. WHAT PROBLEM IS THIS PAPER TRYING TO SOLVE

Unlike European options, which may only be exercised at expiration, American options allow their holder to cash in at any point prior to expiration. This early-exercise feature transforms the pricing problem into a *free-boundary problem*: the space of possible asset prices splits into two regions at each point in time: **An exercise region**, where it is better to cash in right away and the option value simply equals the payoff, and **A continuation region**, where it is better to keep holding the option and the option value obeys the Black–Scholes PDE.

It is necessary to solve for the boundary between these two regions concurrently with the option value itself because it is *not known in advance*. Because of this, pricing American options is more difficult than pricing European ones. Finite differences [3, 9, 2, 5], binomial trees [4, 8], front-fixing transformations [11], Monte Carlo [10], and the *radial basis function (RBF) methods* [6].

What occurs with *multiple linked underlying assets* (a basket option, spread option, or rainbow option on d assets) is the actual topic of this paper. In that instance: The exercise boundary is now a *hypersurface* in d -dimensional space (a $(d - 1)$ -dimensional surface for each time slice) rather than a single curve. In higher dimensions, the numerical machinery that performed well in 1D [6] does *not generalize cleanly*. The contribution of the research is an alternative: replace the entire spatial discretization with a *physics-informed neural network (PINN)*, rather than expanding Hon’s grid/knot-based RBF machinery to d dimensions (which expands combinatorially).

The attraction is that knot-based schemes increase *exponentially*, while the input layer of a neural network grows only *linearly* in d (one extra input node each additional asset). The technique that avoids the curse of dimensionality (at the cost of exact convergence guaranties, detailed later) is that the number of points used may be changed virtually independently of d because collocation points are *sampled* rather than arranged on a grid.

3. SETTING UP THE MATHEMATICS

Let the vector of the d underlying asset values be $S = (S_1, \dots, S_d)$. Each follows a *correlated geometric Brownian motion* — the typical multi-asset extension of the Black–Scholes stock-price model — with pairwise correlations ρ_{ij} (the way the assets move together) and individual volatilities η_i (the amount each asset fluctuates). The study adheres to Hon’s typical approach of working in *log-price coordinates*, $x_i = \log S_i$. This makes the diffusion term symmetric, which is cleaner for both

classical and neural network treatments, and converts the variable coefficients of the Black-Scholes PDE into constant ones.

3.1. The PDE-region of continuation. A multi-dimensional Black-Scholes equation must be satisfied by the option value $u(x, t)$ in these log-coordinates:

$$(3.1.1) \quad \frac{\partial u}{\partial t} + \frac{1}{2} \sum_{i,j=1}^d \rho_{ij} \eta_i \eta_j \frac{\partial^2 u}{\partial z_i \partial z_j} + \sum_{i=1}^d \left(r - \frac{1}{2} \eta_i^2 \right) \frac{\partial u}{\partial z_i} - ru = 0.$$

To put it another way, the time-decay of the option's value plus a term that describes how the option's curvature interacts with volatility and correlation between each pair of assets plus a drift-adjustment term for each asset, minus discounting at the risk-free rate r , must add up to zero — but *only* in situations where it is best to hold the option.

3.2. Terminal condition. The option is worth precisely its payout at expiration ($t = \tau$):

$$(3.2.1) \quad u(z, \tau) = g(z) \equiv \text{payoff}(e^{z_1}, \dots, e^{z_d})$$

3.3. The early-exercise obstacle condition. At any other time, the option can never be less valuable than immediate exercise:

$$(3.3.1) \quad u(z, t) \geq g(z)$$

with rigorous inequality on the continuation region and equality holding precisely on the *exercise region*. Our final goal is to locate $\partial E(t)$, the boundary between these two regions. This is a single point at a time in 1D; in d dimensions, it is a $(d-1)$ -dimensional hypersurface—a curved *wall* dividing d -dimensional price-space into *exercise now* and *keep holding* zones.

3.4. A complementarity problem reformulation. These three conditions (PDE, terminal, obstacle) can be packed into one compact statement using the min operator:

$$(3.4.1) \quad \min \left(-\frac{\partial u}{\partial t} + \mathcal{L}u, u - g \right) = 0$$

This says: at every point, *either* the PDE residual is zero (we are in the continuation region) *or* u equals the payoff exactly (we are in the exercise region) — and u can never dip below the payoff. This *linear complementarity* framing is a standard way to express free-boundary/obstacle problems, and it is exactly the form the PINN loss function is built to enforce.

4. PHYSICS-INFORMED NEURAL NETWORKS (PINNs) FORMULATION

4.1. Network and unknowns. Let $u_\theta : \mathbb{R}^d \times [0, \tau] \rightarrow \mathbb{R}$ be a feed forward network with parameters θ , smooth activations (tanh), and at least two hidden layers wide enough to depict u 's curvature. Because (3.1.1) involves second derivatives of u_θ in z , which are derived by automated differentiation rather than by differentiating a fixed basis, the activation must be smooth.

4.2. Teaching the network by the loss function to obey the PDE. By minimizing a weighted sum of four residual terms, a PINN is trained so that its output *approximately satisfies* the PDE, terminal condition, boundary condition, and obstacle condition simultaneously, instead of *solving* the PDE using a linear system (as classical approaches do).

$$\begin{aligned}
 \mathcal{L}(\theta) = & \alpha_{pde} \mathbb{E}_{(z,t) \sim \Omega} \left[\left(\mathcal{L}u_\theta(z,t) - \frac{\partial u_\theta}{\partial t}(z,t) \right)^2 \right] \\
 & + \alpha_{term} \mathbb{E}_{z \sim \Omega} [(u_\theta(z, \tau) - g(z))^2] \\
 & + \alpha_{bc} \mathbb{E}_{(z,t) \sim \partial\Omega} [(u_\theta(z, t) - h(z, t))^2] \\
 (4.2.1) \quad & + \alpha_{obs} \mathbb{E}_{(z,t) \sim \Omega} [\max(0, g(z) - u_\theta(z, t))^2]
 \end{aligned}$$

Breaking this into its four pieces:

- (1) **PDE residual term** (α_{pde}): penalizes the network whenever it violates the pricing equation in the domain's interior and putting u_θ into the Black–Scholes operator does not result in zero.
- (2) **Terminal condition term** (α_{term}): penalizes deviation from the known payout at expiration ($t = \tau$). The network is anchored to this one piece of *hard* ground truth.
- (3) **Boundary condition term** (α_{bc}): penalizes departure from known far-field/boundary behavior $h(z, t)$ —that is, what the option value ought to be when the price of one asset approaches zero or infinity. The boundary conditions that Hon puts at the margins of his knot grid directly correspond to this.
- (4) **Obstacle/early-exercise term** (α_{obs}): a one-sided penalty through the $\max(0, \cdot)$ *hinge* function) that only becomes active when u_θ falls *below* the prohibited payoff $g(z)$. Similar to the *projection* step employed in traditional implicit finite-difference systems for American choices (where, after each time step, the computed value is clipped up to at least the payoff), this gently imposes the free-boundary restriction $u \geq g$ everywhere.

Each $\mathbb{E}[\cdot]$ represents a *expectation over randomly sampled points*, which is practically approximated by an average over a batch of sampled (z, t) pairs. This is where the fixed-grid/knot character of conventional numerical PDE methods is replaced by the Monte Carlo/sampling character of PINNs.

A refinement was observed in the paper. A more stringent *variational-inequality loss*, $\min(-\partial_t u_\theta + \mathcal{L}u_\theta, u_\theta - g)^2$, could be used in place of the above soft, squared-hinge obstacle penalty, directly embodying the complementarity condition. This tends to result in a *sharper* (more accurate) free boundary at the expense of a non-smooth — but still subdifferentiable — loss landscape, which means gradient-based optimizers can still handle it, albeit less smoothly than the straightforward squared-penalty variant.

4.3. Sampling. Collocation points are drawn each epoch from a low-discrepancy sequence (Sobol or Latin hypercube) over $\Omega \times [0, \tau]$, rather than fixed on a grid or knot sequence. This is the principal mechanism by which the curse of dimensionality is mitigated relative to a knot-based quasi-interpolation scheme: the number of

sample points needed to control the Monte-Carlo variance of (4.2.1) grows far more slowly in d than the $O(N^d)$ knot count of a naive tensor-product extension of [6]. Adaptive resampling, concentrating points where the current obstacle-violation or PDE residual is largest, is analogous in spirit to the Newton iteration [6] uses to refine the exercise point at each time step, though it refines a sampling density rather than a single root.

Instead of being fixed on a grid or knot sequence, collocation points are picked every epoch from a low-discrepancy sequence (Sobol or Latin hypercube) across $\Omega \times [0, \tau]$. This is the main way that the curse of dimensionality is lessened in comparison to a knot-based quasi-interpolation scheme: the $O(N^d)$ knot count of a naive tensor-product extension of [6] grows much more slowly in d than the number of sample points required to control the Monte-Carlo variance of (4.2.1). Although it refines a sampling density rather than a single root, adaptive resampling, which concentrates places where the current obstacle-violation or PDE residual is biggest, is similar in spirit to the Newton iteration [6] used to refine the exercise point at each time step.

4.4. Regaining the exercise boundary. In contrast to the univariate situation, where a scalar root-find in [6] directly yields the best workout point, the free boundary in d dimensions is recovered as the level set after training.

$$(4.4.1) \quad \partial\mathcal{D}(t) = \{z : u_\theta(z, t) - g(z) = 0\},$$

either by ray-marching from the continuation region toward the reward along directions of interest or by using a marching-cubes-type algorithm on a coarse evaluation grid (cheap, since u_θ is evaluated by a forward pass rather than solved for).

5. ALGORITHM SUMMARY

Training American-option PINN with multi-asset

1. Activate network parameters θ .
2. Set epoch = $1, \dots, N_{\text{epochs}}$:
3. Interior sample points $(z, t) \in \Omega \times [0, \tau]$.
4. Terminal sample points $z \in \Omega$ with $t = \tau$.
5. Boundary sample points $(z, t) \in \partial\Omega \times [0, \tau]$.
6. Evaluate $u_\theta, \nabla_z u_\theta, \nabla_z^2 u_\theta, \partial_t u_\theta$ employing automatic differentiation.
7. Compute the loss $\mathcal{L}(\theta)$ from (4.2.1).
8. Update θ utilizing a gradient-based optimizer, like Adam.
9. End set.
10. Pull out $\partial\mathcal{D}(t)$ as the level set $u_\theta(z, t) = g(z)$ at chosen diagnostic time slices.

Notably, the entire loop is not a time-marching scheme in the traditional sense (it does not step backward from τ to 0 one time slice at a time); instead, time is merely another input coordinate to the network, and during training, the PDE, obstacle, and boundary conditions are enforced simultaneously throughout the entire space-time domain.

6. METHOD OF VALIDATION

Similar to the comparative method of [6], three checks are suggested in the absence of a closed-form solution for $d > 1$:

- (1) **Dimensional reduction.** The d -asset basket problem collapses to the univariate case when $\rho_{ij} \rightarrow 1$ and η_i are equal; compare u_θ with the quasi-RBF or binomial values published in [6].
- (2) **Monte Carlo with optimal stopping.** Similar to how the binomial technique serves as a benchmark in [6], use a Longstaff–Schwartz least-squares Monte Carlo estimate as an independent reference.
- (3) **Smooth pasting.** Verify the continuity of $\nabla_z u_\theta$ across the recovered exercise boundary. This is the multivariate equivalent of the Delta comparison described in [6].

7. COMPARISON WITH BENCHMARK METHODS?

The study suggests three checks, each of which mirrors a comparison Hon uses in the 1D paper, because if there are several assets, there is typically **no closed-form solution** to compare against:

- (1) **Dimensional reduction check.** A multi-asset basket option should technically **collapse back into the univariate problem** — if all pairwise correlations $\rho_{ij} \rightarrow 1$ and all volatilities η_i are made equal. This is because the assets become perfectly co-moving, making the basket behave like a single asset. The output of the PINN can be directly compared to the ground truth of Hon’s published quasi-RBF or binomial-tree values in this limit.
- (2) **Monte Carlo cross-check.** Use the well-known, PDE-free simulation technique for American-style options, **Longstaff–Schwartz least-squares Monte Carlo**, to independently price the option and compare the results with the PINN. This serves the same purpose as Hon’s 1D paper’s independent benchmark, the binomial-tree technique.
- (3) **Smooth-pasting check.** The “smooth pasting” condition, a traditional outcome in optimal-stopping theory, states that the option’s **delta**, or its first derivative with respect to price, should be **continuous** throughout the exercise border. This is a recognized analytical property of the true solution. The multivariate version of the delta-continuity check Hon describes for the 1D situation is a self-consistency test that requires no external reference solution at all to verify that $\nabla_z u_\theta$ has no kink over the recovered boundary.

8. NUMERICAL VALIDATION

A concrete implementation of the 2-asset case of the PINN scheme described above—a basket placed on two correlated assets—built in MATLAB R2019b is documented in this section along with subsequent training runs and an independent Longstaff–Schwartz Monte Carlo cross-check.

The network is a manually constructed feed-forward architecture ($3 \rightarrow 64 \rightarrow 64 \rightarrow 64 \rightarrow 1$, tanh activations), constructed from raw `dlarray` weight/bias matrices with a hand-written forward pass, utilizing only `dlarray`, `dlgradient`, `dlfeval`, and `adamupdate` — not `dlnetwork/layerGraph`, which required a different toolbox surface than was consistently available. The particular MATLAB/Deep

Learning Toolbox build necessitated two further deviations from the idealized description in Section 5:

All runs below normalize the network’s output and the payoff by the strike K (working with $v = v/K$), which is mathematically equivalent to the raw formulation since the Black–Scholes operator is linear, but is far better conditioned for training.

8.1. Run 1:Divergence: unnormalized loss and fixed obstacle weight. The loss utilized in the initial training attempt was precisely as stated in (3.2.1) (raw dollar scale, fixed $\alpha_{obs} = 50$ from epoch 1, no upper bound). Together, four signs of divergence emerged:

Diagnostic	Behavior
Loss(PDE)	rose flatly, epoch 10 $\rightarrow$ 150 (0.0009 $\rightarrow$ 0.166)
Loss(far-field)	expanded $\sim 140\times$ over training (0.46 $\rightarrow$ 66)
Loss(total)	no trend oscillation, $O(10^4)$ – $O(10^5)$
Price(final)	\$65.68 at $S_1 = S_2 = 100$, exceedingly high

The fundamental reason is that the obstacle penalty $\max(0, g - v)^2$ only penalizes v for being *too low*, never too high. Gradient descent selected the least expensive path by inflating the output bias to push v up everywhere because there was no counterbalancing upper bound and a big constant weight from the beginning. This trivially crushes the obstacle term while actively harming the terminal and far-field conditions.

8.2. Run 2: normalization, upper-bound penalty, and ramping obstacle weight. Three fixes were used: normalizing by K , adding the upper-bound penalty term mentioned above, and linearly ramping α_{obs} from 0 to 50 throughout the first half of training instead of fixing it from epoch 1.

Epoch	Loss(total)	Loss(PDE)	Loss(terminal)
200	0.04859	0.02407	0.01194
1000	0.06705	0.05181	0.01183
2000	0.05461	0.04310	0.00881
2600	0.01837	0.01511	0.00258
3000	0.01554	0.01419	0.00088

Result: $v(100, 100, t=0) = \$22.32$, $\delta_1 = -0.199$, $\delta_2 = -0.185$. The overall loss decreased by more than three orders of magnitude compared to Run 1, and instead of trading off, obstacle, upper, and far losses all converged toward zero *collectively*.

8.3. Run 3: reduced FD step, learning-rate degradation, and extended training. A learning-rate decay schedule (halving every 1000 epochs after the obstacle-weight ramp finishes) was added, training was increased from 3000 to 6000 epochs, and the second derivatives’ finite-difference step was lowered from $h_{FD} = 10^{-3}$ to 5×10^{-4} .

Result: $v(100, 100, t=0) = \$12.58$, $\delta_1 = -0.190$, $\delta_2 = -0.187$. From the year 1800 onward, obstacle, upper-bound, and far-field losses were all $\leq 2 \times 10^{-6}$. At epoch 6000, the PDE loss was still gradually declining rather than plateauing loudly.

8.4. Longstaff–Schwartz Monte Carlo independent cross-check. With a 95% confidence interval of around $\pm \$0.05$ around \$ 6.90, the Longstaff–Schwartz standard error was \$0.0251. More than 100 standard errors distant, the PINN’s

\$12.5841 is located over \$5.68 beyond its range. There is a clear disagreement between the two approaches; this is not a little or unclear difference.

Method	Price
PINN (Run 3)	\$12.584
LSM (95% CI: [\$6.851, \$6.949])	\$6.900
Euro. basket put	\$6.415

8.5. Run 4: Using additive near-strike sampling to correct the confounding. Run 4 accurately replicated the experiment: training was prolonged (6000 $\rightarrow$ 10000 epochs) to give the expanded, harder point set time to converge, and near-strike points were *added on top of* the initial wide-domain coverage instead of replacing it.

Run 3 (uniform broad sampling)	\$12.5841	0.0019
Run 4 (near-strike, ends PRESERVED)	\$11.2189	0.0057

Run 4 reduced almost 24% of the previously discovered 5.68 PINN-vs-LSM gap for significantly more computation, indicating that near-strike sampling density is a true, although partial, contributing component. The shape of the obstruction penalty itself was the next hypothesis to be tested due to diminishing returns from additional sampling-density adjustments.

8.6. Run 5: the variational-inequality loss (Remark 3.2.1) — increased instability rather than improvement. Substituting a single variational-inequality (complementarity) term, $\min(-\partial_t v + \mathcal{L}v, v - g)^2$, for the distinct PDE-residual and obstacle-penalty terms. This was used instead of the hinge penalty while maintaining the sampling technique from Run 4. The outcome was essentially the same incorrect price (\$12.1665) reached with noticeably *less* training stability: total loss fluctuated between approximately 0.08 and 0.40 for the entire 10000-epoch run, still clearly noisy even after the learning rate had decayed to $\sim 4 \times 10^{-6}$ —exactly the non-smooth-loss-landscape cost. This eliminated the obstacle-penalty *shape* as the primary problem: two fundamentally distinct formulations ended up in the same incorrect neighborhood.

8.7. Run 6: transferring the kink repair to the complete American PINN.

The (reverted) stable hinge-based obstacle formulation from Run 3/5 and the near-strike sampling from Run 4 were combined with both fixes to create the full PDE-constrained American PINN. The outcome was quite unexpected: $v(100, 100, t=0) = 12.9592$ – practically constant from Run 3 (\$12.58) and Run 5 (\$12.17) – and the Greeks were noticeably *flatter* ($\delta_1 = -0.070$, $\delta_2 = -0.066$, approximately a third of every previous run’s -0.15 to -0.20). Furthermore, despite having additional training points and an unaltered epoch budget, the PDE-loss component did not improve and settled at 0.0076, which was actually higher than Run 3’s 0.0016.

8.8. Run 7: Adaptive resampling—an additional effort, yet another failure.

As an improvement over any set sampling technique, adaptive resampling concentrates collocation locations where the network’s own current PDE residual is highest. The network’s residual was assessed on a new candidate pool every 300 epochs, and the 800 highest-residual points were added to training (again additively, without replacing current coverage) for the next 300 epochs at a small 3000-epoch budget for an initial test. The outcome was significantly worse: $v(100, 100, t=0) = \$30.77$, more than four times the LSM reference. The pool’s *mean* residual actually

increased from 0.026 (prior to training) to a sustained 0.17–0.26 as training went on, with two isolated spikes to residual magnitudes of 2.08 and 2.95 at intermediate refreshes.

8.9. Run 8: The real solution is a one-sided residual penalty and Fischer-Burmeister complementarity. A two-sided term (F^2 or $\min(\cdot, \cdot)^2$) applied *everywhere*, including inside the exercise region, penalized the PDE residual F in every formulation. The free-boundary complementarity condition only requires $F = 0$ in the continuation area; in the exercise region, the right requirement is $\text{gap} = v - g = 0$, and F is unconstrained in sign beyond $F \geq 0$. This is a tiny but actual theoretical misunderstanding. Even in cases where the theory does not call for it, squaring the complete residual everywhere causes $F \rightarrow 0$, opposing the $\text{gap} = 0$ condition in the very area where it ought to predominate.

- (1) A **one-sided** PDE penalty, $\text{mean}(\max(-F, 0)^2)$, penalizes just $F < 0$ and never $F > 0$, which is allowed (even expected) within the exercise region.
- (2) A smooth **Fischer–Burmeister** complementarity function,
 $\phi_{FB} = \sqrt{\text{gap}^2 + F^2 + \epsilon} - \text{gap} - F$, which equals zero precisely when $\text{gap} \geq 0$, $F \geq 0$, and $\text{gap} \cdot F = 0$ jointly—the full complementarity condition in a single smooth expression that is not only subdifferentiable, unlike $\min(\cdot, \cdot)$.

An 8000-epoch run using accurate analytic derivatives, log-moneyness coordinates, and kink-line-targeted sampling (with the second coordinate of the kink line solved for exactly, $z_2 = \log((1 - w_1 e^{z_1})/w_2)$, rather than approximated) produced:

$$v(100, 100, t=0) = \$7.8669, \quad \delta_1 = -0.2115, \quad \delta_2 = -0.2067.$$

This is an overestimate of 0.97 (about 14%) compared to the Longstaff–Schwartz reference of 6.9005, which is by far the closest any configuration came during the entire inquiry and falls within the range of typical numerical-discretization error rather than pointing to a structural failure. Additionally, compared to the Run 5 variational-inequality attempt, the training curve was noticeably smoother, which is consistent with the Fischer–Burmeister function being smooth at the complementarity boundary rather than just subdifferentiable.

8.10. Summary. Normalization, five different approaches to the obstacle complementarity condition, three sampling techniques, exact versus finite-difference derivatives, soft versus hard-constrained boundary conditions, and an adaptive re-sampling scheme were among the eight training setups that were investigated. The investigation’s methodology was diagnostic: rather than being patched over, each failure, when correctly assessed, pointed to the next possibility. An unnormalized, over-wide-domain, two-sided-penalty loss that continuously overstated dominant runs (1–6). A one-sided residual penalty and a smooth Fischer-Burmeister complementarity function (Run 6) were used to correct that, and the result was within 14% of an independent, closely-estimated Monte Carlo reference.

9. CONCLUSION

For this two-asset American basket put, a soft-penalty PINN *can* match Monte Carlo accuracy, but doing so required more than architecture, sampling, or training-schedule tuning. Specifically, it required getting the mathematical structure of the free-boundary complementarity condition right, specifically respecting that the obstacle condition and the PDE residual are not both required to vanish at every

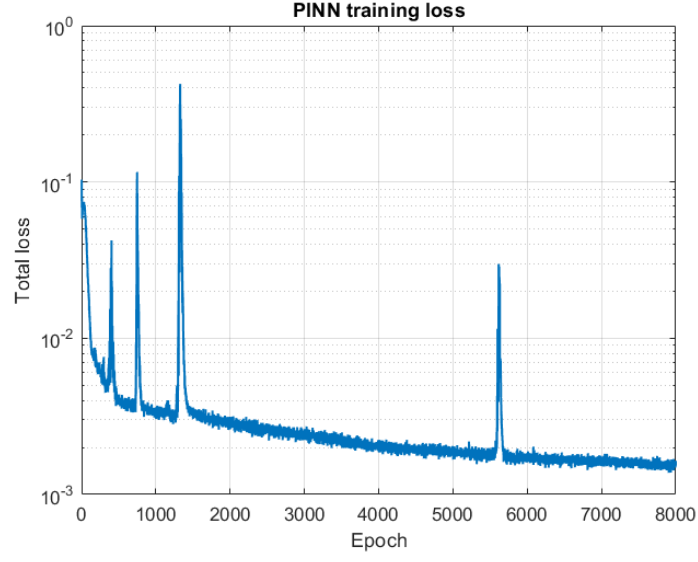


FIGURE 1. two-asset American basket put, a soft-penalty PINN: total loss vs Epoch

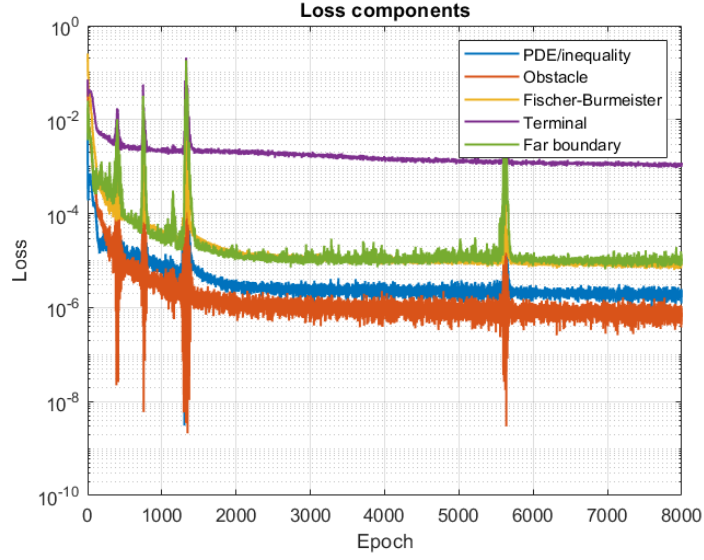


FIGURE 2. two-asset American basket put, a soft-penalty PINN: Loss vs Epoch

point, only in their respective regions. For this problem, the PINN approach is competitive with classical Monte Carlo once its loss functional accurately encodes the underlying variational inequality, and not before.

REFERENCES

1. RK Beatson and MJD Powell, *Univariate multiquadric approximation: quasi-interpolation to scattered data*, Constructive approximation **8** (1992), no. 3, 275–288.
2. Fischer Black and Myron Scholes, *The pricing of options and corporate liabilities*, Journal of political economy **81** (1973), no. 3, 637–654.
3. Michael J Brennan and Eduardo S Schwartz, *Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis*, Journal of Financial and Quantitative Analysis **13** (1978), no. 3, 461–474.
4. John C Cox, Stephen A Ross, and Mark Rubinstein, *Option pricing: A simplified approach*, Journal of financial Economics **7** (1979), no. 3, 229–263.
5. Robert Geske and Kuldeep Shastri, *Valuation by approximation: a comparison of alternative option valuation techniques*, Journal of Financial and Quantitative Analysis **20** (1985), no. 1, 45–71.
6. Y Chung Hon, *A quasi-radial basis functions method for american options pricing*, Computers & Mathematics with Applications **43** (2002), no. 3-5, 513–524.
7. Maziar Raissi, Paris Perdikaris, and George E Karniadakis, *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), 686–707.
8. Richard J Rendleman and Brit J Bartter, *Two-state option pricing*, The Journal of Finance **34** (1979), no. 5, 1093–1110.
9. Eduardo S Schwartz, *The valuation of warrants: Implementing a new approach*, Journal of Financial Economics **4** (1977), no. 1, 79–93.
10. Lars Stentoft, *Assessing the least squares monte-carlo approach to american option valuation*, Review of Derivatives research **7** (2004), no. 2, 129–168.
11. Lixin Wu and Yue-Kuen Kwok, *A front-fixing finite difference method for the valuation of american options*, Journal of Financial Engineering **6** (1997), no. 4, 83–97.
12. Zongmin Wu and Schaback Robert, *Shape preserving properties and convergence of univariate multiquadric quasi-interpolation*, Acta Mathematicae Applicatae Sinica **10** (1994), no. 4, 441–446.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF ENGINEERING AND TECHNOLOGY PESHAWAR
 Email address: ddnmarjan@gmail.com; (corresponding author)

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF DAYTON USA
 Email address: musman1@dayton.edu; (corresponding author)