The deep parametric PDE method and applications to option pricing

Ahmad Aghapour Amirmahdi Hosseinabadi

Sharif University of Technology

July 25, 2023

Presentation Overview

1. Introduction
   - Problem Definition
   - Novelty and main contributions
2. The deep parametric PDE method
   - Multivariate option pricing in Black-Scholes model
   - Loss Function
   - Neural Network
   - Incorporating prior knowledge
   - Details about parameters
   - Implied Volatility
   - Reference pricers
     - Vanilla European basket options
     - European basket call options
     - Option with geometric payoff
3. Results
   - Overview
   - Single-asset call option
   - Basket call options
     - Fixed parameters
     - Varying parameters
   - Option basket with geometric payoff
   - Greeks

Problem Definition

- recent work mostly solve PDEs with given parameters
- so for every PDEs we need to train whole model again (DGM , PINN)
- some of them only solve PDE for single point and can't solve PDEs in whole domain.(Deep BSDE)
- In this work writers want to solve these problem by presenting Deep Parametric Model that solve al PDEs with same parameter and stock price dimension

Novelty and main contributions

1. Single network training
2. Applications in Finance
3. Incorporating prior knowledge

Multivariate option pricing in Black-Scholes model

- Option price:

$$u(t,x;\mu) = e^{-rt}\mathbb{E}(G(S_T(\mu))|S_t(\mu)=s) \tag{1}$$

- Payoff function:

$$g(x) = G(e^x) = (\frac{1}{d}\sum_{i=1}^{d} e^{x_i} - K)^+ = (\frac{1}{d}\sum_{i=1}^{d} s_i - K)^+ = max(0, \frac{1}{d}\sum_{i=1}^{d} s_i - K) \tag{2}$$

- We have $f(t,x;\mu) = 0$ and:

$$L_x u(t,x;\mu) = ru(t,x;\mu) - \sum_{i=1}^{d}(r - \frac{\sigma_i^2}{2})\partial_{x_i} u(t,x;\mu) - \sum_{i,j=1}^{d} \frac{\rho_{ij}\sigma_i\sigma_j}{2}\partial_{x_i x_j} u(t,x;\mu), \tag{3}$$

Loss Function

- Least squares function
- Interior residual

$$\mathcal{L}_{int}(n) = \frac{1}{|Q \times P|} \int_P \int_Q (\partial_t n(t,x;\mu) + L_x n(t,x;\mu) - f(t,x;\mu))^2 d_{(t,x)} d_\mu \tag{4}$$

- Initial residual

$$\mathcal{L}_{ic}(n) = \frac{1}{|\Sigma \times P|} \int_P \int_\Sigma (n(0,x;\mu) - g(x;\mu))^2 d_x d_\mu \tag{5}$$

- Total loss function

$$\mathcal{L}(n) = \mathcal{L}_{int}(n) + \mathcal{L}_{ic}(n) \tag{6}$$

Loss Function : Monte-Carlo quadrature

- Generate random data from uniform distribution:

$$(t^{(i)}, x^{(i)}, \mu^{(i)}) \in Q \times P, \quad (\hat{x}^{(i)}, \hat{\mu}^{(i)}) \in \Omega \times P, \quad i = 1, 2, ..., N \tag{7}$$

- Monte-Carlo quadrature:

$$\begin{aligned}
\mathcal{L}_{int}(n) &= \frac{1}{N}\sum_{i=1}^{N}(\partial_t n(t^{(i)}, x^{(i)}; \mu^{(i)}) + L_x n(t^{(i)}, x^{(i)}; \mu^{(i)}) - f(t^{(i)}, x^{(i)}; \mu^{(i)}))^2 \\
\mathcal{L}_{ic}(n) &= \frac{1}{N}\sum_{i=1}^{N}(n(0, \hat{x}^{(i)}; \hat{\mu}^{(i)}) - g(\hat{x}^{(i)}; \hat{\mu}^{(i)}))^2
\end{aligned} \tag{8}$$

Deep Neural Network

- Complex network architecture

$$
\begin{aligned}
S^1 &= \sigma(W^1 \overrightarrow{x} + b^1) \\
Z^l &= \sigma(U^{z,l} \overrightarrow{x} + W^{z,l} S^l + b^{z,l}), \quad l = 1, \dots, L, \\
G^l &= \sigma(U^{g,l} \overrightarrow{x} + W^{g,l} S^1 + b^{g,l}), \quad l = 1, \dots, L, \\
R^l &= \sigma(U^{r,l} \overrightarrow{x} + W^{r,l} S^l + b^{r,l}), \quad l = 1, \dots, L, \\
H^l &= \sigma(U^{h,l} \overrightarrow{x} + W^{h,l} (S^l \odot R^l) + b^{h,l}), \quad l = 1, \dots, L, \\
S^{l+1} &= (1 - G^l) \odot H^l + Z^l \odot S^l, \quad l = 1, \dots, L, \\
n(t, x; \mu) &= W S^{L+1} + b
\end{aligned}
\tag{9}
$$

- Objective:

$$\theta^* = argmin_\theta \mathcal{L}(n_\theta) \tag{10}$$

Incorporating prior knowledge

- No-arbitrage bound + Time value:

$$u(t,x;\mu) = v(t.x;\mu) + \hat{u}(t,x;\mu) \tag{11}$$

- No-arbitrage bound:

$$\hat{u} = \max\left(\frac{\sum_{i=1}^{d} e^{x_i}}{d} - Ke^{-rt}, 0\right) \tag{12}$$

- Transformed PDE:

$$\begin{aligned} &\partial_t v(t,x;\mu) + \mathcal{L}_x v(t,x;\mu) = \hat{f}(t,x;\mu), \quad \hat{f} = f - \partial_t \hat{u} \\ &v(0,x;\mu) = \hat{g}(x;\mu), \quad \hat{g} = g - \hat{u} \\ &v(t,x;\mu) = \hat{h}(t,x;\mu), \quad \hat{h} = h - \hat{u} \end{aligned} \tag{13}$$

- we place $\lambda = 0.1$

Incorporating prior knowledge(cont'd)

- $\hat{u}$ is not smooth
- Approximation with Softplus function:

$$\hat{u}_\lambda(t, x; \mu) = \frac{1}{\lambda} \log(1 + e^{\lambda(\frac{1}{d}\sum_{i=1}^{d} e^{x_i} - Ke^{-rt})}), \quad 0 \leq \lambda \tag{14}$$

Details about parameters

- Initial parameter set:

$$\mu = (r, K, (\rho_{ij})_{1 \leq i,j \leq d}, (\sigma_i)_{i=1}^{d}) \tag{15}$$

- Fix strike price at K=100
- Parameterising the correlation matrix:

$$\begin{gathered} \hat{\rho}_i = \rho_{i,i+1} \in (-1, 1) \\ \rho_{ji} = \rho_{ij} = \prod_{k=i}^{j-1} \hat{\rho}_k \quad j > i \end{gathered} \tag{16}$$

Implied Volatility

- Implied volatility in Basket options:

$$BS(t, \sum_{i=1}^{d} \frac{e^{x_i}}{d}, r, \hat{\sigma}_{iv}, K) = n(t, x; \mu) = c \tag{17}$$

- Bounds on c:

$$c_{lb} = (\sum_{i=1}^{d} \frac{e^{x_i}}{d} - Ke^{-rt}) \quad \leq \quad c \quad \leq \quad (\sum_{i=1}^{d} \frac{e^{x_i}}{d}) = c_{ub} \tag{18}$$

Vanilla European options

- Black-Scholes Formula

$$c(t,x;\mu) = BS(t, e^x, r, \sigma, K) = \Phi(d_1)e^x - \Phi(d_2)Ke^{-rt}$$
$$d_1 = \frac{1}{\sigma\sqrt{t}}(x - \ln(K) + rt + \frac{\sigma^2 t}{2}), \quad d_2 = d_1 - \sigma\sqrt{t}$$

- No longer available for basket options

European basket call options

- Evaluate the option price by integrating smoothed payoff

Key steps:

1. Decomposing the covariance matrix yields λ_i and $(v_{i,j})_{ij}$ such that for independent $Y_i \in \mathrm{N}(0, \lambda_i^2)$, the stochastic process of the logarithmic prices is:

$$\log(S^i_T(\mu)) = x_i + (r - \frac{\sigma_i^2}{2})t + Y_1 + \sum_{j=1}^{d} v_{i,j} Y_j, \tag{19}$$

2. Solving a conditional expectation for Y_1 given $Y_2, Y_3, \ldots, Y_d$, yields the option price as a (d-1)-dimensional problem with a smooth payoff function:

$$c(t, x; \mu) = \mathbb{E}(BS(1, h(Y_2, \ldots, Y_d), 0, \lambda_1, e^{-rt}K)$$
$$h(Y_2, \ldots, Y_d) = \frac{1}{d} \sum_{i=1}^{d} e^{x_i - \frac{\sigma^2 t}{2}} e^{\sum_{j=2}^{d} v_{i,j} Y_j} \tag{20}$$

European basket call options(cont'd)

3. The function h is smooth and the dimension is reduced by one. We use Gauβ-Hermite quadrature

Option with geometric payoff

- Geometric payoff:

$$G(x) = (e^{\sum_{i=1}^{d} \frac{x_i}{d}} - K)^+ \tag{21}$$

- Consider underlyings of equal correlations and volatilities
- Explicit solution:

$$\begin{aligned}
&u(x, t; \mu) = N(d_1)e^{-qt}e^{\bar{x}} - N(d_2)Ke^{-rt}, \\
&d_1 = \frac{1}{\bar{\sigma}\sqrt{t}}(\bar{x} - \ln(K) + rt - qt + \frac{\bar{\sigma}^2}{2}t), \quad d_2 - d_1 - \bar{\sigma}\sqrt{t}, \\
&\bar{x} = \sum_{i=1}^{d} \frac{x_i}{d}, \quad \bar{\sigma}^2 = \frac{\sigma^2}{d(1 + (d-1)\rho)}, \quad q = \sigma^2 - \bar{\sigma}^2
\end{aligned} \tag{22}$$

Results

- Options with 1 to 8 underlying assets
- Evaluating model error in option pricing, implied volatility, and Greeks
- Parameters:

$$\sigma_i \in [0.1, 0.3], \quad \hat{\rho}_i \in [0.2, 0.8], \quad r \in [0.01, 0.03], \quad T = 4, \quad K = 100, \\ r = 0.02, \quad \sigma_i = 0.2, \quad \hat{\rho}_i = 0.5 \tag{23}$$

Single-asset call option

- compare to Blach-Scholes model

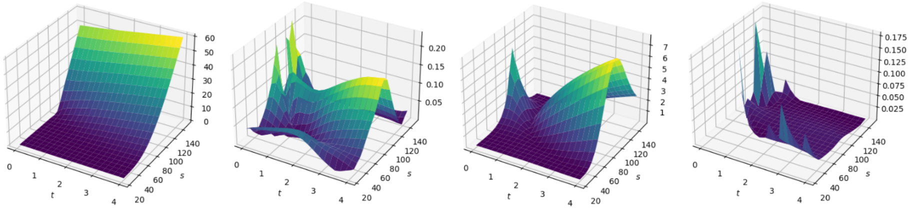

Figure: One-dimensional deep parametric PDE solution (left) and errors (middle left), residual value of the solution (middle right) and relative error of the implied volatility (right). All values shown for fixed parameters.

Basket call options

- Evaluation for fixed parameters
- Parameter dependency
- Consider one to eight underlying assets

Fixed parameters: $d = 2$

- Option Pricing

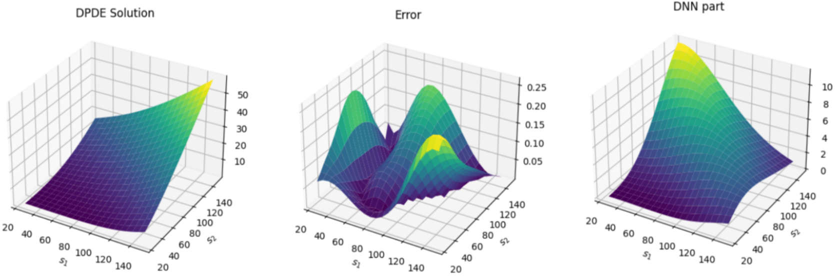

Figure: From left to right: Deep parametric PDE approximation, errors and approximated residual value for two underlyings. All pictures evaluated for $\sigma_1 = 0.1$ and $\sigma_2 = 0.3$ with $\rho_{12} = 0.5$ at maximal time to maturity $t = 4$.

Fixed parameters: $d = 2$ (cont'd)

- Implied volatility

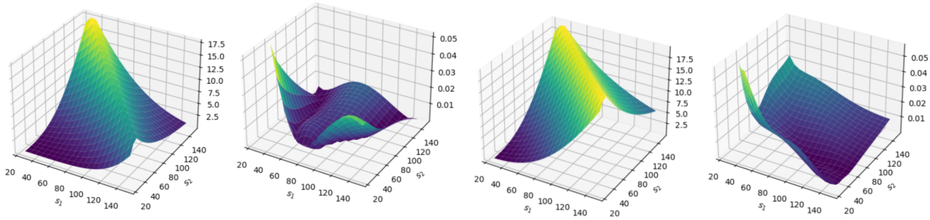

Figure: Left: Difference between the exact option price and the lower trivial no-arbitrage bound c_{lb}. Right: relative error of the implied volatility, computed where the difference is larger than 0.5.

Fixed parameters: $d = 3$

- Option pricing and Implied volatility

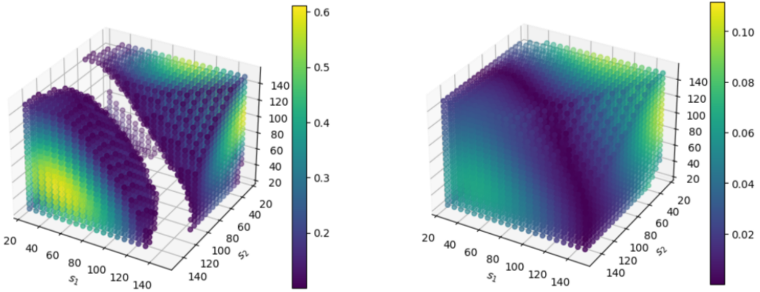

Figure: Different errors for three underlyings. Left: Pricing error, only shown where it is larger than 0.1. Right: Relative error in the volatility, evaluated where the difference to clb is larger than 0.5. Both images share the same perspective, facing the corner where $S_1 = S_2 = S_3 = 25$. The parameters are $\sigma_1 = \sigma_3 = 0.1$, $\sigma_2 = 0.2$, $\rho_{1,2} = 0.2$ and $\rho_{2,3} = 0.5$.

Fixed parameters: $d > 3$

- Option pricing

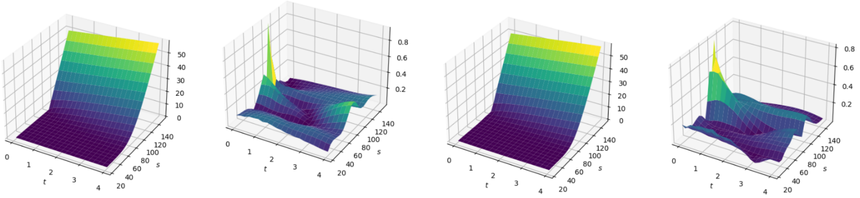

Figure: solution (left) and maximal errors (right) for $d = 5$ and $d = 8$

Varying parameters

- Option pricing and implied volatility
- 1000 random points

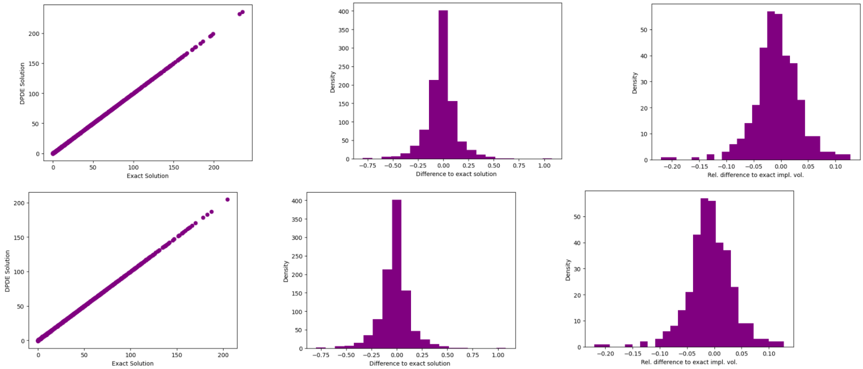

Figure: Top: Five underlyings. Bottom: Eight underlyings. Left: Exact solution and approximation on the x- and the y-axis. Points close to the diagonal show a small error. Middle: Histogram of the difference against the exact solution. Right: Histogram of the relative difference to the implied volatility (IV).

Varying parameters

- Maximal error for samples with the same mean and asset price

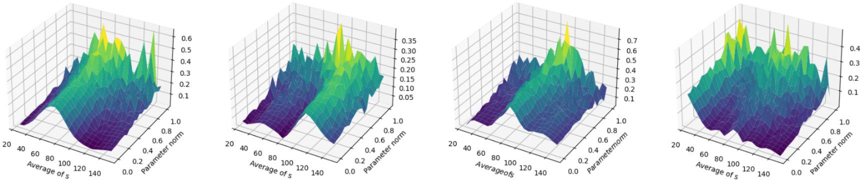

Figure: Maximal error for the basket-options depending on the parameter and the average asset price at maximal time to maturity. From left to right $d = 2, 3, 5, 8$.

Option basket with geometric payoff

- Explicit solution
- Prior knowledge:

$$\hat{u}_\lambda(t, x; \mu) = \frac{1}{\lambda} \log(1 + e^{\lambda(e^{-Bt} e^{\sum_{i=1}^{d} \frac{x_i}{d}} - Ke^{-rt})}) \tag{24}$$

- Option pricing for $d = 2$:

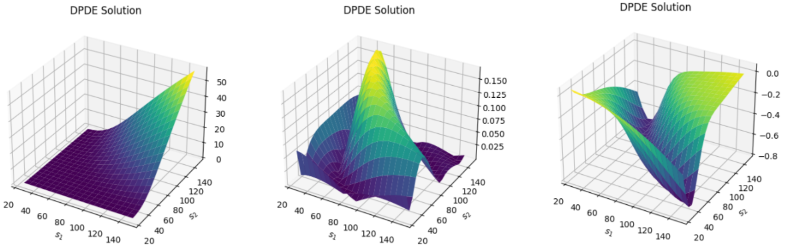

Figure: From left to right: Deep parametric PDE approximation, errors and approximated residual value for two underlyings. All pictures evaluated for $\sigma = 0.1$ at maximal time to maturity using the geometric payoff.

Option basket with geometric payoff(cont'd)

- Maximal error for samples with the same mean and asset price for $d = 3, 5, 8$:

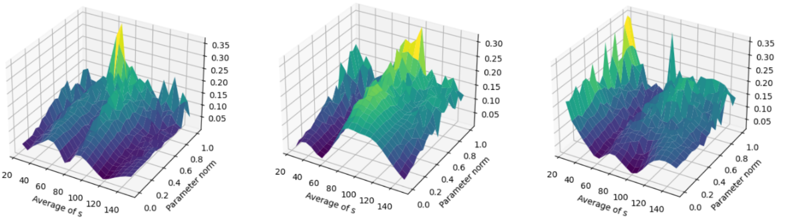

Figure: Maximal error using geometric payoff depending on the parameter and the average asset price at maximal time to maturity for different dimensions. From left to right $d = 3, 5, 8$

- Implied volatility:

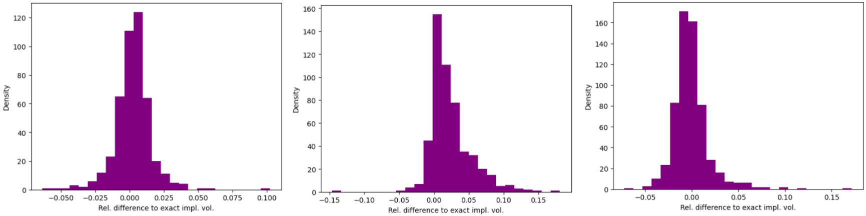

Figure: Histogram of relative difference to exact implied volatility for $d = 3, 5, 8$ (from left to right).

Greeks

- Easy access to derivatives
- Derivative with respect to the log-price of the first underlying
- Derivative with respect to volatility

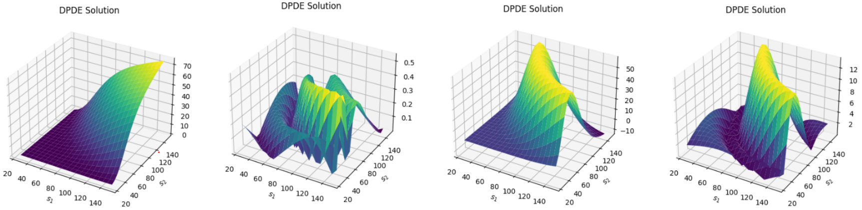

Figure: Accuracy of derivatives for $d = 2$. Left: Derivative; Right: Error.

Pros

- They don't claim that their models can solve PDEs with very high dimension (100 or 150). learning simple function in that space is hard alone! L2 loss of approximation is highly dependent to sampling distribution

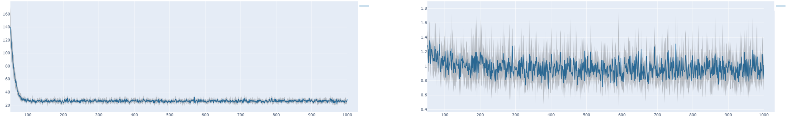

Figure: L2 loss for different sampling distribution for learning min function

Cons

- They implement monte-carlo method in parallel. they stated that monte-carlo method needs 6 min to run. but they can't use GPU A-100 for training their model and run monte-carlo in cpu! we use Cupy and GPU A-100 of google colab and found that monte-carlo method only need 4ms !

Thank you for your attention

Questions?