Deep learning methods for option pricing in jump-diffusion models

21th summer meeting in risk, finance and stochastics, 9–13 September 2024

Costas Smaragdakis (kesmarag@aegean.gr)

Collaborators : E. Georgoulis, A. Papapantoleon

UNIVERSITY OF THE AEGEAN

DEPARTMENT OF STATISTICS AND ACTUARIAL–FINANCIAL MATHEMATICS

An option pricing problem involves determining the fair value of an option, which is a financial derivative that gives the holder the right (but not the obligation) to buy or sell an asset at a specified price on or before a specified date.

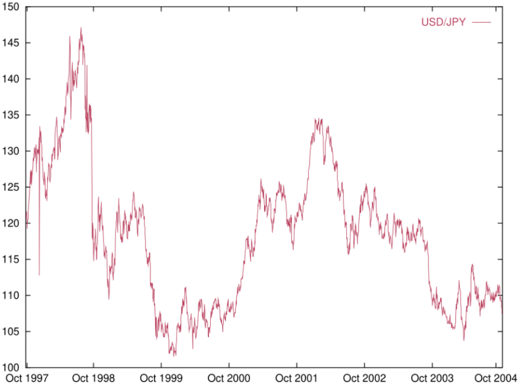
Types of Options

- **Call Option**: The right to buy an asset at a specified strike price.
- **Put Option**: The right to sell an asset at a specified strike price.

Basket Options

Options based on the weighted average of several underlying assets. It can be more complex to price due to the correlations between the assets in the basket.

- Pricing options in scenarios where the underlying stock values exhibit discontinuities.
- Options involving a large number of underlying assets ($d \geq 5$).



- A **Jump Diffusion Model** is an extension of the classical **Black-Scholes model** used in finance.
- It incorporates both continuous diffusion and jumps to capture asset price movements.
- The model is represented by the following stochastic differential equation (SDE):

$$\frac{dS_t}{S_t} = b dt + \sigma dW_t + dJ_t$$

- S_t is the asset price at time t, b is the drift rate, σ is the volatility, W_t is a Brownian motion process, and J_t is a jump process.

- The **Merton Model** is a Jump Diffusion Model introduced by Merton in 1976.
- The jump process J_t is modeled as a Poisson process with intensity λ and jump size e^{Z_i}, where Z_i (i.i.d) are normal distributed.

$$J_t = \sum_{i=1}^{N_t} (e^{Z_i} - 1)$$

- The Merton model's stochastic equation is:

$$\frac{dS_t}{S_t} = bdt + \sigma dW_t + (e^{Z} - 1)dN_t$$

- The model is widely used in option pricing, risk management, and credit risk modeling.

$$S_t = S_0 \exp\left(bt + \sigma W_t + \sum_{k=1}^{N_t} Z_k\right),\ t \in \mathbb{T} = [0, T]$$

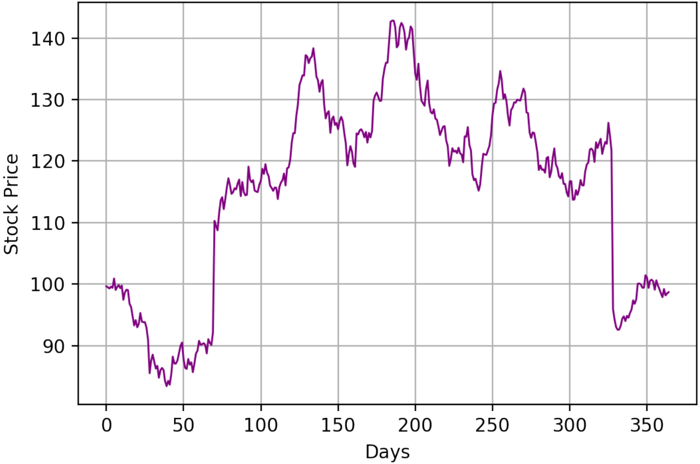
W_t : Brownian motion

(i.i.d) $Z_k \sim \mathcal{N}(\mu_J, \sigma_J^2)$: Normal random variable

N_t : Poisson process

$$\text{(drift term)} \quad b = r - \frac{1}{2}\sigma^2 - \lambda \exp\left(\mu_J + \frac{1}{2}\sigma_J^2\right) + \lambda$$

A simulation path



We assume d correlated stocks with values modelled by the stochastic processes

$$S_t^{(i)} = S_0^{(i)} \exp\left(b_i t + \sigma_i W_t^{(i)} + \sum_{k=1}^{N_t} Z_k^{(i)} \right),\ t \in \mathbb{T} = [0, T],\ i = 1, \dots, d$$

$$W_t^{(i)} : \text{Brownian motions}$$

$$Z_k^{(i)} : \text{Normal random variables}$$

$$N_t : \text{Poisson process}$$

$$\mathrm{Corr}[W_t^{(i)}, W_t^{(j)}] = \rho_{ij} \in [-1, 1], \quad \mathrm{Corr}[Z_k^{(i)}, Z_k^{(j)}] = \rho_{Jij} \in [-1, 1]$$

Payoff function

- $\{\alpha_i\}_{i=1,\ldots,d}$: weights on underlyings ($\sum_i \alpha_i = 1$)
- K : the strike price

$$\text{Payoff}(S) = \left(\sum_{i=1}^{d} \alpha_i S_i - K\right)^+$$

Moneynesses : $X_i = S_i/K$

$$\text{Payoff}(X) = \left(\sum_{i=1}^{d} \alpha_i X_i - 1\right)^+$$

Arbitrage-free price

$$u(t,x) := \mathbb{E}^{\mathbb{Q}}[e^{-rt}\text{Payoff}(X_t)|X_0 = x], \quad [0,T] \ni t : \text{time of maturity}$$

$$[0,\infty)^d \ni x = (x_1, \ldots, x_d)^T : \text{the initial values of the assets}$$

From moneyness to actual prices

$$\tilde{u}(t,s) = Ku(t,x)$$

Arbitrage-free price

$$\mathbb{E}^{\mathbb{Q}}[e^{-rt}\text{Payoff}(S_t)|S_0], \quad t : \text{time of maturity}$$

Using FTAP and Feynman-Kac the option price is provided by:

$$\partial_t u(t,x) + \mathcal{A}u(t,x) = 0,\ t > 0,\ x \in [0,\infty)^d$$

$$u(0,x) = u_0(x) = \text{Payoff}(x),\ x \in [0,\infty)^d$$

- $\mathcal{A}$: PIDE operator

$$\mathcal{A}u = -\sum_{i,j=1}^{d} a_{ij}(x)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{d} b_i(x)\frac{\partial u}{\partial x_i} + ru - I_\nu[u]$$

where $a_{ij}(x) = a_{ji}(x),\ i,j = 1,\ldots,d$ and $I_\nu[u] = \int_{\mathbb{R}^d}[u(t,xe^z) - u(t,x)]\nu(dz)$.

Universal approximation theorem

A feedforward neural network with a single hidden layer containing a finite number of neurons can approximate any continuous function on a compact subset of $\mathbb{R}^d$ to any desired degree of accuracy, given sufficient training parameters and an appropriate activation function.

Approaches

- Global representation

$$u(t,x) \approx U(t,x;\theta),\ \forall t \in [0,T],\ \forall x \in [0,\infty)^d$$

- Global in space representation

$$t_k = kT/n,\ k = 0,\dots,n$$

$$u(t_k,x) \approx U^k(x;\theta^k),\ \forall x \in [0,\infty)^d$$

$$\mathcal{A}u = -\sum_{i,j=1}^{d} a_{ij}(x)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{d} b_i(x)\frac{\partial u}{\partial x_i} + ru - I_\nu[u],\ \ I_\nu[u] = \int_{\mathbb{R}^d}[u(t, xe^z) - u(t,x)]\nu(dz)$$

We rewrite the operator as follows

$$\mathcal{A}u = -\sum_{j=1}^{d}\frac{\partial}{\partial x_j}\left(\sum_{i=1}^{d} a_{ij}(x)\frac{\partial u}{\partial x_i}\right) + \sum_{i=1}^{d}\left(b_i(x) + \sum_{j=1}^{d}\frac{\partial}{\partial x_j}a_{ij}(x)\right)\frac{\partial u}{\partial x_i} + ru - I_\nu[u]$$

$$\mathcal{A}u = \mathcal{L}u + f[u]$$

$$\begin{aligned} \mathcal{L}u &= -\sum_{j=1}^{d}\frac{\partial}{\partial x_j}\left(\sum_{i=1}^{d} a_{ij}(x)\frac{\partial u}{\partial x_i}\right) + ru \quad \text{(symmetric)} \\ f[u] &= \sum_{i=1}^{d}\left(b_i(x) + \sum_{j=1}^{d}\frac{\partial}{\partial x_j}a_{ij}(x)\right)\frac{\partial u}{\partial x_i} - I_\nu[u] \quad \text{(remainder)} \end{aligned}$$

Suppose we would like to estimate the solution of the following equation:

$$\begin{aligned} \partial_t u(t,x) + \mathcal{L}u(t,x) + f[u(t,x)] &= 0, \quad x \in [0, x_{\max}]^d = \Omega, \quad t \in [0,T] \\ u(0,x) &= \text{Payoff}(x) \end{aligned}$$

We consider a time subdivision of the time interval $[0,T]$

$$\tau = T/n, \quad t_k = k\tau, \quad k = 0, \dots, n$$

$$u^k(x) \doteq u(t_k, x)$$

Implicit–Explicit BDF-p

$$\frac{\beta_p u^k - \sum_{j=0}^{p-1} \beta_j u^{k-j-1}}{\tau} + \mathcal{L}u^k + \sum_{j=0}^{p-1} \gamma_j f[u^{k-j-1}] = 0$$

Implicit–Explicit BDF-2

$$\frac{u^k - 4/3u^{k-1} + 1/3u^{k-2}}{\tau} + \frac{2}{3}\mathcal{L}u^k + \frac{2}{3}(2f[u^{k-1}] - f[u^{k-2}]) = 0, \quad k = 2, \dots, n$$

Initialization (u^0, u^1)

- u^0 is known (initial condition)
- $u^0 \to u^1$

$$\frac{u^1 - u^0}{\tau} + \mathcal{L}u^1 + f[u^0] = 0$$

$$\mathcal{A}u = \mathcal{L}u + f[u], \quad E[u] = \int_\Omega \mathcal{E}(x, u, \nabla u) dx$$

$$\mathcal{L}u = -\sum_{j=1}^{d} \frac{\partial}{\partial x_j}\left(\sum_{i=1}^{d} a_{ij}(x)\frac{\partial u}{\partial x_i}\right) + ru \quad \text{(symmetric)}$$

$$= \frac{d}{d\epsilon}E[u+\epsilon v]\Big|_{\epsilon=0} = \int_\Omega \sum_{k=1}^{d} \frac{\partial \mathcal{E}}{\partial(\nabla u)_k}\frac{\partial v}{\partial x_k} + \frac{\partial \mathcal{E}}{\partial u} v \; dx$$

$$= \int_\Omega \left(-\sum_{k=1}^{d} \frac{\partial}{\partial x_k}\left(\frac{\partial \mathcal{E}}{\partial(\nabla u)_k}\right) + \frac{\partial \mathcal{E}}{\partial u}\right) v \; dx.$$

$$\mathcal{L}u = 0 \Leftrightarrow -\sum_{k=1}^{d} \frac{\partial}{\partial x_k}\left(\frac{\partial \mathcal{E}}{\partial(\nabla u)_k}\right) + \frac{\partial \mathcal{E}}{\partial u} = 0 \quad \text{(Euler-Lagrange)}$$

$$\mathcal{E} = \frac{1}{2}\sum_{i,j=1}^{d} a_{ij}(x)\frac{\partial u}{\partial x_i}\frac{\partial u}{\partial x_j} + \frac{1}{2}ru^2$$

- Approximate $u^1, u^2, \dots, u^p$ using the implicit-explicit Euler method

$$\beta_p u - \sum_{j=0}^{p-1} \beta_j u^{k-j-1} + \tau \mathcal{L} u + \tau \sum_{j=0}^{p-1} \gamma_j f[u^{k-j-1}] = 0, \quad k \geq p$$

Minimization Problem : $u^{k-p}, \dots, u^{k-1} \to u^k$

$$L = \frac{1}{2}\left(\beta_p u - \sum_{j=0}^{p-1} \beta_j u^{k-j-1}\right)^2 + \tau \mathcal{E}[u] + \tau \sum_{j=0}^{p-1} \gamma_j f[u^{k-j-1}] u$$

Dirichlet energy functional

$$\mathcal{E}[u] = \frac{1}{2}\left(\sum_{i,j=1}^{d} a_{ij}(x) \frac{\partial u}{\partial x_i} \frac{\partial u}{\partial x_j} + r u^2\right)$$

$$\mathcal{C}[u] \doteq \frac{1}{2}\left\|\beta_p u - \sum_{j=0}^{p-1} \beta_j u^{k-j-1}\right\|_{L^2(\Omega)}^2 + \tau \int_\Omega \mathcal{E}[u] dx + \tau \int_\Omega \sum_{j=0}^{p-1} \gamma_j f[u^{k-j-1}] u \; dx \to \min$$

ANN Representation

We approximate the solution at the step t_k by a ANN with parameters θ^k.

$$u^k(x) \approx U^k(x;\theta^k)$$

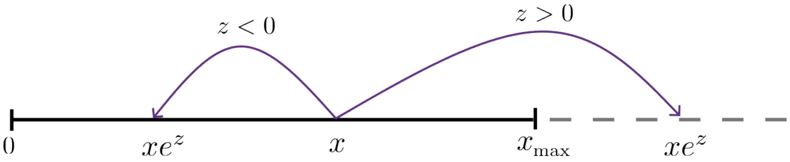
(E. H. Georgoulis, M. Loulakis, and A. Tsiourvas (2023))

Discretized Cost Functional

$$\begin{aligned}\mathscr{C}_k(\theta) \quad &:= \quad \frac{(x_{\max})^d}{N}\sum_{i=1}^{N}\Bigg\{\frac{1}{2}\Bigg[\beta_p U^k(x^i;\theta) - \sum_{j=0}^{p-1}\beta_j U^{j(k)}(x^i;\theta^{j(k)})\Bigg]^2 \\ &+ \quad \tau\mathcal{E}[U^k(x^i;\theta)] + \tau\sum_{j=0}^{p-1}\gamma_j f[U^{j(k)}(x^i;\theta^{j(k)})]U^{j(k)}(x^i;\theta)\Bigg\}\end{aligned}$$

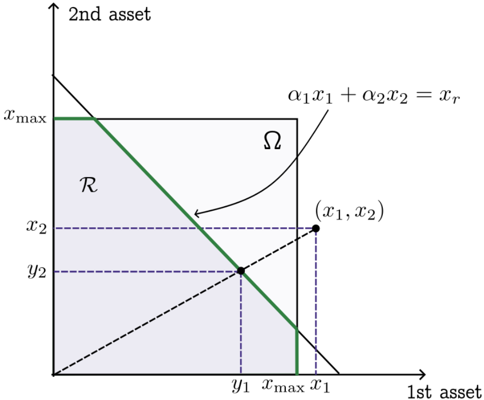
where $j(k) = k - j - 1$

Optimization step : $\theta^k \leftarrow \min_\theta \mathscr{C}_k(\theta)$



- $u(t, x)$ depends on $u(t, xe^z)$

$$y := q(x)x, \quad q(x) = \begin{cases} x_{\max}/\max\{x_i\}, & \text{if } \max\{x_i\} \geq \max\left(\sum_{i=1}^{d} \alpha_i x_i, x_r\right) x_{\max}/x_r \\ x_r/\max\left(\sum_{i=1}^{d} \alpha_i x_i, x_r\right), & \text{otherwise.} \end{cases}$$



$$x \in \mathbb{R}_+^d \to y \in \mathcal{R} \cup \partial\mathcal{R} \to U^k(y;\theta^k) \to U^k(x;\theta^k)$$

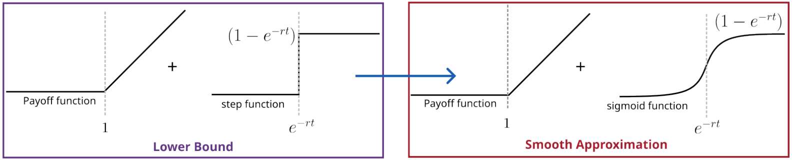
$$U^k(x;\theta^k) \approx U^k(y;\theta^k) + \sum_{i=1}^{d} \alpha_i (x_i - y_i)$$

(Approximation of the solution at time t_k) = (Lower Bound at time t_k) + (positive function)

$$U^k(y;\theta^k) = \tilde{v}(t_k, y) + w^k(y;\theta^k), \quad y \in \mathcal{R} \cup \partial\mathcal{R}$$

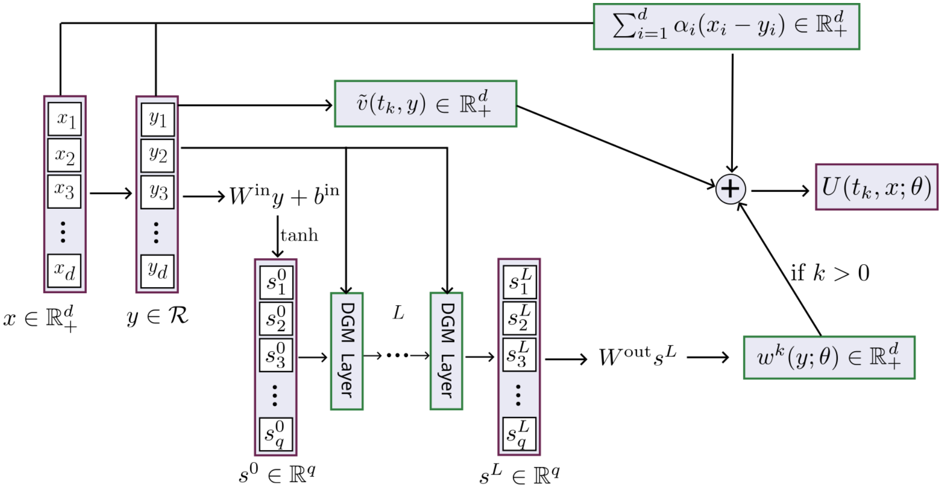
Lower bound $v(t,x)$

$$v(t,x) = \text{Payoff}(x) + (1 - e^{-rt})H(\sum_{i=1}^{d} \alpha_i x_i - e^{-rt})$$

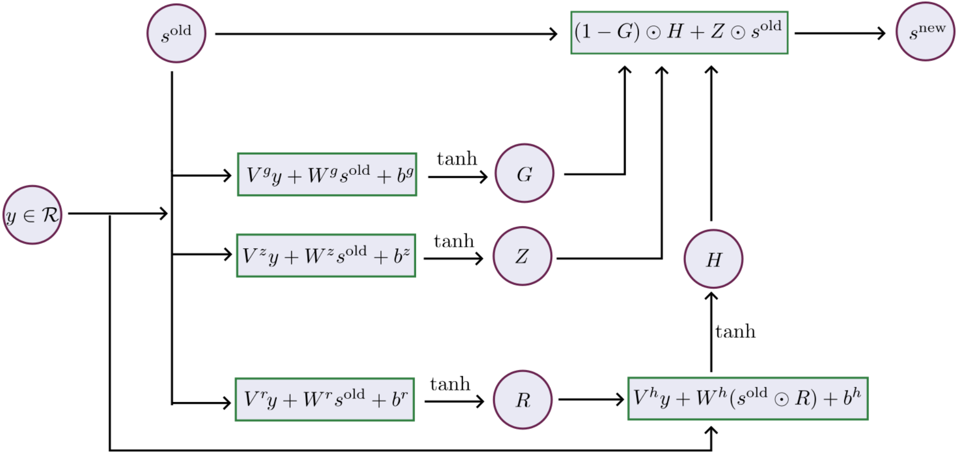


Smooth approximation $\tilde{v}(t,x;\eta)$

$$\tilde{v}(t,x;\eta) = \text{Payoff}(x) + (1 - e^{-rt})\text{Sig}(\sum_{i=1}^{d} \alpha_i x_i - e^{-rt};\eta), \ \text{Sig}(x;\eta) = (1 + e^{-\eta x})^{-1}, \ \eta > 0$$


$\sum_{i=1}^{d} \alpha_i(x_i - y_i) \in \mathbb{R}^d_+$
$\tilde{v}(t_k, y) \in \mathbb{R}^d_+$
x_1 x_2 x_3 x_d
y_1 y_2 y_3 y_d
$W^{\text{in}} y + b^{\text{in}}$
tanh
$U(t_k, x; \theta)$
if $k > 0$
s^0_1 s^0_2 s^0_3 s^0_q
DGM Layer
L
DGM Layer
s^L_1 s^L_2 s^L_3 s^L_q
$W^{\text{out}} s^L$
$w^k(y; \theta) \in \mathbb{R}^d_+$
$x \in \mathbb{R}^d_+$
$y \in \mathcal{R}$
$s^0 \in \mathbb{R}^q$
$s^L \in \mathbb{R}^q$

DGM Layer


s^{old}
$(1-G)\odot H + Z\odot s^{\text{old}}$
s^{new}
$y \in \mathcal{R}$
$V^g y + W^g s^{\text{old}} + b^g$
tanh
G
$V^z y + W^z s^{\text{old}} + b^z$
tanh
Z
H
tanh
$V^r y + W^r s^{\text{old}} + b^r$
tanh
R
$V^h y + W^h (s^{\text{old}} \odot R) + b^h$

$$\mathscr{C}_k(\theta) := \frac{(x_{\max})^d}{N}\sum_{i=1}^{N}\Bigg\{\frac{1}{2}\Big[\beta_p U^k(x^i;\theta) - \sum_{j=0}^{p-1}\beta_j U^{j(k)}(x^i;\theta^{j(k)})\Big]^2$$

$$+ \quad \tau\mathcal{E}[U^k(x^i;\theta)] + \tau\sum_{j=0}^{p-1}\gamma_j f[U^{j(k)}(x^i;\theta^{j(k)})]U^{j(k)}(x^i;\theta)\Bigg\}$$

$$\mathcal{E}[u] = \frac{1}{2}\left(\sum_{i,j=1}^{d} a_{ij}(x)\frac{\partial u}{\partial x_i}\frac{\partial u}{\partial x_j} + ru^2\right), \quad f[u] = \sum_{i=1}^{d}\left(b_i(x) + \sum_{j=1}^{d}\frac{\partial}{\partial x_j}a_{ij}(x)\right)\frac{\partial u}{\partial x_i} - I_\nu[u]$$

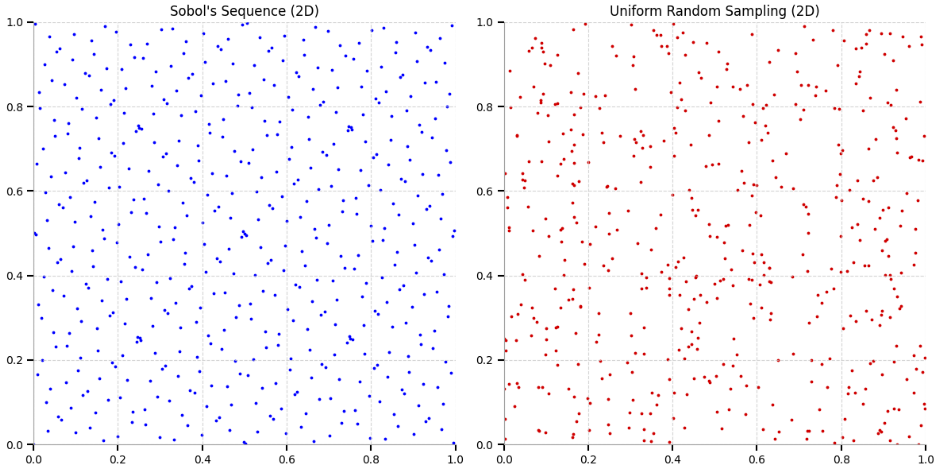
Merton model

$$a_{ij}(x) = \frac{1}{2}\sigma_i\rho_{ij}\sigma_j x_i x_j, \qquad b_i(x) = [-r + \frac{1}{2}\sigma_i^2 - \lambda\exp(\mu_{Ji} + \frac{1}{2}\sigma_{Ji}^2) - \lambda]x_i$$

$$I_\nu[u] = \lambda\int_{\mathbb{R}^d}(u(t,xe^z) - u(t,x))p(z)dz, \quad p(z) : \text{multivariate normal pdf}$$

$$\sigma_i = 0.5,\ \rho_{ij} = \delta_{ij} + 0.5(1-\delta_{ij}),\ \ i,j = 1,\dots,d,\ r = 0.05 \quad \text{(diffusion parameters)}$$

$$\lambda = 1,\ \mu_{Ji} = 0,\ \sigma_{Ji} = 0.5,\ \rho_{Jij} = \delta_{ij} + 0.2(1-\delta_{ij}),\ i,j = 1,\dots,d \quad \text{(jump parameters)}$$


Sobol's Sequence (2D)
Uniform Random Sampling (2D)
0.0
0.2
0.4
0.6
0.8
1.0

$$\sum_{j=1}^{d} \gamma_j I_\nu[U^{j(k)}(x;\theta^{j(k)})], \quad I_\nu[U^{j(k)}(x;\theta^{j(k)})] = \lambda \int_{\mathbb{R}^d} (U^{j(k)}(xe^z;\theta^{j(k)}) - U^{j(k)}(x;\theta^{j(k)}))p(z)dz$$

The integral : Gauss-Hermite quadrature

In Merton model, the integral is simply the expected value of the function $h^{j(k)}(x,z) = U^{j(k)}(xe^z) - U^{j(k)}(x)$ multiplied by the Poisson parameter λ.

- Singular Value Decomposition (SVD) of Σ_J

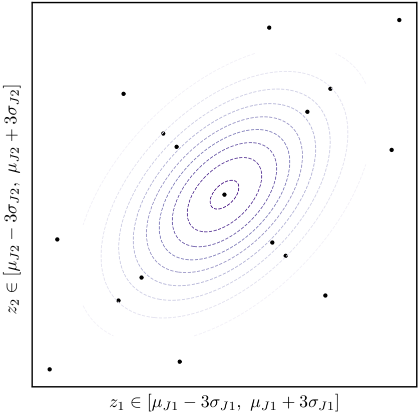
$$\Sigma_J = A\Lambda A^T = A\Lambda^{1/2}\Lambda^{1/2}A^T = BB^T,$$

where $B = A\Lambda^{1/2}$.

- change of variable $Z - \mu_J = \sqrt{2}BY$

$$\begin{aligned} I_\nu[U^{j(k)}(x^i)] = \lambda \int_{\mathbb{R}^d} h^{j(k)}(x^i,z)p(z)dz &= \lambda\pi^{-d/2}\int_{\mathbb{R}^d} \exp(y^T y) h^{j(k)}(x^i, \mu + \sqrt{2}By)dy \\ &\approx \lambda\pi^{-d/2}\sum_{\mathbf{r}\in\Theta_p} h^{j(k)}(x^i, \mu + \sqrt{2}By^{\mathbf{r}})W^{\mathbf{r}}, \end{aligned}$$

The integral : Gauss-Hermite quadrature



$$\sum_{j=1}^{d} \gamma_j I_\nu[U^{j(k)}(x;\theta^{j(k)})], \quad I_\nu[U^{j(k)}(x;\theta^{j(k)})] = \lambda \int_{\mathbb{R}^d} (U^{j(k)}(xe^z;\theta^{j(k)}) - U^{j(k)}(x;\theta^{j(k)}))p(z)dz$$

Unbiased estimator of the integral operator

$$\min_{\phi\in\Phi} \mathbb{E}\left[\mathcal{I}^k(x;\phi) - \sum_{j=1}^{p-1} \gamma_j I_\nu[U^{j(k)}(x;\theta^{j(k)})]\right]^2$$

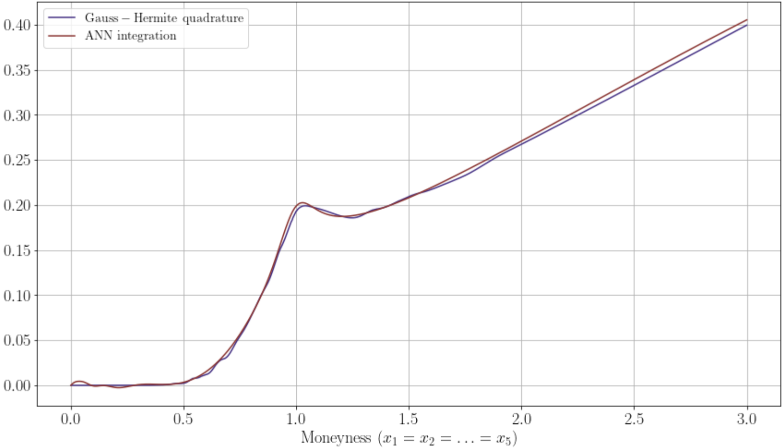
Additional term in the cost functional

Optimizer : (θ^k, ϕ^k)

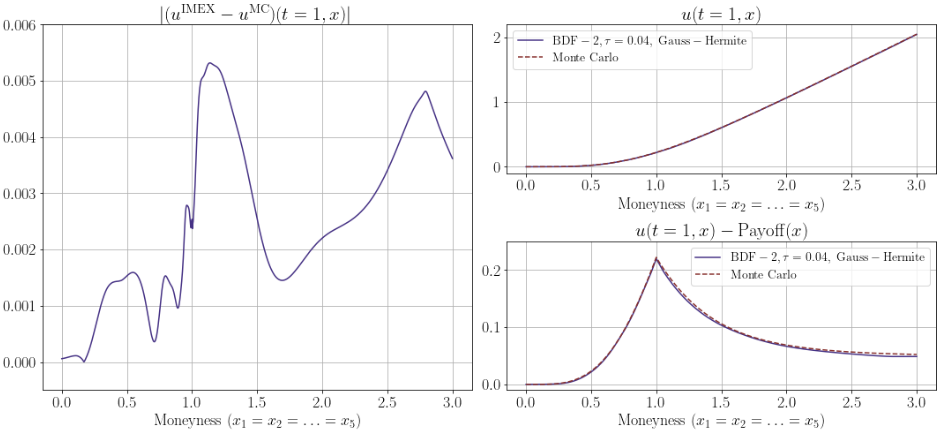
$$\frac{(x_{\max})^d}{N} \sum_{i=1}^{N} \left[\mathcal{I}^k(x^i;\phi) - \frac{\lambda}{M} \sum_{r=1}^{M} \sum_{j=1}^{p-1} \gamma_j h^{j(k)}(x^i, z^r)\right]^2$$

where $\{z^r\}_{r=1}^{M}$ are sampled from $p(z)$.

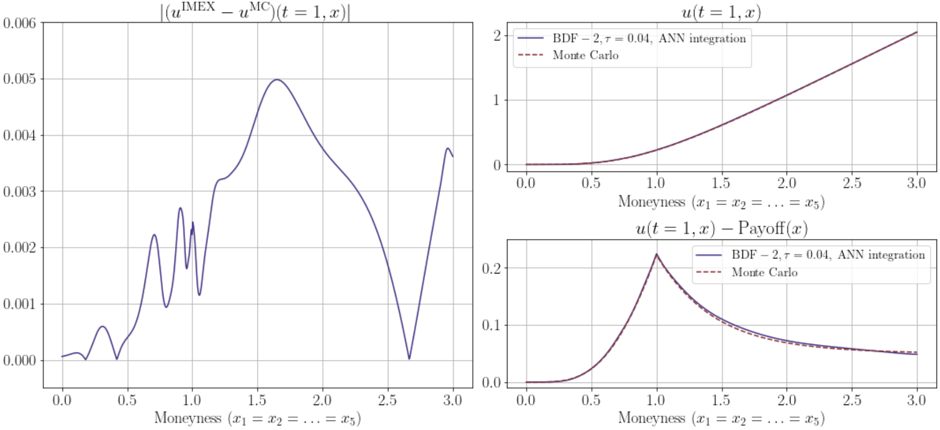
ANN Integration vs Gauss Hermite Quadrature



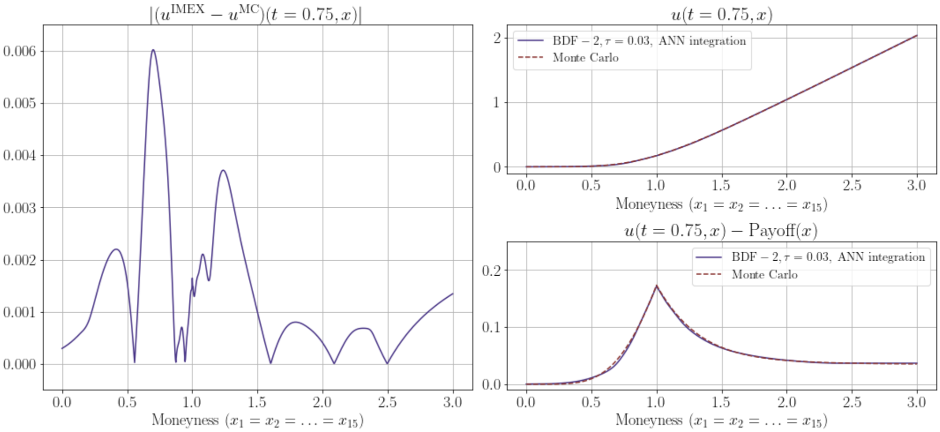
5 assets - 12 months - BDF–2 - Gauss–Hermite Integration



5 assets - 12 months - BDF–2 - ANN Integration



15 assets - 9 months - BDF–2 - ANN Integration



Conclutions

- We proposed a new deep implicit–explicit minimizing movement method for option pricing.
- Our method is capable of accurately approximating the solutions of the partial integro-differential equations (PIDEs) that arise in European basket call options.
- To evaluate the effectiveness of our method, we compared its results with those obtained using Monte-Carlo simulations.

Thank you for your attention

E.H. Georgoulis, A. Papapantoleon, C. Smaragdakis: A deep implicit-explicit minimizing movement method for option pricing in jump-diffusion models. Preprint and submitted for publication, 2024 [arXiv:2401.06740].

Costas Smaragdakis - webpage: `https://kesmarag.github.io`