Time-stepping Deep Gradient Flow Method for Option Pricing in (rough) Diffusion Models

International Conference on Computational Finance

Jasper Rou
joint work with Antonis Papapantoleon

April 4, 2024

Pricing

Price of a derivative with pay-off $\Phi(S_T)$

$$u(t) = \mathbb{E}\left[e^{-r(T-t)}\Phi(S_T)|S_t\right]$$

Pricing

Price of a derivative with pay-off $\Phi(S_T)$

$$u(t) = \mathbb{E}\left[e^{-r(T-t)}\Phi(S_T)|S_t\right]$$

Feynman-Kac formula:

$$\frac{\partial u}{\partial t} + \sum_{i,j=0}^{n} a^{ij}\frac{\partial^2 u}{\partial x_i \partial x_j} - \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} - ru = 0$$

$$u(T) = \Phi(S_T)$$

Pricing

Price of a derivative with pay-off $\Phi(S_T)$

$$u(t) = \mathbb{E}\left[e^{-r(T-t)}\Phi(S_T)|S_t\right]$$

Feynman-Kac formula:

$$\frac{\partial u}{\partial t} + \sum_{i,j=0}^{n} a^{ij}\frac{\partial^2 u}{\partial x_i \partial x_j} - \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} - ru = 0$$

$$u(T) = \Phi(S_T)$$

Can we solve this PDE using a neural network?

Deep Galerkin Method

$$\frac{\partial u}{\partial t} + \sum_{i,j=0}^{n} a^{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} - \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} - ru = 0$$

$$u(T) = \Phi(S_T)$$

Deep Galerkin Method

$$\frac{\partial u}{\partial t} + \sum_{i,j=0}^{n} a^{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} - \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} - ru = 0$$

$$u(T) = \Phi(S_T)$$

Minimize

$$\left\| \frac{\partial u}{\partial t} + \sum_{i,j=0}^{n} a^{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} - \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} - ru \right\|_{[0,T]\times\Omega}^2 + \|u(T) - \Phi(S_T)\|_{\Omega}^2$$

Deep Galerkin Method

$$\frac{\partial u}{\partial t} + \sum_{i,j=0}^{n} a^{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} - \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} - ru = 0$$

$$u(T) = \Phi(S_T)$$

Minimize

$$\left\| \frac{\partial u}{\partial t} + \sum_{i,j=0}^{n} a^{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} - \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} - ru \right\|^2_{[0,T]\times\Omega} + \|u(T) - \Phi(S_T)\|^2_{\Omega}$$

Issue: Taking second derivative makes training in high dimensions slow

Idea

Rewrite PDE as energy minimization problem

Idea

Rewrite PDE as energy minimization problem

- Only first order derivative
- No norm

Idea

Rewrite PDE as energy minimization problem

- Only first order derivative
- No norm

Split in symmetric and non-symmetric part

Splitting method

$$\frac{\partial u}{\partial t} = -\sum_{i,j=0}^{n} a^{ij}\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} + ru$$

Splitting method

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= -\sum_{i,j=0}^{n} a^{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} + ru \\
&= -\sum_{i,j=0}^{n} \frac{\partial}{\partial x_j}\left(a^{ij} \frac{\partial u}{\partial x_i}\right) + \sum_{i,j=0}^{n} \frac{\partial a^{ij}}{\partial x_j} \frac{\partial u}{\partial x_i} + \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} + ru
\end{aligned}
$$

Splitting method

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= -\sum_{i,j=0}^{n} a^{ij}\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} + ru \\
&= -\sum_{i,j=0}^{n} \frac{\partial}{\partial x_j}\left(a^{ij}\frac{\partial u}{\partial x_i}\right) + \sum_{i,j=0}^{n} \frac{\partial a^{ij}}{\partial x_j}\frac{\partial u}{\partial x_i} + \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} + ru \\
&= -\sum_{i,j=0}^{n} \frac{\partial}{\partial x_j}\left(a^{ij}\frac{\partial u}{\partial x_i}\right) + \sum_{i=0}^{n}\left(b^i + \sum_{j=0}^{n}\frac{\partial a^{ij}}{\partial x_j}\right)\frac{\partial u}{\partial x_i} + ru
\end{aligned}
$$

Splitting method

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= -\sum_{i,j=0}^{n} a^{ij}\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} + ru \\
&= -\sum_{i,j=0}^{n} \frac{\partial}{\partial x_j}\left(a^{ij}\frac{\partial u}{\partial x_i}\right) + \sum_{i,j=0}^{n} \frac{\partial a^{ij}}{\partial x_j}\frac{\partial u}{\partial x_i} + \sum_{i=0}^{n} b^i \frac{\partial u}{\partial x_i} + ru \\
&= -\sum_{i,j=0}^{n} \frac{\partial}{\partial x_j}\left(a^{ij}\frac{\partial u}{\partial x_i}\right) + \sum_{i=0}^{n}\left(b^i + \sum_{j=0}^{n}\frac{\partial a^{ij}}{\partial x_j}\right)\frac{\partial u}{\partial x_i} + ru \\
&= -\nabla\cdot(A\nabla u) + ru + F(u) \\
F(u) &= \mathbf{b}\cdot\nabla u
\end{aligned}
$$

Example: Heston model

$$
\begin{aligned}
dS_t &= rS_t dt + \sqrt{V_t} S_t dW_t && S_0 > 0 \\
dV_t &= \kappa(\theta - V_t)dt + \eta\sqrt{V_t} dB_t && V_0 > 0
\end{aligned}
$$

Example: Heston model

$$
\begin{aligned}
dS_t &= rS_t dt + \sqrt{V_t} S_t dW_t && S_0 > 0 \\
dV_t &= \kappa(\theta - V_t) dt + \eta \sqrt{V_t} dB_t && V_0 > 0
\end{aligned}
$$

$$
\frac{\partial u}{\partial t} = -rS\frac{\partial u}{\partial S} - \kappa(\theta - V)\frac{\partial u}{\partial V} - \frac{1}{2}S^2 V \frac{\partial^2 u}{\partial S^2} - \frac{1}{2}\eta^2 V \frac{\partial^2 u}{\partial V^2} - \rho\eta SV \frac{\partial^2 u}{\partial S \partial V} + ru
$$

Example: Heston model

$$\frac{\partial u}{\partial t} = - rS\frac{\partial u}{\partial S} - \kappa(\theta - V)\frac{\partial u}{\partial V} - \frac{1}{2}S^2V\frac{\partial^2 u}{\partial S^2} - \frac{1}{2}\eta^2 V\frac{\partial^2 u}{\partial V^2} - \rho\eta SV\frac{\partial^2 u}{\partial S\partial V} + ru$$

Example: Heston model

$$
\begin{aligned}
\frac{\partial u}{\partial t} =& - rS\frac{\partial u}{\partial S} - \kappa(\theta - V)\frac{\partial u}{\partial V} - \frac{1}{2}S^2 V\frac{\partial^2 u}{\partial S^2} - \frac{1}{2}\eta^2 V\frac{\partial^2 u}{\partial V^2} - \rho\eta SV\frac{\partial^2 u}{\partial S \partial V} + ru \\
=& - rS\frac{\partial u}{\partial S} - \kappa(\theta - V)\frac{\partial u}{\partial V} - \frac{\partial}{\partial S}\left(\frac{1}{2}S^2 V\frac{\partial u}{\partial S}\right) + SV\frac{\partial u}{\partial S} \\
& - \frac{\partial}{\partial V}\left(\frac{1}{2}\eta^2 V\frac{\partial u}{\partial V}\right) + \frac{1}{2}\eta^2\frac{\partial u}{\partial V} - \frac{\partial}{\partial S}\left(\frac{1}{2}\rho\eta SV\frac{\partial u}{\partial V}\right) + \frac{1}{2}\rho\eta V\frac{\partial u}{\partial V} \\
& - \frac{\partial}{\partial V}\left(\frac{1}{2}\rho\eta SV\frac{\partial u}{\partial S}\right) + \frac{1}{2}\rho\eta S\frac{\partial u}{\partial S} + ru
\end{aligned}
$$

Example: Heston model

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= -rS\frac{\partial u}{\partial S} - \kappa(\theta - V)\frac{\partial u}{\partial V} - \frac{1}{2}S^2V\frac{\partial^2 u}{\partial S^2} - \frac{1}{2}\eta^2V\frac{\partial^2 u}{\partial V^2} - \rho\eta SV\frac{\partial^2 u}{\partial S\partial V} + ru \\
&= -rS\frac{\partial u}{\partial S} - \kappa(\theta - V)\frac{\partial u}{\partial V} - \frac{\partial}{\partial S}\left(\frac{1}{2}S^2V\frac{\partial u}{\partial S}\right) + SV\frac{\partial u}{\partial S} \\
&\quad - \frac{\partial}{\partial V}\left(\frac{1}{2}\eta^2V\frac{\partial u}{\partial V}\right) + \frac{1}{2}\eta^2\frac{\partial u}{\partial V} - \frac{\partial}{\partial S}\left(\frac{1}{2}\rho\eta SV\frac{\partial u}{\partial V}\right) + \frac{1}{2}\rho\eta V\frac{\partial u}{\partial V} \\
&\quad - \frac{\partial}{\partial V}\left(\frac{1}{2}\rho\eta SV\frac{\partial u}{\partial S}\right) + \frac{1}{2}\rho\eta S\frac{\partial u}{\partial S} + ru \\
&= -\nabla\cdot\left(\frac{1}{2}\begin{bmatrix} S^2V & \rho\eta SV \\ \rho\eta SV & \eta^2V \end{bmatrix}\nabla u\right) + \begin{bmatrix} SV - rS + \frac{1}{2}\rho\eta S \\ \kappa(V-\theta) + \frac{1}{2}\rho\eta V + \frac{1}{2}\eta^2 \end{bmatrix}\cdot\nabla u \\
&\quad + ru
\end{aligned}
$$

Time Deep Gradient Flow

$$\begin{cases} u_\tau - \nabla \cdot (A\nabla u) + ru + F(u) = 0 & (\tau, \mathbf{x}) \in [0, T] \times \Omega \\ u(0, \mathbf{x}) = \Phi(\mathbf{x}) & \mathbf{x} \in \Omega \end{cases}$$

Time Deep Gradient Flow

$$\begin{cases} u_\tau - \nabla \cdot (A \nabla u) + ru + F(u) = 0 & (\tau, \mathbf{x}) \in [0, T] \times \Omega \\ u(0, \mathbf{x}) = \Phi(\mathbf{x}) & \mathbf{x} \in \Omega \end{cases}$$

- Divide $[0, T]$ in intervals $(\tau_{k-1}, \tau_k]$ with $h = \tau_k - \tau_{k-1}$

$$\frac{U^k - U^{k-1}}{h} - \nabla \cdot \left(A \nabla U^k \right) + rU^k + F\left(U^{k-1} \right) = 0$$

Time Deep **Gradient Flow**

$$\frac{U^k - U^{k-1}}{h} - \nabla \cdot \left(A \nabla U^k \right) + rU^k + F(U^{k-1}) = 0$$

Time Deep **Gradient Flow**

$$\frac{U^k - U^{k-1}}{h} - \nabla \cdot \left(A \nabla U^k \right) + rU^k + F(U^{k-1}) = 0$$

$$0 = \int_\Omega \left(\left(U^k - U^{k-1} \right) + h \left(-\nabla \cdot \left(A \nabla U^k \right) + rU^k + F\left(U^{k-1} \right) \right) \right) v d\mathbf{x}$$

Time Deep **Gradient Flow**

$$\frac{U^k - U^{k-1}}{h} - \nabla \cdot \left(A \nabla U^k\right) + rU^k + F(U^{k-1}) = 0$$

$$\begin{aligned} 0 &= \int_\Omega \left(\left(U^k - U^{k-1}\right) + h\left(-\nabla \cdot \left(A \nabla U^k\right) + rU^k + F\left(U^{k-1}\right)\right)\right) v d\mathbf{x} \\ &= i'(0) \end{aligned}$$

$$i(\tau) = I^k(U^k + \tau v)$$

Time Deep **Gradient Flow**

$$\frac{U^k - U^{k-1}}{h} - \nabla \cdot \left(A \nabla U^k \right) + rU^k + F(U^{k-1}) = 0$$

$$\begin{aligned} 0 &= \int_\Omega \left(\left(U^k - U^{k-1} \right) + h \left(-\nabla \cdot \left(A \nabla U^k \right) + rU^k + F\left(U^{k-1} \right) \right) \right) v d\mathbf{x} \\ &= i'(0) \end{aligned}$$

$$i(\tau) = I^k(U^k + \tau v)$$

$$\begin{aligned} I^k(u) &= \frac{1}{2} \left\| u - U^{k-1} \right\|^2 + h \int_\Omega \frac{1}{2} \left((\nabla u)^T A \nabla u + ru^2 \right) + F\left(U^{k-1} \right) u dx \\ U^k &= \underset{u \in H^1(\Omega)}{\arg\min}\, I^k(u) \end{aligned}$$

Time **Deep** Gradient Flow

$$I^k(u) = \frac{1}{2}\left\|u - U^{k-1}\right\|^2 + h\int_\Omega \frac{1}{2}\left((\nabla u)^T A\nabla u + ru^2\right) + F\left(U^{k-1}\right) udx$$

$$U^k = \underset{u\in H^1(\Omega)}{\arg\min}\, I^k(u)$$

Time **Deep** Gradient Flow

$$
\begin{aligned}
I^k(u) &= \frac{1}{2}\left\| u - U^{k-1} \right\|^2 + h \int_\Omega \frac{1}{2}\left((\nabla u)^T A \nabla u + r u^2 \right) + F\left(U^{k-1} \right) u dx \\
U^k &= \operatorname*{arg\,min}_{u \in H^1(\Omega)} I^k(u) \\
f^k(\theta) &= \operatorname*{arg\,min}_{u \in \mathcal{C}(\theta)} I^k(u) \\
\mathcal{C}(\theta) &= \text{space of neural networks with parameters } \theta
\end{aligned}
$$

Algorithm

for each time step $k = 1, ..., N_t$ **do**

    Initialize $\theta_0^k = \theta^{k-1}$

Algorithm

for each time step $k = 1, ..., N_t$ **do**
    Initialize $\theta_0^k = \theta^{k-1}$
    for each sampling stage n **do**
        Generate random points $\mathbf{x}^i$ for training

Algorithm

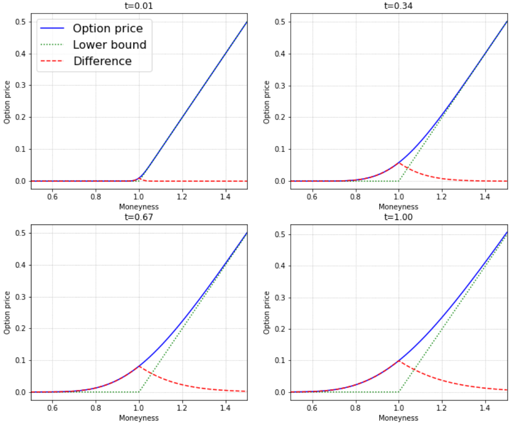
for each time step $k = 1, ..., N_t$ **do**
    Initialize $\theta_0^k = \theta^{k-1}$
    for each sampling stage n **do**
        Generate random points $\mathbf{x}^i$ for training
        Calculate the cost functional $I^k(f(\theta_n^k; \mathbf{x}^i))$

Algorithm

for each time step $k = 1, ..., N_t$ **do**
    Initialize $\theta_0^k = \theta^{k-1}$
    for each sampling stage n **do**
        Generate random points $\mathbf{x}^i$ for training
        Calculate the cost functional $I^k(f(\theta_n^k; \mathbf{x}^i))$
        Take a descent step $\theta_{n+1}^k = \theta_n^k - \alpha_n \nabla_\theta I^k(f(\theta_n^k; \mathbf{x}^i))$
    end for
end for

Base

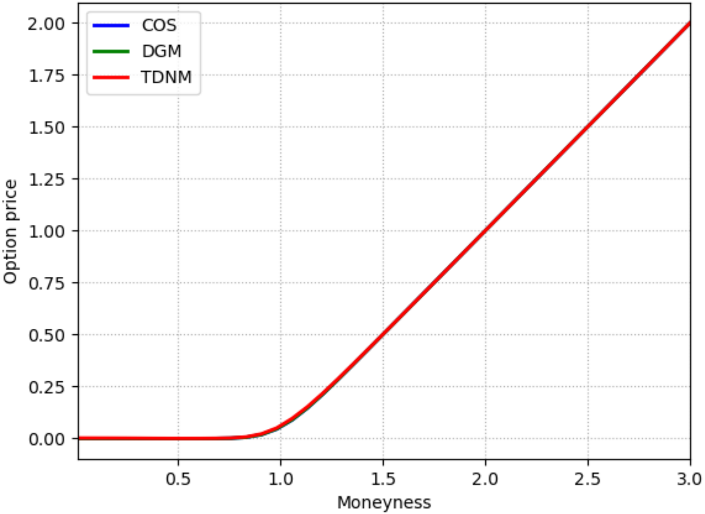
No-arbitrage bound: $u(t,S) \geq S_t - Ke^{-rt}$



Architecture

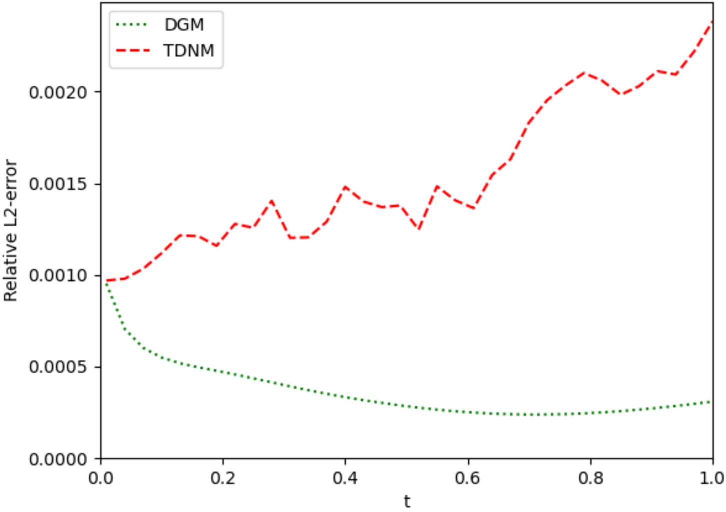
$$
\begin{aligned}
S^1 &= \sigma_1\left(W^1\mathbf{x} + b^1\right) & & \\
Z^l &= \sigma_1\left(U^{z,l}\mathbf{x} + W^{z,l}S^l + b^{z,l}\right) & & l = 1, \ldots, L \\
G^l &= \sigma_1\left(U^{g,l}\mathbf{x} + W^{g,l}S^l + b^{g,l}\right) & & l = 1, \ldots, L \\
R^l &= \sigma_1\left(U^{r,l}\mathbf{x} + W^{r,l}S^l + b^{r,l}\right) & & l = 1, \ldots, L \\
H^l &= \sigma_1\left(U^{h,l}\mathbf{x} + W^{h,l}\left(S^l \odot R^l\right) + b^{h,l}\right) & & l = 1, \ldots, L \\
S^{l+1} &= \left(1 - G^l\right) \odot H^l + Z^l \odot S^l & & l = 1, \ldots, L \\
f(\theta) &= \text{base} + \sigma_2\left(WS^{L+1} + b\right) & & \sigma_2 > 0
\end{aligned}
$$

Heston



COS
DGM
TDNM
Option price
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.5
1.0
1.5
2.0
2.5
3.0
Moneyness

Heston


DGM
TDNM
Relative L2-error
0.0000
0.0005
0.0010
0.0015
0.0020
0.0
0.2
0.4
0.6
0.8
1.0
t

Lifted Heston

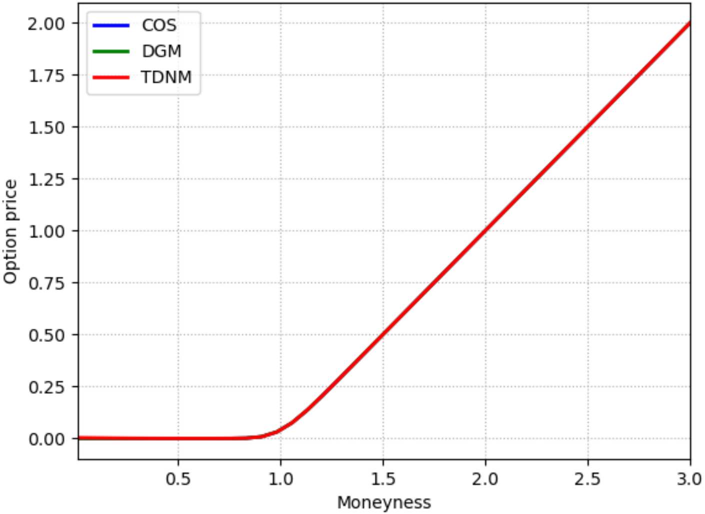
- Rough Heston: more accurate, but V not Markovian

Lifted Heston

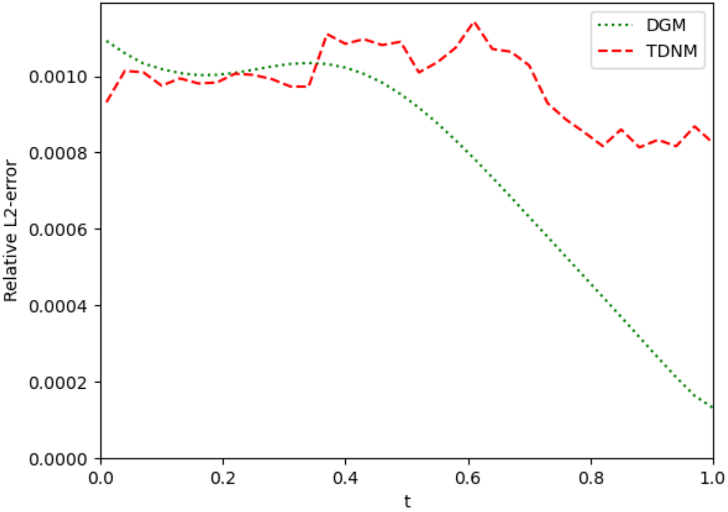
- Rough Heston: more accurate, but V not Markovian
- Lifted Heston: Markovian, but multiple dimensions

$$
\begin{aligned}
dS_t &= rS_t dt + \sqrt{V_t^n} S_t dW_t & S_0 > 0 \\
V_t^n &= g^n(t) + \sum_{i=1}^{n} c_i^n V_t^{n,i} & \\
dV_t^{n,i} &= -\left(\gamma_i^n V_t^{n,i} + \lambda V_t^n\right) dt + \eta \sqrt{V_t^n} dB_t & V_0^{n,i} = 0 \\
g^n(t) &= V_0 + \lambda\theta \sum_{i=1}^{n} c_i^n \int_0^t e^{-\gamma_i^n (t-s)} ds &
\end{aligned}
$$

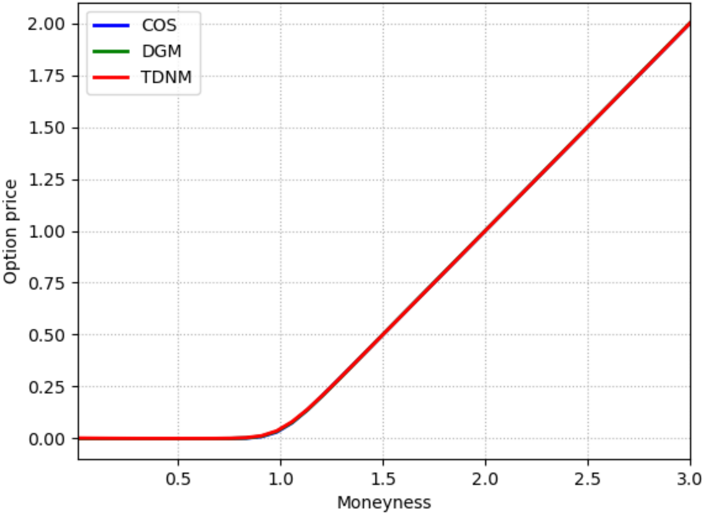
Lifted Heston, $n = 1$



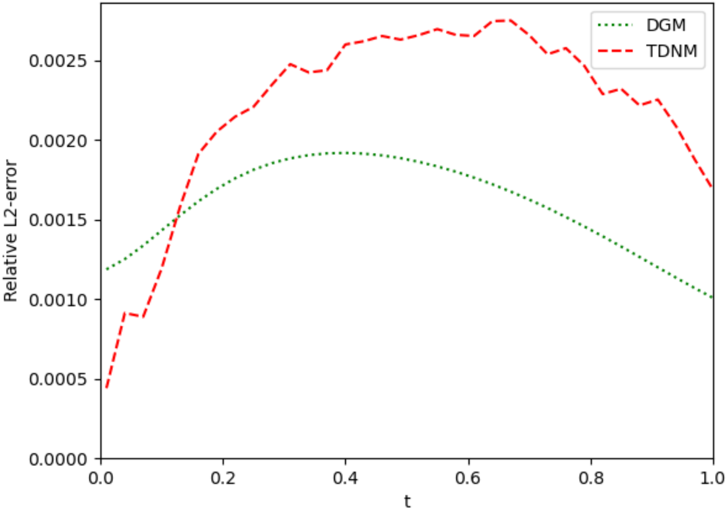
Lifted Heston, $n = 1$



Lifted Heston, $n = 20$



Lifted Heston, $n = 20$



Running times

Model	Heston	LH, n=1	LH, n=20
DGM	12.5×10^3	13.3×10^3	56.1×10^3
TDGF	6.0×10^3	6.4×10^3	7.6×10^3

Table: Training time

Running times

Model	Heston	LH, n=1	LH, n=20
DGM	12.5×10^3	13.3×10^3	56.1×10^3
TDGF	6.0×10^3	6.4×10^3	7.6×10^3

Table: Training time

Model	Heston	LH, n=1	LH, n=20
COS	0.018	8.9	10.4
DGM	0.0016	0.0034	0.0053
TDGF	0.0060	0.020	0.025

Table: Computing time

Conclusion

	Accurate	Fast
Simple model	$\times$	✓
Complicated model	✓	$\times$

Conclusion

	Accurate	Fast
Simple model	$\times$	✓
Complicated model	✓	$\times$
Complicated model with neural networks	✓	✓

Time-stepping Deep Gradient Flow Method for Option Pricing in (rough) Diffusion Models

International Conference on Computational Finance

Jasper Rou

April 4, 2024

j.g.rou@tudelft.nl www.jasperrou.nl