Time Deep Gradient Flow Method for Option Pricing in rough Diffusion Models

SIAM Conference on Financial Mathematics and Engineering

Jasper Rou
joint work with Antonis Papapantoleon

July 17, 2025

Pricing

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

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

Pricing

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

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

Feynman-Kac formula:

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

$$u(0, \mathbf{x}) = \Phi(\mathbf{x})$$

Rough Heston

- Not Markovian $\implies$ no PDE

Rough Heston

- Not Markovian $\implies$ no PDE
- Lifted Heston: Markovian, but multiple dimensions

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sqrt{V_t^n}S_t\mathrm{d}W_t \qquad S_0 > 0$$

$$V_t^n = g^n(t) + \sum_{i=1}^{n} c_i^n V_t^{n,i}$$

$$\mathrm{d}V_t^{n,i} = -\left(\gamma_i^n V_t^{n,i} + \lambda V_t^n\right)\mathrm{d}t + \eta\sqrt{V_t^n}\mathrm{d}B_t \quad V_0^{n,i} = 0$$

$$g^n(t) = V_0 + \lambda\theta\sum_{i=1}^{n} c_i^n \int_0^t e^{-\gamma_i^n(t-s)}\mathrm{d}s$$

Deep Galerkin Method

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

$$u(0, \mathbf{x}) = \Phi(\mathbf{x})$$

Deep Galerkin Method

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

$$u(0, \mathbf{x}) = \Phi(\mathbf{x})$$

Minimize

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

Deep Galerkin Method

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

$$u(0, \mathbf{x}) = \Phi(\mathbf{x})$$

Minimize

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

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

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

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

Splitting method

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

Splitting method

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

Example, $n = 1$

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sqrt{V_t}S_t\mathrm{d}W_t$$
$$\mathrm{d}V_t = -\lambda V_t\mathrm{d}t + \eta\sqrt{V_t}\mathrm{d}B_t$$

Example, $n = 1$

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sqrt{V_t}S_t\mathrm{d}W_t$$

$$\mathrm{d}V_t = -\lambda V_t\mathrm{d}t + \eta\sqrt{V_t}\mathrm{d}B_t$$

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

Example: $n = 1$

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

Example: $n = 1$

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

Example: $n = 1$

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

Example: $n = 1$

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

Example: $n = 1$

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

Time Deep Gradient Flow

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

Time Deep Gradient Flow

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

- Divide $[0, T]$ in intervals $(t_{k-1}, t_k]$ with $h = t_k - t_{k-1}$

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

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**

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

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

Time Deep **Gradient Flow**

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

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

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

Time Deep **Gradient Flow**

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

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

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

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

Time **Deep** Gradient Flow

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

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

Time **Deep** Gradient Flow

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

Algorithm

for each time step $k = 1, ..., K$ **do**
    Initialize $\theta_0^k = \theta^{k-1}$

Algorithm

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

Algorithm

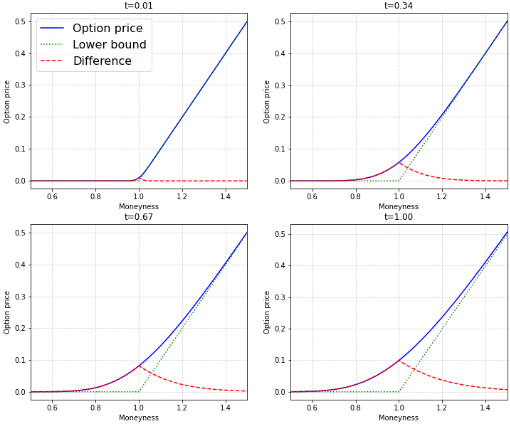
for each time step $k = 1, ..., K$ **do**
    Initialize $\theta_0^k = \theta^{k-1}$
    for each sampling stage n **do**
        Generate random points $\mathbf{x}^i$ for training
        Calculate the cost functional $I^k(f(\mathbf{x}^i; \theta_n^k))$

Algorithm

for each time step $k = 1, ..., K$ **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 $I^k(f(\mathbf{x}^i; \theta_n^k))$
    Take a descent step $\theta_{n+1}^k = \theta_n^k - \alpha_n \nabla_\theta I^k(f(\mathbf{x}^i; \theta_n^k))$
  end for
end for

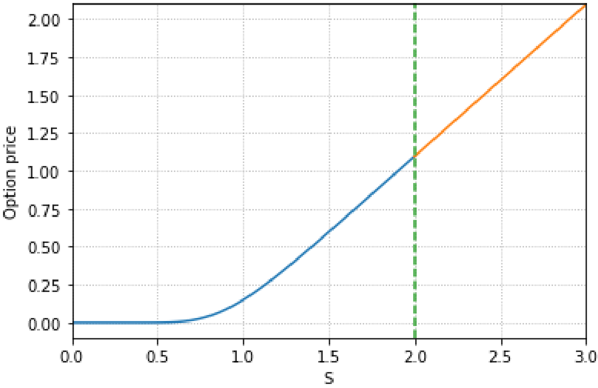
Base

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



Linearization

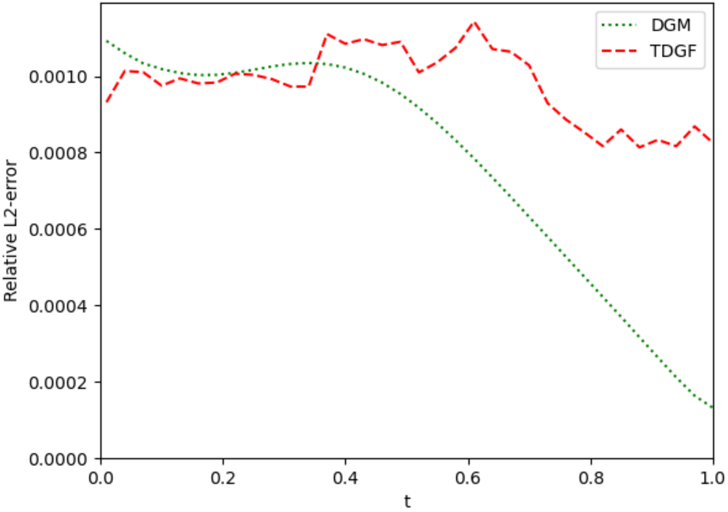
$$f(x_p + y; \theta) = f(x_p; \theta) + y, \quad y > 0.$$



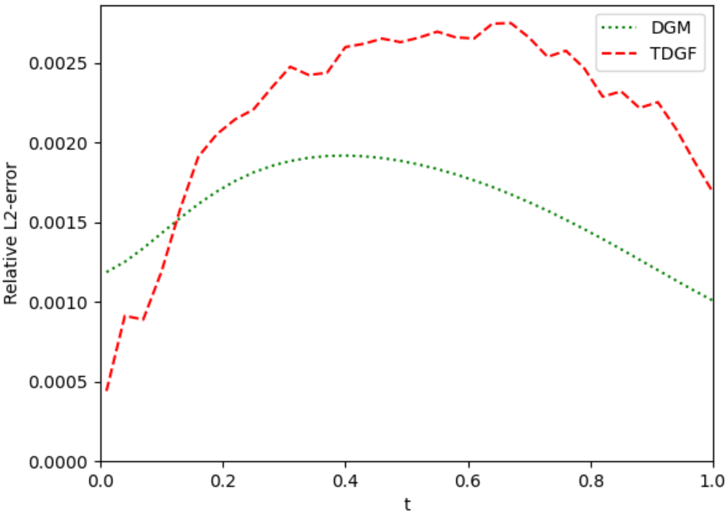
Architecture

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

Lifted Heston, $n = 1$



Lifted Heston, $n = 20$



Running times

Model	LH, n=1	LH, n=20
DGM	12.5×10^3	54.3×10^3
TDGF	5.9×10^3	7.4×10^3

Table: Training time

Running times

Model	LH, n=1	LH, n=20
DGM	12.5×10^3	54.3×10^3
TDGF	5.9×10^3	7.4×10^3

Table: Training time

Model	LH, n=1	LH, n=20
COS	9.1	10.0
DGM	0.0043	0.0015
TDGF	0.0019	0.0018

Table: Computing time

Time Deep Gradient Flow Method for Option Pricing in rough Diffusion Models

SIAM Conference on Financial Mathematics and Engineering

Jasper Rou

July 17, 2025

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