

HighDimPDE.jl

Victor Boussange

Swiss Federal Institute for Forest, Snow and Landscape Research
WSL - ETH Zürich, Switzerland

Reaction-diffusion equations in Biology

Reaction-diffusion equations can model concisely the dynamics of populations structured in spatial & in phenotypic space.

$$\partial_t u = u \left[\overbrace{r - bu}^{\text{growth and self regulation}} + \underbrace{\int_{\Omega} \alpha(x, \mathbf{x}) u(t, x) dx}_{\text{interactions}} \right] + \overbrace{\nabla(g \nabla u)}^{\text{evolution}} \quad (1)$$

where $u: \Omega \times [0, T] \rightarrow \mathbb{R}^+$ captures the distribution of the population under study in the multi-dimensional space Ω .

Specific systems are e.g.

- Tumor cells (short time scale \sim day),
- Planktons (intermediate time \sim years),
- Tetrapods (large-scale eco-evolutionary dynamics $\sim 10^5$ years)

Curse of dimensionality

- Solving PDEs like eq. (1) is NOT easy!
- The **computational complexity** of standard approaches to numerically solve eq. (1) grows exponentially with the number of dimensions d .
- Standard numerical schemes for solving PDEs suffer the **curse of dimensionality**.

Overcoming the curse of dimensionality

- Recent schemes relying on **Machine Learning techniques** have been shown to **overcome** the curse of dimensionality.
- Their computational complexity is no more exponential but **polynomial in d !**
- They rely on the **Feynman-Kac theorem**, that establishes that eq. (2) characterises the expected trajectory X_t of particles which dynamics is described by a **Stochastic Differential Equation** associated to the PDE.

HighDimPDE.jl is a Julia package to solve highly dimensional non-linear, non-local PDEs of the form

$$\begin{aligned} \left(\frac{\partial}{\partial t} u\right)(t, \mathbf{x}) = & \int_{\Omega} f(t, x, \mathbf{x}, u(t, x), u(t, \mathbf{x}), (\nabla_x u)(t, x), (\nabla_x u)(t, \mathbf{x})) \nu_x(dx) \\ & + \langle \mu(t, x), (\nabla_x u)(t, x) \rangle + \frac{1}{2} \text{Trace}(\sigma(t, x)[\sigma(t, x)]^* (\text{Hess}_x u)(t, x)). \end{aligned} \quad (2)$$

where $u: \Omega \rightarrow \mathbb{R}^+$ is a function defined over Ω , a highly dimensional space.

- **HighDimPDE.jl implements schemes that break down the curse of dimensionality**, including

- the Deep BSDE scheme
- the Deep Splitting scheme
- the Multi-Level Picard iterations scheme.

- **HighDimPDE.jl is user friendly, and in particular it is grid-free!**

\Rightarrow no mesh required.

- **HighDimPDE.jl can run both on CPUs and GPUs with no extra fuss**

```
use_cuda = true
```

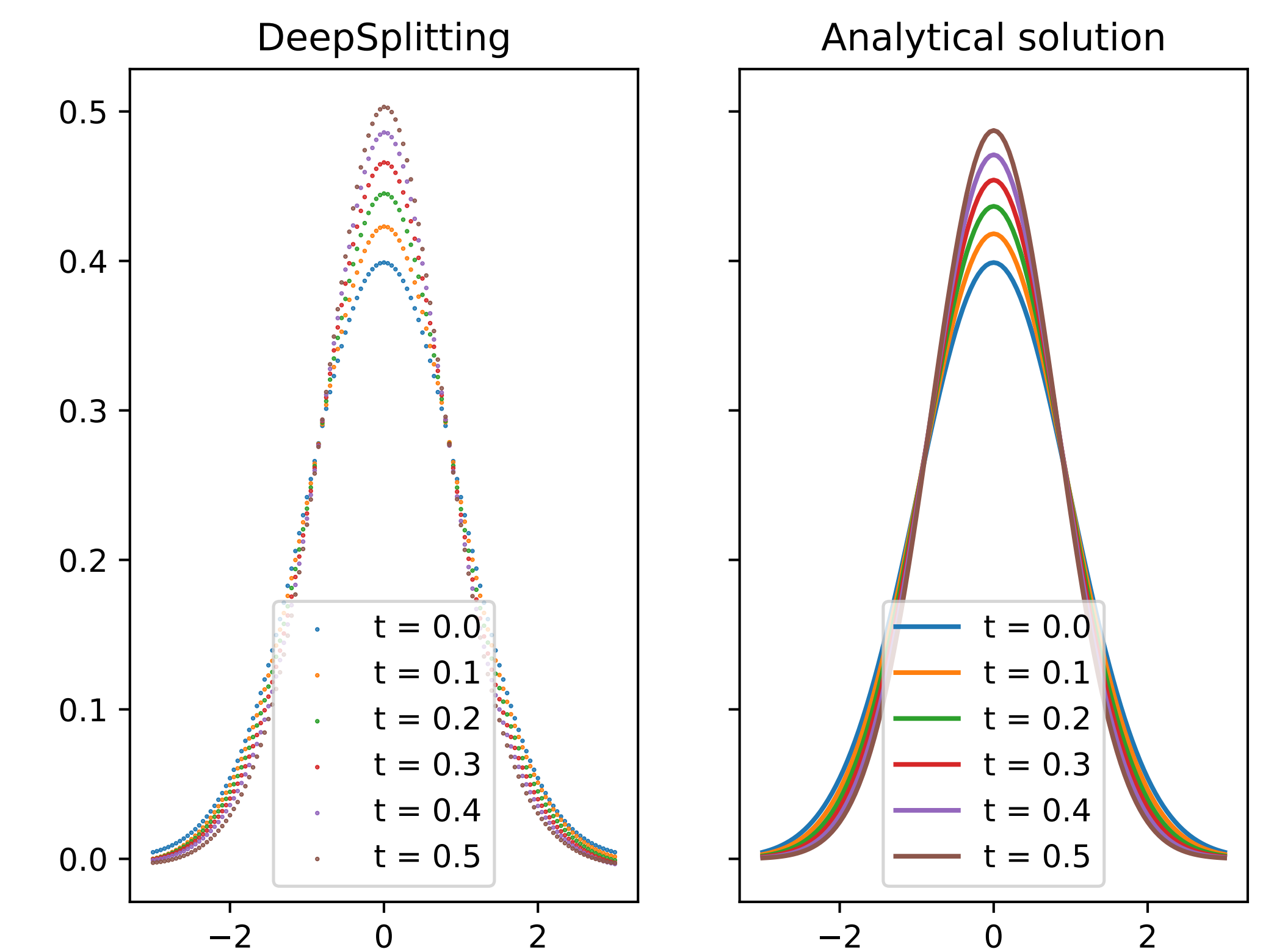
Example: Replicator Mutator equation in dimension $d = 10$

- Consider the Replicator Mutator equation which describes the dynamics of some biological population characterised by 10 traits

$$\left(\frac{\partial}{\partial t} u\right)(t, x) = u(t, x) \left(-\frac{1}{2} \|x\|^2 + \int_D \frac{1}{2} \|x\|^2 u(t, \mathbf{x}) dx \right) + \frac{1}{2} \sigma^2 \Delta_x u(t, x) \quad (3)$$

with initial conditions $u(x, 0) = e^{-\|x\|^2}$.

- Almost impossible to numerically solve eq. (3) in dimension $d = 10$ with e.g. finite differences before the start of CMSB2021.
- Yet we obtain with the DeepSplitting scheme implemented in **HighDimPDE.jl** the following results after ≈ 100 seconds.



Slice of the solution u from eq. (3) along the first dimension of Ω .

Perspectives

The DeepSplitting scheme offers the possibility to **perform parameter fitting with experimental data** on the fly, with minimal extra computational cost.

The package is in its infancy \Rightarrow contributions are welcome!



Take a picture to look at the github repo of HighDimPDE.jl