Physics Informed Neural Networks for PDE inverse problems

Mattéo Clémot
Supervised by Elisa Riccietti and Stefania Bellavia

January 2023

The direct problem

- Solve a PDE on $\Omega \subset \mathbb{R}^d$:

$$\begin{cases} \mathcal{N}[u] = 0 & \text{on } \Omega \\ u = \psi & \text{on } \partial\Omega \end{cases}$$

- Example: $\mathcal{N}[u] = -\Delta u + cu - \varphi$, for given functions c and φ
- Main idea of PINNs [Raissi et al., 2019]: represent the solution with a neural network $u_\theta : \mathbb{R}^d \to \mathbb{R}$ and minimize the residual of the PDE.

Architecture

- MLP with dense layers
- Periodic activation function: $\sigma = \sin$ [Sitzmann et al., 2020]

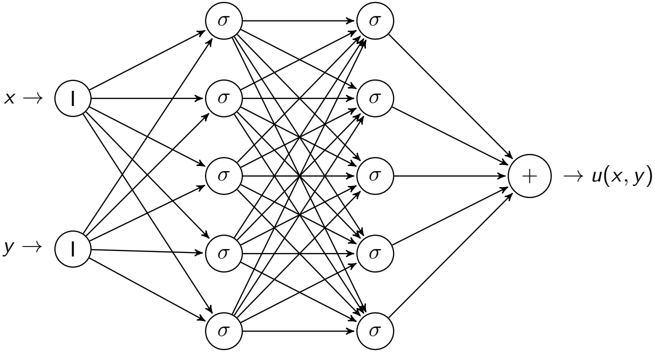


Figure: Example of neural network architecture for a 2-dimensional PDE direct problem, with 2 hidden layers of 5 neurons, and dense layers.

Method

- Formulation as a minimization problem:

$$\min_{\theta} \underbrace{\int_{\Omega} \mathcal{N}[u_\theta]^2}_{\text{PDE residual}} + \underbrace{\int_{\partial\Omega} (u_\theta - \psi)^2}_{\text{boundary conditions}}$$

Method

- Given sets of points $\mathbf{x}_R$, $\mathbf{x}_M$ in Ω, $\mathbf{x}_B$ in $\partial\Omega$, we define the following terms of the loss function:

$$\mathcal{L}_{\mathrm{R}}(\theta) = \frac{1}{n_R}\|\mathcal{N}[u_\theta](\mathbf{x}_R)\|^2$$

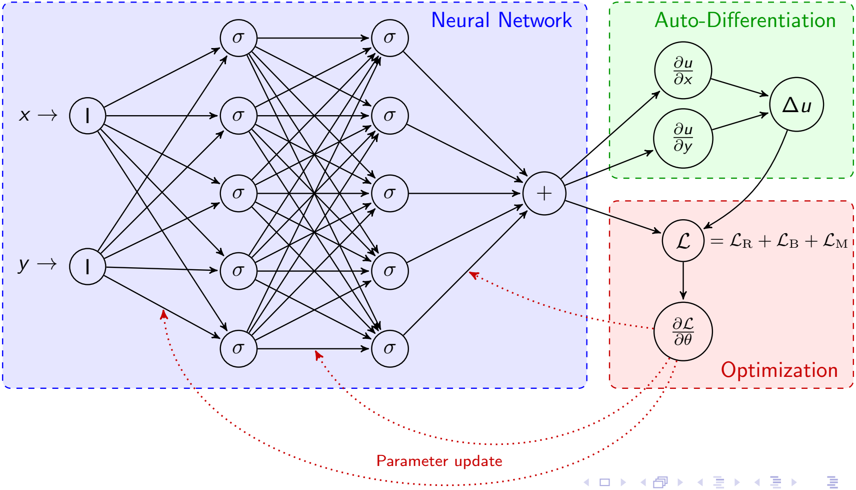
$$\mathcal{L}_{\mathrm{B}}(\theta) = \frac{1}{n_B}\|u_\theta(\mathbf{x}_B) - \psi(\mathbf{x}_B)\|^2$$

$$\mathcal{L}_{\mathrm{M}}(\theta) = \frac{1}{n_M}\|u_\theta(\mathbf{x}_M) - \bar{u}\|^2$$

- Use of auto-differentiation to compute the loss function.
- The optimization problem is:

$$\min_\theta \lambda_{\mathrm{R}}\mathcal{L}_{\mathrm{R}}(\theta) + \lambda_{\mathrm{B}}\mathcal{L}_{\mathrm{B}}(\theta) \underbrace{+\lambda_{\mathrm{M}}\mathcal{L}_{\mathrm{M}}(\theta)}_{\text{if measures provided}}$$

Overview of the method



Example 1: heat equation

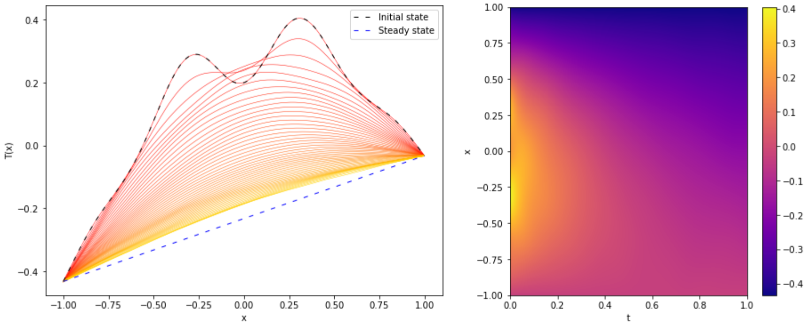


Figure: Resolution of the 1D heat equation $\frac{\partial T}{\partial t} + \frac{\partial^2 T}{\partial x^2} = 0$, from $t = 0$ (red) to $t = 1$ (yellow).

Example 2: non-linear Schrödinger equation

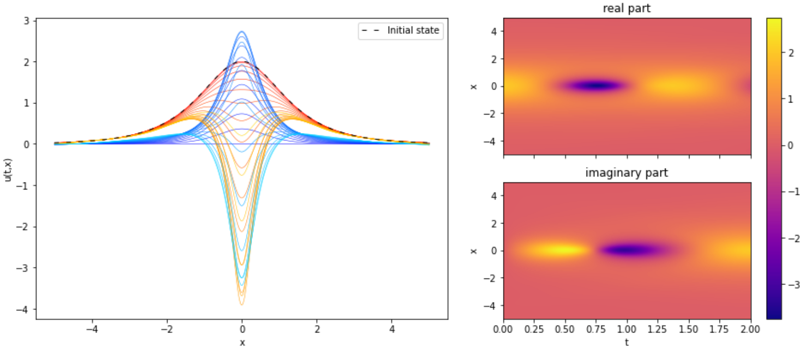


Figure: Resolution of the 1D non-linear Schrödinger equation $i\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial^2 u}{\partial x^2} + |u|^2 u = 0$, from $t = 0$ (red for $\Re(u)$, dark blue for $\Im(u)$) to $t = 1$ (yellow for $\Re(u)$, light blue for $\Im(u)$). Data from [Rudy et al., 2017].

Inverse problems

- Parameter identification in PDEs. Example: recover c in the following elliptical PDE from noisy measures $\bar{u}$ of the solution u:

$$-\Delta u + cu = \varphi.$$

- Inverse problems are typically ill-posed
- Well-posed problem:
  - existence
  - uniqueness
  - **stability of the solution(s)**

Inverse PDE problem

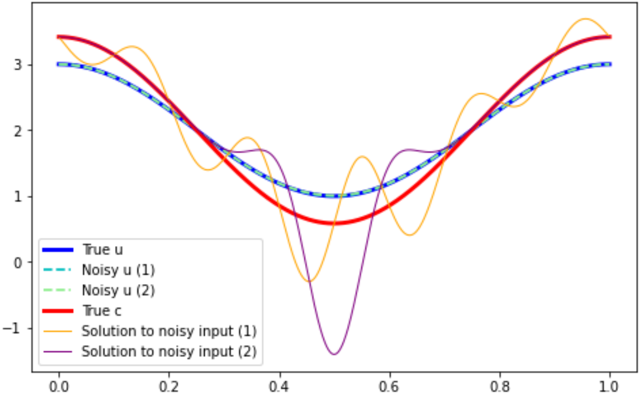


Figure: Ill-posedness of the 1D elliptical inverse problem. These noise terms are not even visible on this graph.

Architecture

$$\begin{aligned} \Phi_\theta : \mathbb{R}^d &\to \mathbb{R}^2 \\ \mathbf{x} &\mapsto (u_\theta(\mathbf{x}), c_\theta(\mathbf{x})) \end{aligned}$$

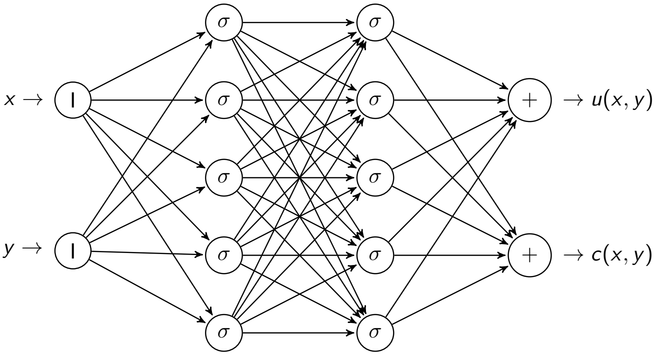


Figure: Neural network architecture for the 2D elliptical inverse problem.

Method

- We use almost the same loss function:

$$\mathcal{L}_{\mathrm{R}}(\theta) = \frac{1}{n_R} \|\mathcal{N}[u_\theta, c_\theta](\mathbf{x}_R)\|^2$$
$$\text{where } \mathcal{N}[u_\theta, c_\theta] = -\Delta u_\theta + c_\theta u_\theta - \varphi$$
$$\mathcal{L}_{\mathrm{B}}(\theta) = \frac{1}{n_B} \|u_\theta(\mathbf{x}_B) - \psi(\mathbf{x}_B)\|^2$$
$$\mathcal{L}_{\mathrm{M}}(\theta) = \frac{1}{n_M} \|u_\theta(\mathbf{x}_M) - \bar{u}\|^2$$

- We minimize:

$$\min_\theta \lambda_{\mathrm{R}}\mathcal{L}_{\mathrm{R}}(\theta) + \lambda_{\mathrm{B}}\mathcal{L}_{\mathrm{B}}(\theta) + \lambda_{\mathrm{M}}\mathcal{L}_{\mathrm{M}}(\theta)$$

2D example, no noise

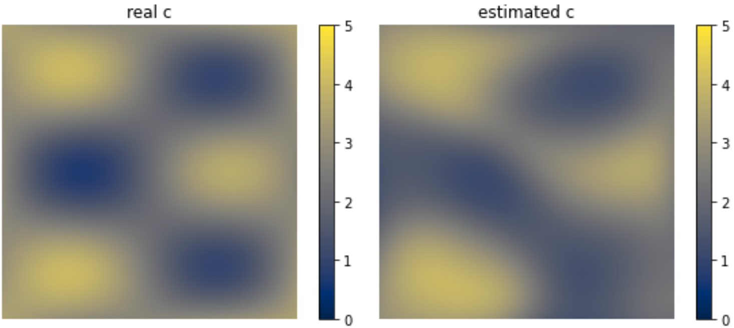


Figure: Real c and estimated c in the 2-dimensional elliptical inverse problem $-\Delta u + cu = \varphi$.
MSE=0.15

2D example, noisy

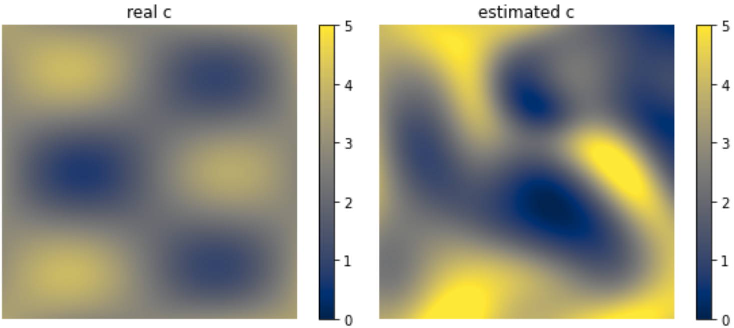


Figure: Real c and estimated c in the 2-dimensional elliptical inverse problem $-\Delta u + cu = \varphi$, with gaussian noise ($\sigma = 2 \times 10^{-2}$) on $\bar{u}$. MSE=1.26

1D noisy example

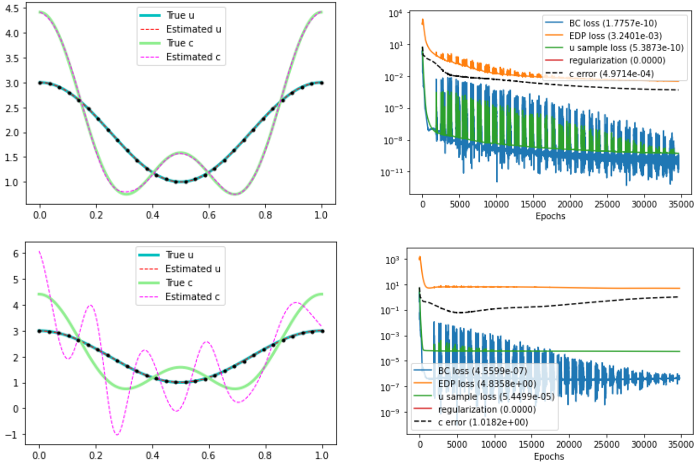


Figure: 1D elliptical inverse problem, noiseless (top) and with noise $\sigma = 10^{-2}$ (bottom).

Regularization

- Need of regularization to deal with the noise, like in classical approaches.
- Do the use of neural networks already have some implicit regularizing properties?

$\rightarrow$ main aim of the internship

Tikhonov regularization in a classical context

Tikhonov regularization

To solve an ill-conditioned system $\mathbf{Ax} = \mathbf{b}$, instead of minimizing $\|\mathbf{Ax} - \mathbf{b}\|^2$, minimize

$$\|\mathbf{Ax} - \mathbf{b}\|^2 + \lambda \|\mathbf{x}\|^2$$

to prefer solutions with smaller norms.

Explicit regularizations

Inspired by Tikhonov regularization, add a regularizing 2-norm term to the loss function. Several attempts:

- on the output: $\lambda \| c_\theta \|^2$
  $\rightarrow$ avoid high values solutions
- on the output's derivative: $\lambda \left\| \frac{\partial c_\theta}{\partial x} \right\|^2$
  $\rightarrow$ avoid oscillating solutions
- on (a subset of) the parameters: $\lambda \| \theta \|^2$ or $\lambda \| \theta_l \|^2$ (e.g. with l the last layer)

Penalization of the weights of the neural network

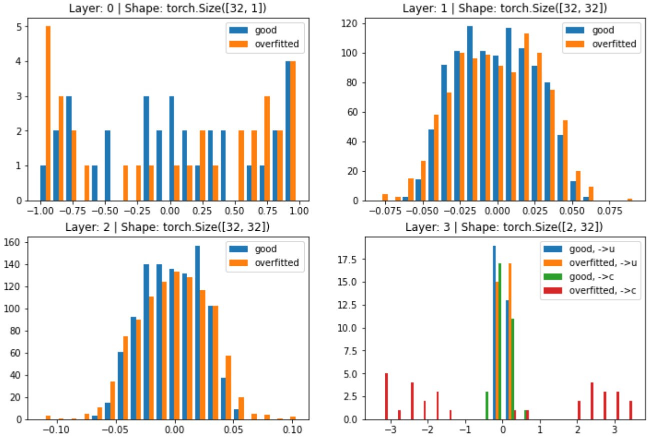


Figure: Histograms of the weights of the layers of two networks with 3 hidden layers of 32 neurons, one giving the right c, the other overfitted on noise, for the 1D elliptic inverse problem.

Implicit regularizations

- Optimizing u and c together is regularizing
- With too much iterations, overfitting of noise. Main difficulty: find a reliable stopping criterion.

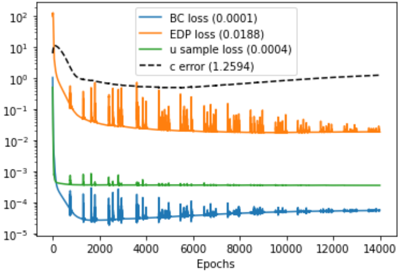


Figure: Too much iterations, leading to noise overfitting. 2D noisy elliptical inverse problem.

This is not enough

- Hard to achieve satisfactory results with λ fixed
- We want to update λ in an automatic way (i.e. use λ_k instead of λ)

Formulations

Two ways of formulating the problem:

- unconstrained formulation:

$$\min \text{PDE} + \text{MEASURES}$$

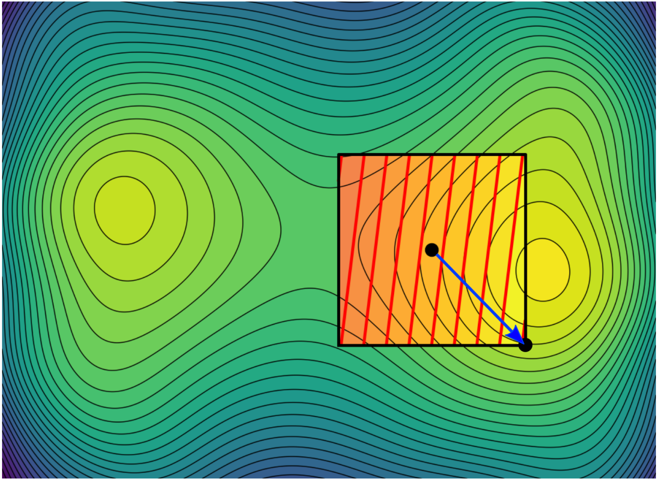
- constrained formulation:

$$\begin{array}{ll} \min & \text{PDE} \\ s.t. & \text{MEASURES} \end{array}$$

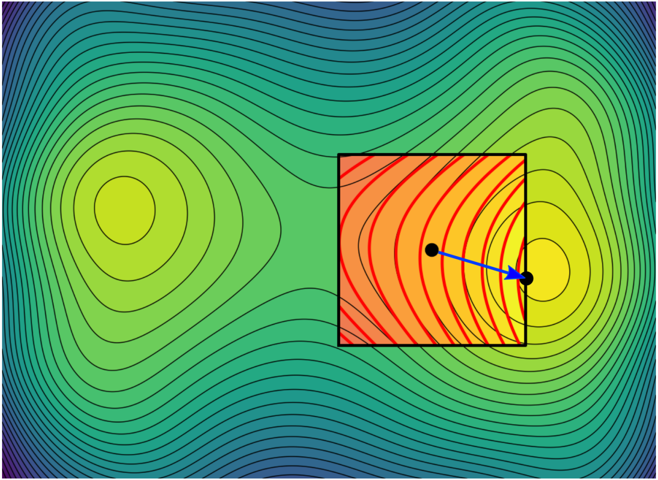
Trust region methods

- A class of globally convergent iterative optimization methods
- Choose a model ℓ (linear, quadratic...) of the objective function $\mathcal{L}$ and minimize it in the trust region

Linear model



Quadratic model



Trust-region update

- Acceptance according to the ratio between the *actual reduction* and the *predicted reduction*:

$$\rho_k = \frac{\mathcal{L}(\mathbf{x}_k) - \mathcal{L}(\mathbf{x}_{k+1})}{\ell(\mathbf{x}_k) - \ell(\mathbf{x}_{k+1})}$$

- If ρ_k too small, reject the step and reduce the trust-region
- If ρ_k sufficiently big, increase the trust-region

With inverse problems, slow decrease of the size of the trust region

Unconstrained formulation: least-squares

- Nonlinear least-squares: given $F : \mathbb{R}^n \to \mathbb{R}^m$, find

$$\min_{\mathbf{x}} \frac{1}{2}\|F(\mathbf{x})\|^2.$$

- For our problem:

$$\begin{array}{rccl} F: & \mathbb{R}^p & \to & \mathbb{R}^{m+r} \\ & \theta & \mapsto & \left(u_\theta(\mathbf{x}_M) - \bar{u},\ \mathcal{N}[u_\theta, c_\theta](\mathbf{x}_R)\right) \end{array}$$

Unconstrained formulation: Levenberg-Marquardt method

- Levenberg-Marquardt method: like Gauss-Newton, plus a regularization term

$$\min_{\mathbf{p}} m_k^{LM}(\mathbf{p}) = \frac{1}{2}\|F(\mathbf{x}_k) + J(\mathbf{x}_k)\mathbf{p}\|^2 + \frac{\lambda_k}{2}\|\mathbf{p}\|^2.$$

- The minimizer $\mathbf{p}_k^{LM}(\lambda_k)$ of this model satisfies the following equation:

$$(B_k + \lambda_k I)\mathbf{p}_k^{LM}(\lambda_k) = -g_k$$

where $B_k = J(\mathbf{x}_k)^T J(\mathbf{x}_k)$ and $g_k = J(\mathbf{x}_k)^T F(\mathbf{x}_k)$.

Unconstrained formulation: Trust-region method

We modify this method into a trust-region method:

$$\begin{aligned} \min_{\mathbf{p}} \quad & \frac{1}{2}\|F(\mathbf{x}_k) + J(\mathbf{x}_k)\mathbf{p}\|^2 \\ s.t. \quad & \|\mathbf{p}\| \leq \Delta_k \end{aligned} \tag{1}$$

Lemma

A vector $\mathbf{p}$ *is a solution of the trust-region subproblem 1 if and only if* $\mathbf{p}$ *is feasible and there exists a scalar* $\lambda_k \geq 0$ *such that*

$$(B_k + \lambda_k I)\mathbf{p} = -g_k$$
$$\lambda_k(\Delta_k - \|\mathbf{p}\|) = 0.$$

Constrained formulation

- We want to solve a constrained problem

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) \qquad s.t. \quad g(\mathbf{x}) \leq \delta$$

- Typically we would take $f = \mathcal{L}_{\mathrm{R}}$ and $g = \mathcal{L}_{\mathrm{M}}$ (to avoid overfitting of the noise)
- We transform the constraint into a penalization term:

$$\min_{\mathbf{x}} \quad \Phi(\mathbf{x}) \triangleq f(\mathbf{x}) + \nu \max\{g(\mathbf{x}) - \delta, 0\}$$

Constrained formulation: Sequential Linear Programming

- We linearize the objective function to solve it sequentially with a linear model:

$$\begin{aligned} \min_{d_k} \quad & f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k) \cdot d_k + \nu_k \max\{g(\mathbf{x}_k) + \nabla g(\mathbf{x}_k) \cdot d_k - \delta, 0\} \\ s.t. \quad & \|d_k\|_\infty \leq \Delta_k \end{aligned}$$

- With a linear program:

$$\begin{aligned} \min_{d_k, t} \quad & f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k) \cdot d_k + \nu_k t \\ s.t. \quad & \|d_k\|_\infty \leq \Delta_k \\ & t \geq 0 \\ & t \geq g(\mathbf{x}_k) + \nabla g(\mathbf{x}_k) \cdot d_k - \delta \end{aligned}$$

- ν_k dynamically set with the Lagrange multipliers

Thank you!

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019).
Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.
Journal of Computational physics, 378:686–707.

Rudy, S. H., Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2017).
Data-driven discovery of partial differential equations.
Science advances, 3(4):e1602614.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. (2020).
Implicit neural representations with periodic activation functions.
Advances in Neural Information Processing Systems, 33:7462–7473.

Finding the good λ_k

First solve the Gauss-Newton solution $B_k\mathbf{p} = -g_k$.

- if $\|\mathbf{p}_k^{GN}\| < \Delta_k$, it solve the TR subproblem
- otherwise, find the λ_k such that $\|\mathbf{p}_k^{LM}(\lambda_k)\| = \Delta_k$, i.e. $\frac{1}{\|\mathbf{p}_k^{LM}(\lambda_k)\|} - \frac{1}{\Delta_k} = 0$, iteratively solved with Newton's method.

Help from finite differences?

- Introduce a new term:

$$\mathcal{L}_{\mathrm{FD}} = \frac{1}{n_M} \left\|(-L + \operatorname{diag} c_\theta(\mathbf{x}_M))\bar{u} - \bar{\varphi}\right\|^2$$

  where L is the matrix of the discretized laplacian
- Too much error introduced due to the discretization, even without noise