Propagation d'incertitude et analyse de sensibilité en dynamique des fluides numérique

Camilla Fiorini

le cnam

Séminaire M2N
14 décembre 2021

Parcours

2014—2018 **Thèse en Mathématiques Appliquées, *Université de Versailles.***

Analyse de sensibilité pour des équations hyperboliques non linéaires.

2018—2020 **Post-doc, *Sorbonne Université*.**

Propagation d'incertitude pour les équations de Navier-Stokes.

2020—2021 **Post-doc, *Inria*, Rennes.**

Transport stochastique dans les courants océaniques surfaciques (ERC STUOD).

Parcours

2018—2020 **Post-doc, *Sorbonne Université*.**

Propagation d'incertitude pour les équations de Navier-Stokes.

2014—2018 **Thèse en Mathématiques Appliquées, *Université de Versailles.***

Analyse de sensibilité pour des équations hyperboliques non linéaires.

2020—2021 **Post-doc, *Inria*, Rennes.**

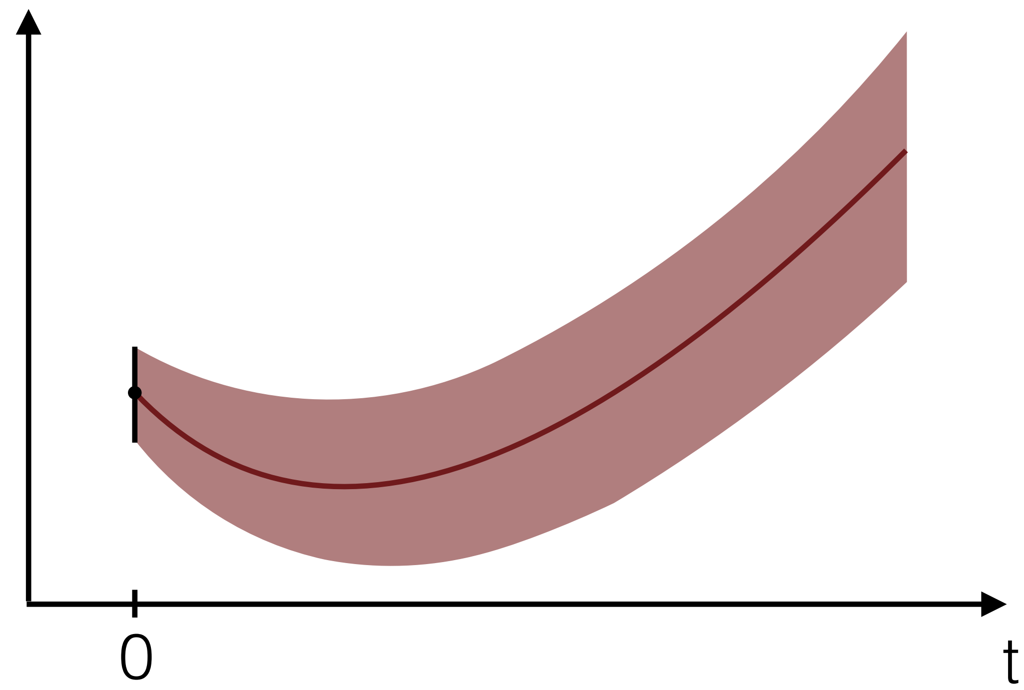
Transport stochastique dans les courants océaniques surfaciques (ERC STUOD).

Propagation d'incertitude pour les équations de Navier—Stokes

Aim and context

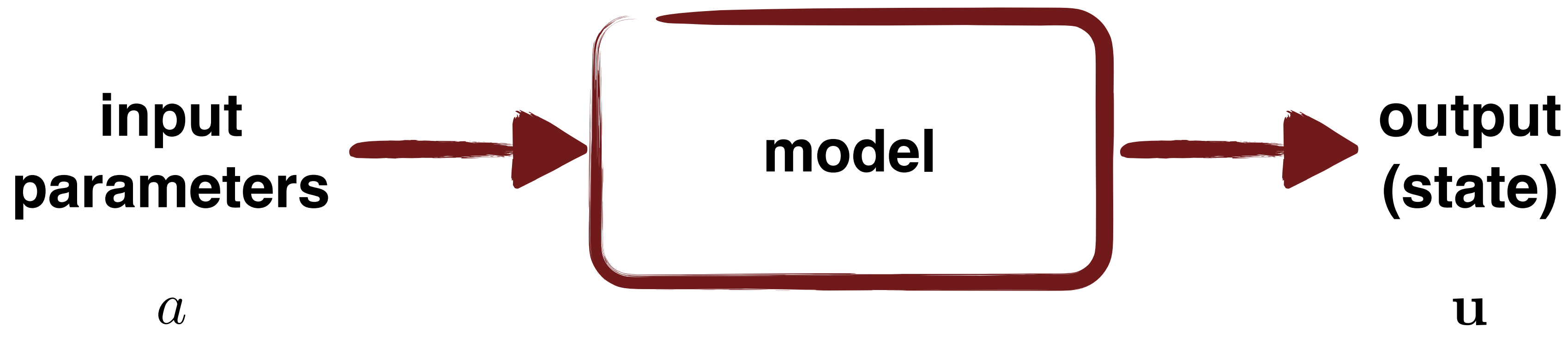
AIM: develop an **uncertainty propagation tool** and integrate it into the TRUST code.

uncertain initial condition



Ideal properties:	**Sensitivity Equation Method**
▸ computationally affordable	✓
▸ requires minimal code development	✓ / ✗

Sensitivity analysis (SA) : study of how changes in the **inputs** of a model affect the **outputs**



Sensitivity :

$$\frac{\partial \mathbf{u}}{\partial a} = \mathbf{u}_a$$

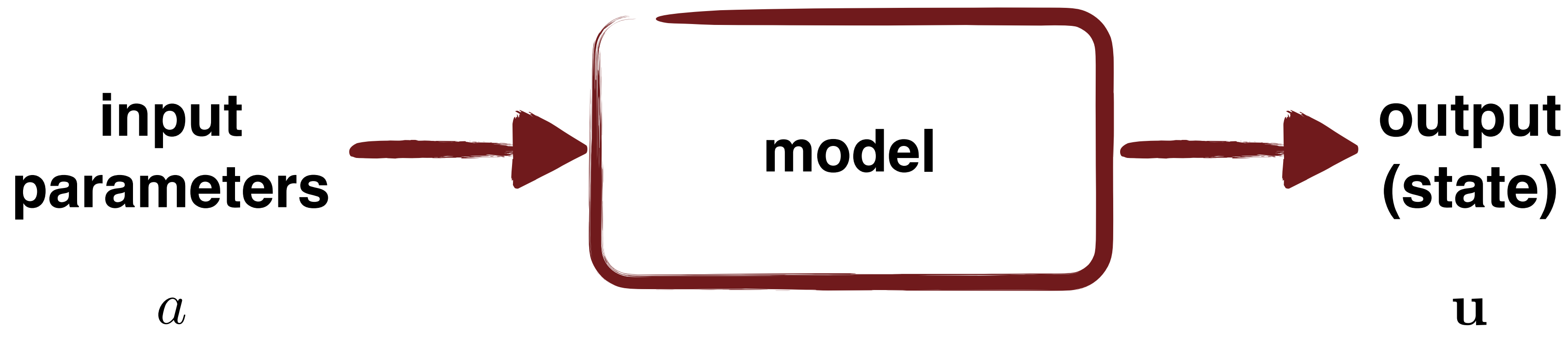
Continuous sensitivity equation (CSE) method :

$$\partial_t \mathbf{u} + \mathcal{L}(\mathbf{u}) = \mathbf{f} \quad \Omega,\ t > 0$$

+ initial and boundary conditions.

Sensitivity Analysis

Sensitivity analysis (SA) : study of how changes in the **inputs** of a model affect the **outputs**



Sensitivity :

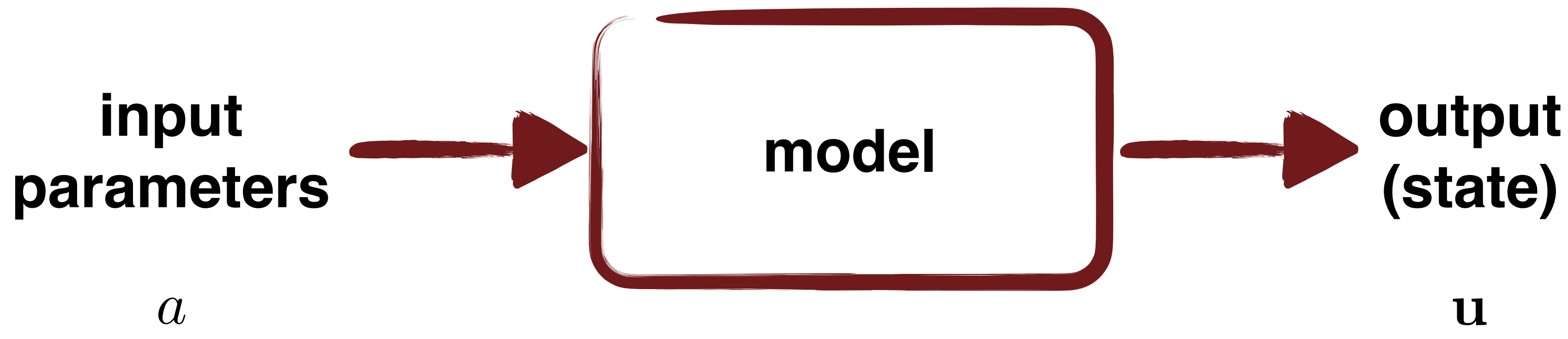
$$\frac{\partial \mathbf{u}}{\partial a} = \mathbf{u}_a$$

Continuous sensitivity equation (CSE) method :

$$\partial_a \partial_t \mathbf{u} + \partial_a \mathcal{L}(\mathbf{u}) = \partial_a \mathbf{f} \quad \Omega,\ t > 0$$

+ initial and boundary conditions.

Sensitivity analysis (SA) : study of how changes in the **inputs** of a model affect the **outputs**



Sensitivity :

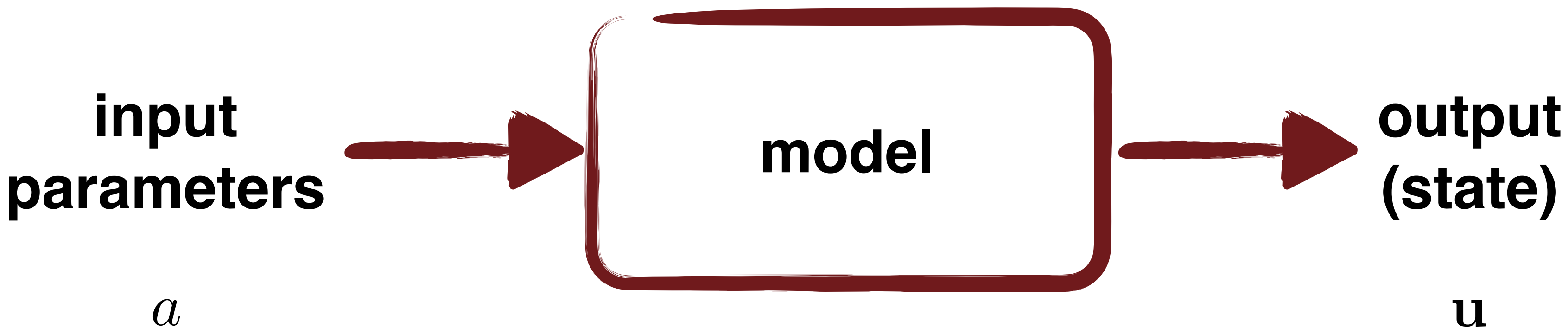
$$\frac{\partial \mathbf{u}}{\partial a} = \mathbf{u}_a$$

Continuous sensitivity equation (CSE) method :

$$\partial_a \partial_t \mathbf{u} + \partial_a \mathcal{L}(\mathbf{u}) = \partial_a \mathbf{f} \quad \Omega, \ t > 0$$

+ initial and boundary conditions.

Sensitivity analysis (SA) : study of how changes in the **inputs** of a model affect the **outputs**



Sensitivity :

$$\frac{\partial \mathbf{u}}{\partial a} = \mathbf{u}_a$$

Continuous sensitivity equation (CSE) method :

$$\partial_t \mathbf{u}_a + \mathcal{L}(\mathbf{u}, \mathbf{u}_a) = \mathbf{f}_a \quad \Omega,\ t > 0$$

+ initial and boundary conditions.

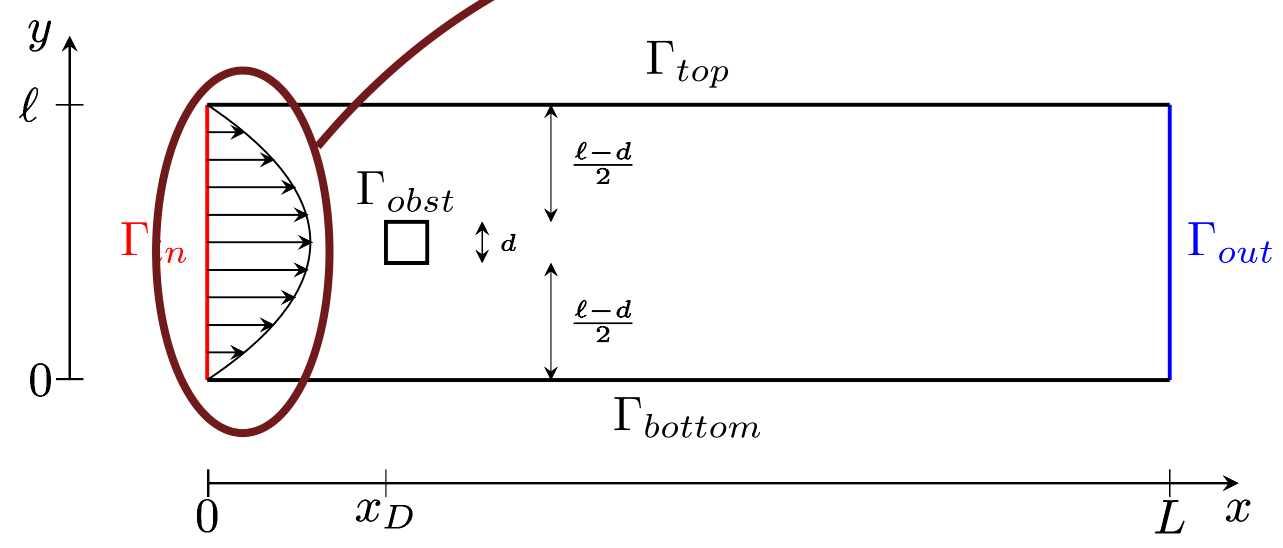
State and sensitivity equations

The Navier—Stokes equations :

$$
\begin{cases}
\partial_t \mathbf{u} - \nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f} & \Omega, t>0,\\
\nabla\cdot\mathbf{u} = 0 & \Omega, t>0,\\
\mathbf{u}(\mathbf{x},0) = 0 & \Omega, t=0,\\
\mathbf{u} = -g(y)\mathbf{n} & \text{on } \Gamma_{in},\\
\mathbf{u} = 0 & \text{on } \Gamma_{w},\\
(\nu\nabla\mathbf{u} - pI)\mathbf{n} = 0 & \text{on } \Gamma_{out}.
\end{cases}
$$

Domain :

$$g(y) = \frac{4A}{\ell^2} y(\ell - y)$$



$$\Gamma_w = \Gamma_{obst} \cup \Gamma_{top} \cup \Gamma_{bottom}$$

The sensitivity equations :

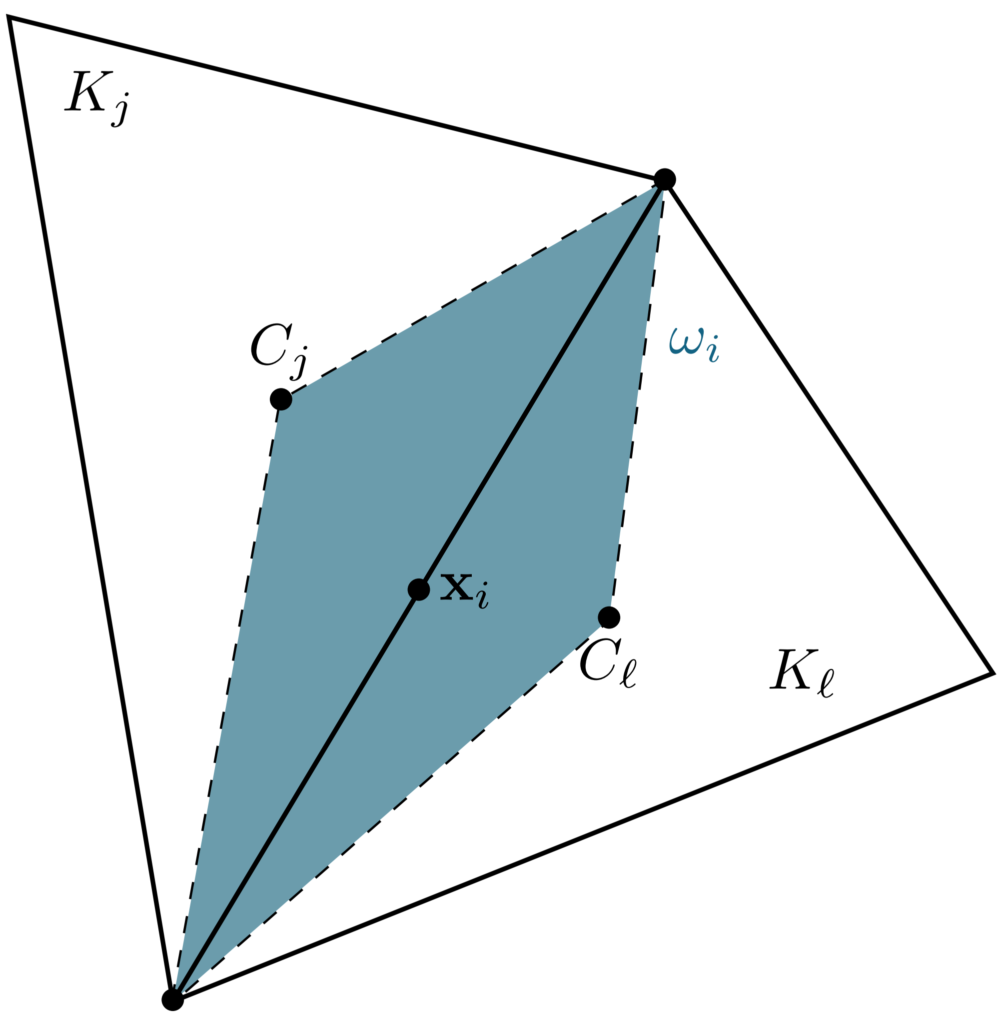
$$
\begin{cases}
\partial_t \mathbf{u}_a - \nu\Delta\mathbf{u}_a + (\mathbf{u}_a\cdot\nabla)\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u}_a + \nabla p_a = \bar{\mathbf{f}}_a & \Omega, t>0,\\
\nabla\cdot\mathbf{u}_a = 0 & \Omega, t>0,\\
\mathbf{u}_a(\mathbf{x},0) = 0 & \Omega, t=0,\\
\mathbf{u}_a = -g_a(y)\mathbf{n} & \text{on } \Gamma_{in},\\
\mathbf{u}_a = 0 & \text{on } \Gamma_{w},\\
(\nu\nabla\mathbf{u}_a - p_a I)\mathbf{n} = 0 & \text{on } \Gamma_{out}.
\end{cases}
$$

$$\bar{\mathbf{f}}_a = \partial_a \mathbf{f} + \partial_a \nu \Delta \mathbf{u}$$

Remark : these are known as the Oseen equations.

Spatial discretisation

The code TRUST TrioCFD is based on a **finite elements volumes** method (FEV).



Ingredients:

$\mathcal{T}_h$ triangulation of the domain Ω

$K_j \in \mathcal{T}_h$ triangles $j = 1, \dots, N_T$

$\mathbf{x}_i$ nodes $i = 1, \dots, N_N$

ω_i control volume

Spaces:

$$Q_h = \{q_h : \forall K \in \mathcal{T}_h,\ q_h|_K \in P_0(K)\},$$
$$V_h = \{w_h \text{ continuous in } \mathbf{x}_i : \forall K \in \mathcal{T}_h,\ w_h|_K \in P_1(K)\},$$
$$\mathbf{V}_h = \{\mathbf{w}_h = (w_x, w_y)^t : w_x,\ w_y \in V_h\}.$$

Basis functions : $\varphi_i(\mathbf{x}_j) = \delta_{i,j}$ for V_h

χ_K for Q_h

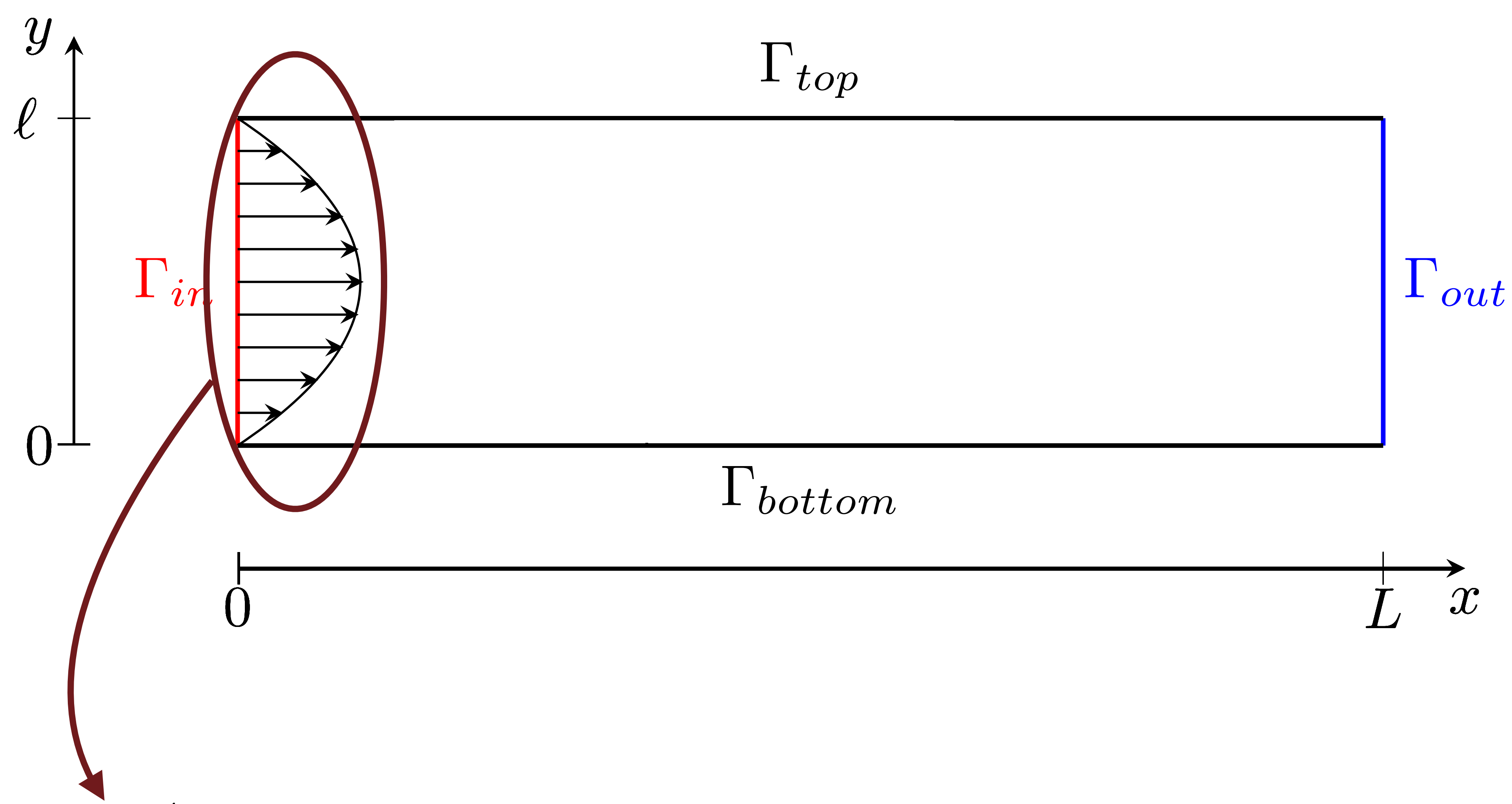
Remark : $V_h \not\subset H^1(\Omega)$

Spatial discretisation

$$
\begin{aligned}
A\mathbf{U} + B^t P + L(\mathbf{U})\mathbf{U} &= \mathbf{F} \\
B\mathbf{U} &= D \\
A\mathbf{U}_a + B^t P_a + L(\mathbf{U}_a)\mathbf{U} + L(\mathbf{U})\mathbf{U}_a &= \mathbf{F}_a \\
B\mathbf{U}_a &= D_a
\end{aligned}
$$

Test case description

Domain :



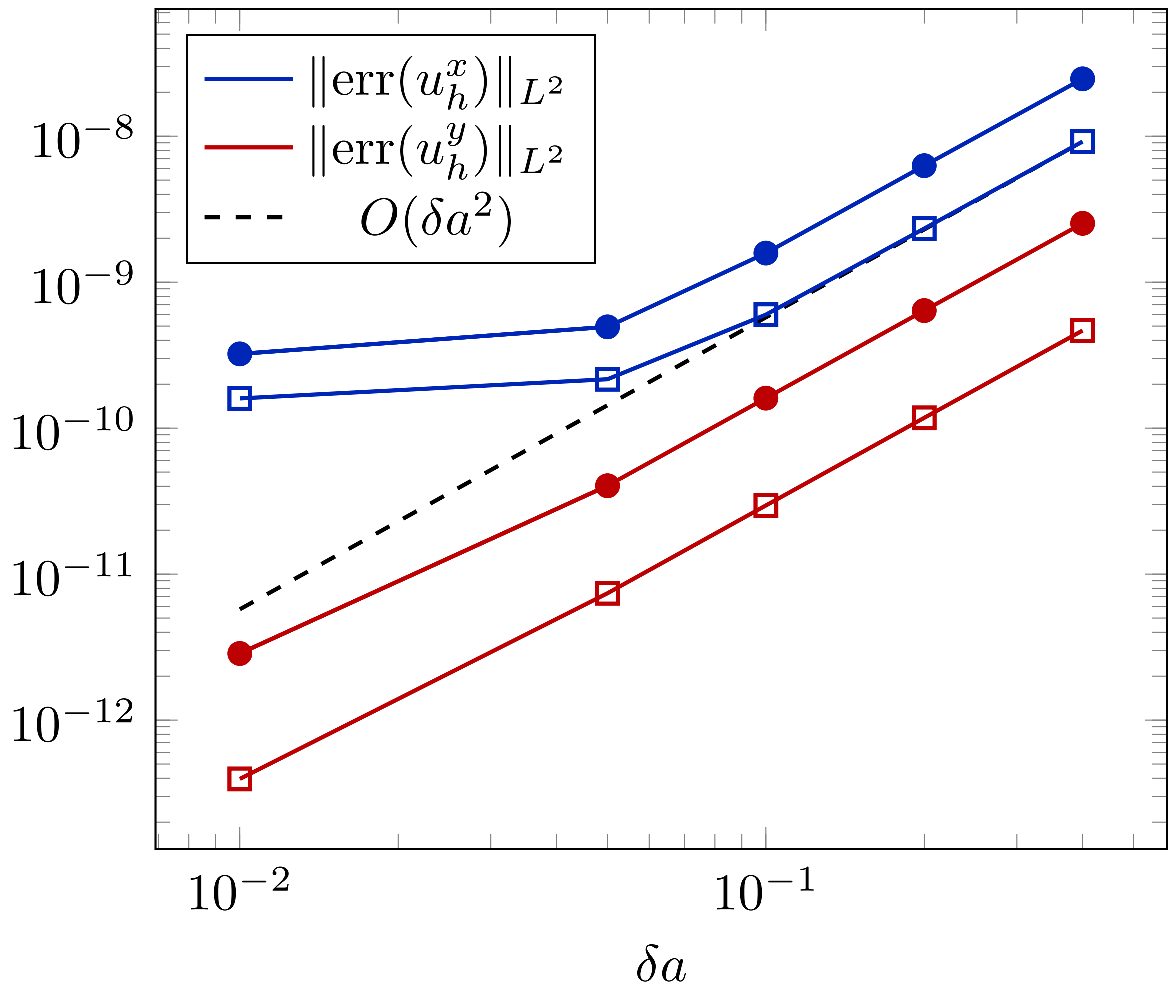
$$g(y) = \frac{4A}{\ell^2} y(\ell - y)$$

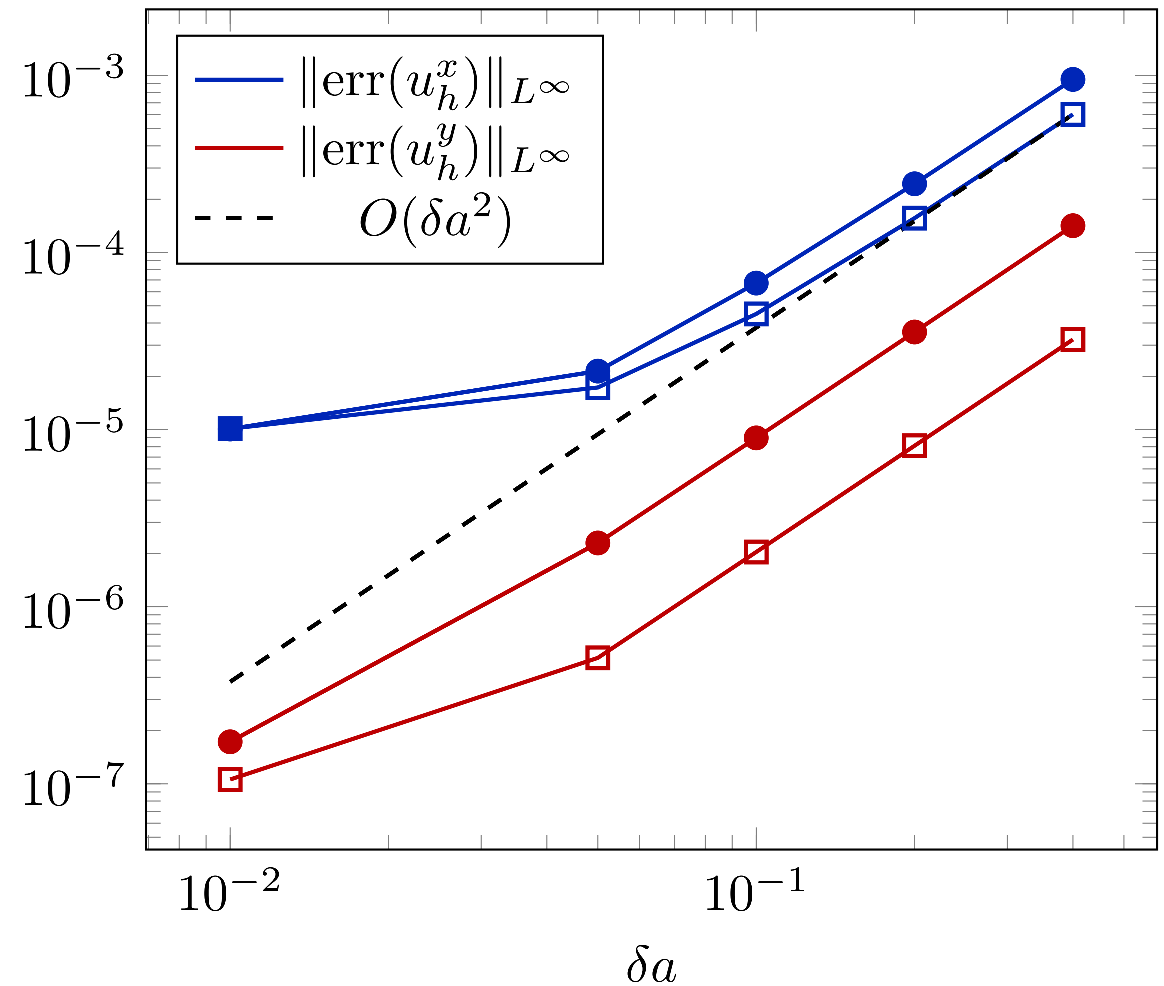
Uncertain parameter $a = A$

Validation of the CSE method

Results

$$\mathrm{err}(\mathbf{u}) = \mathbf{u}(x,T;a+\delta a) - \mathbf{u}(x,T;a) - \delta a \mathbf{u}_a(x,T;a) \simeq O(\delta a^2)$$





The symbol ▫ corresponds to a finer mesh (h=0.001), the symbol • to the coarser one (h=0.002).

Uncertainty quantification

Let **a** be a gaussian random vector, with the following average and variance:

$$\mu_{\mathbf{a}} = \begin{bmatrix} \mu_{a_1} \\ \vdots \\ \mu_{a_M} \end{bmatrix}, \qquad \sigma_{\mathbf{a}} = \begin{bmatrix} \sigma_{a_1}^2 & \mathrm{cov}(a_1, a_2) & \dots & \mathrm{cov}(a_1, a_M) \\ \mathrm{cov}(a_1, a_2) & \sigma_{a_2}^2 & \dots & \mathrm{cov}(a_2, a_M) \\ \vdots & & \ddots & \vdots \\ \mathrm{cov}(a_1, a_M) & \dots & & \sigma_{a_M}^2 \end{bmatrix},$$

Aim: determine a **confidence interval**

$$CI_X = [\mu_X - d(\sigma_X), \mu_X + d(\sigma_X)]$$

$$P(X \in CI_X) \geqslant 1 - \alpha$$

Monte Carlo approach: *N* samples of the state X_k

$$\mu_X = \frac{1}{N} \sum_{k=1}^{N} X_k \qquad \sigma_X^2 = \frac{1}{N-1} \sum_{k=1}^{N} |\mu_X - X_k|^2$$

Uncertainty quantification

Sensitivity approach: let us consider the following first order Taylor expansion

$$X(\mathbf{a}) = X(\mu_{\mathbf{a}}) + \sum_{i=1}^{M} (a_i - \mu_{a_i}) X_{a_i}(\mu_{\mathbf{a}}) + o(\|\mathbf{a} - \mu_{\mathbf{a}}\|^2)$$

Then, computing, the average one has:

$$\mu_X = E[X(\mathbf{a})] = X(\mu_{\mathbf{a}}) + \sum_{i=1}^{M} X_{a_i}(\mu_{\mathbf{a}}) E[a_i - \mu_{a_i}] = X(\mu_{\mathbf{a}}),$$

And for the variance:

$$\sigma_X^2 = E[(X(\mathbf{a}) - \mu_X)^2] = E\left[\left(\sum_{i=1}^{M} X_{a_i}(\mu_{\mathbf{a}})(a_i - \mu_{a_i})\right)^2\right] =$$

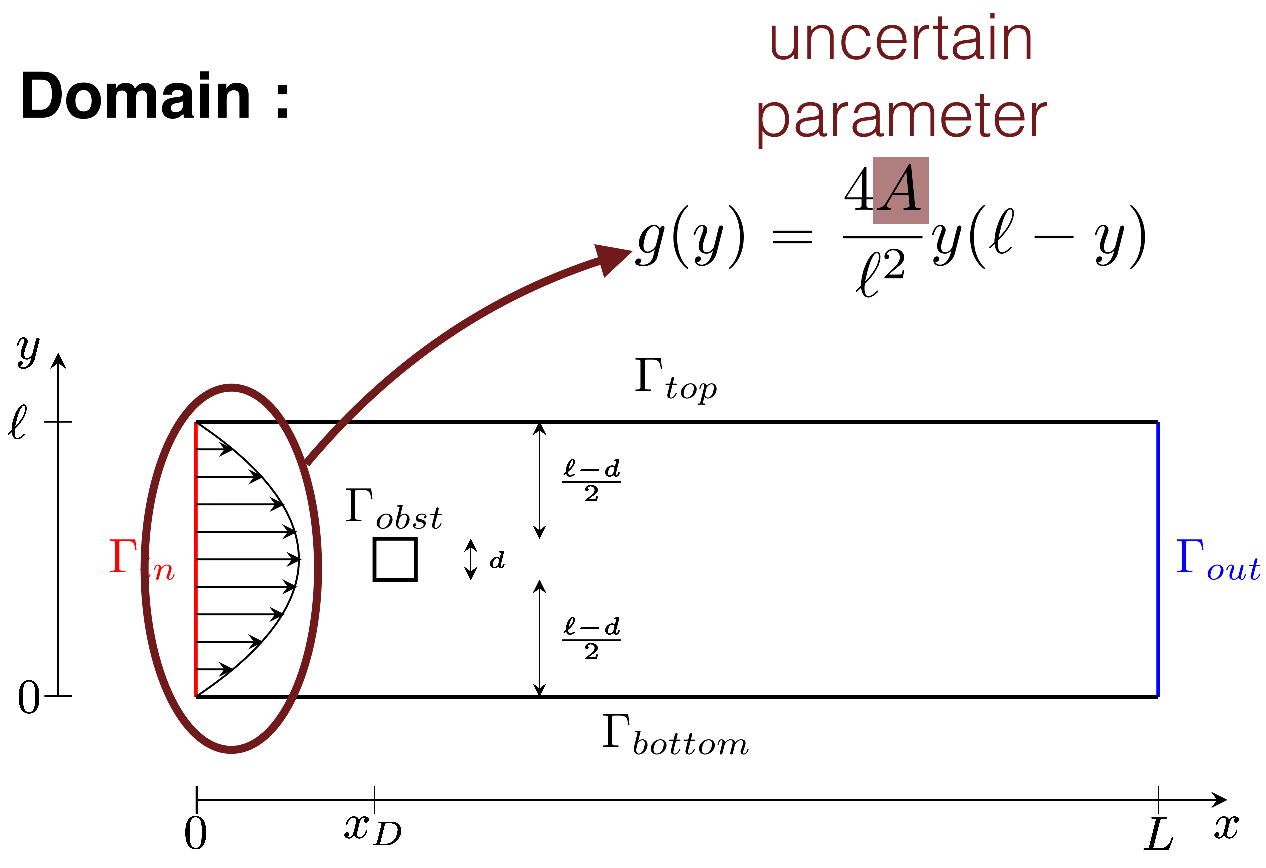
$$= \sum_{i=1}^{M} X_{a_i}^2(\mu_{\mathbf{a}}) E[(a_i - \mu_{a_i})^2] + \sum_{\substack{i,j=1 \\ i \neq j}}^{M} X_{a_i}(\mu_{\mathbf{a}}) X_{a_j}(\mu_{\mathbf{a}}) E[(a_i - \mu_{a_i})(a_j - \mu_{a_j})].$$

Therefore, we have the following first order estimates:

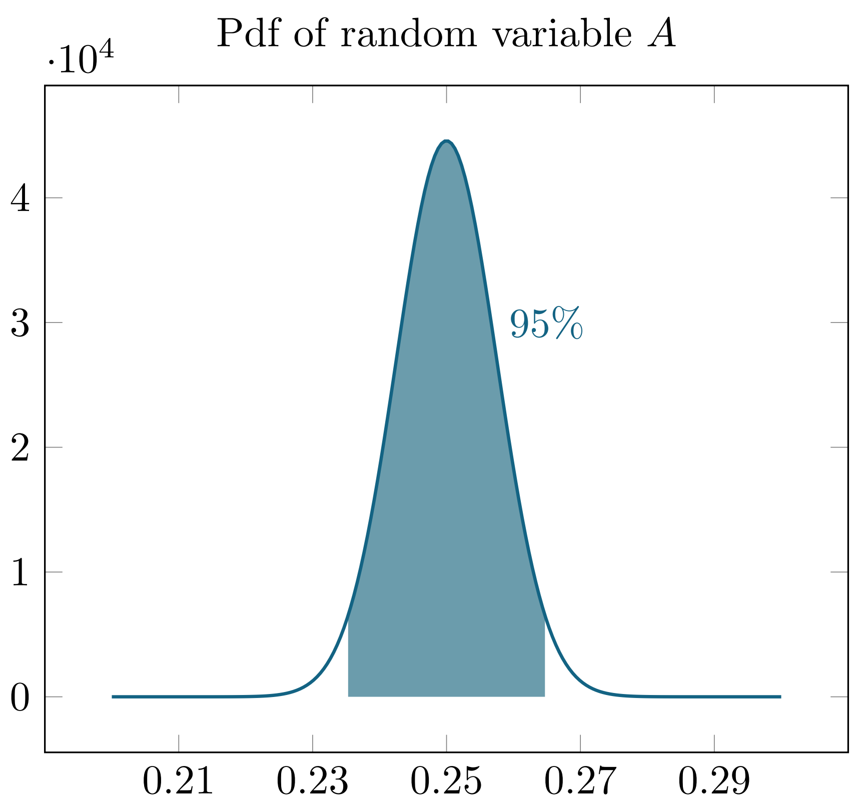
$$\mu_X = X(\mu_{\mathbf{a}}), \quad \sigma_X^2 = \sum_{i=1}^{M} X_{a_i}^2 \sigma_{a_i}^2 + \sum_{\substack{i,j=1 \\ i \neq j}}^{M} X_{a_i} X_{a_j} \mathrm{cov}(a_i, a_j).$$

Steady test case

Domain :



$$\Gamma_w = \Gamma_{obst} \cup \Gamma_{top} \cup \Gamma_{bottom}$$

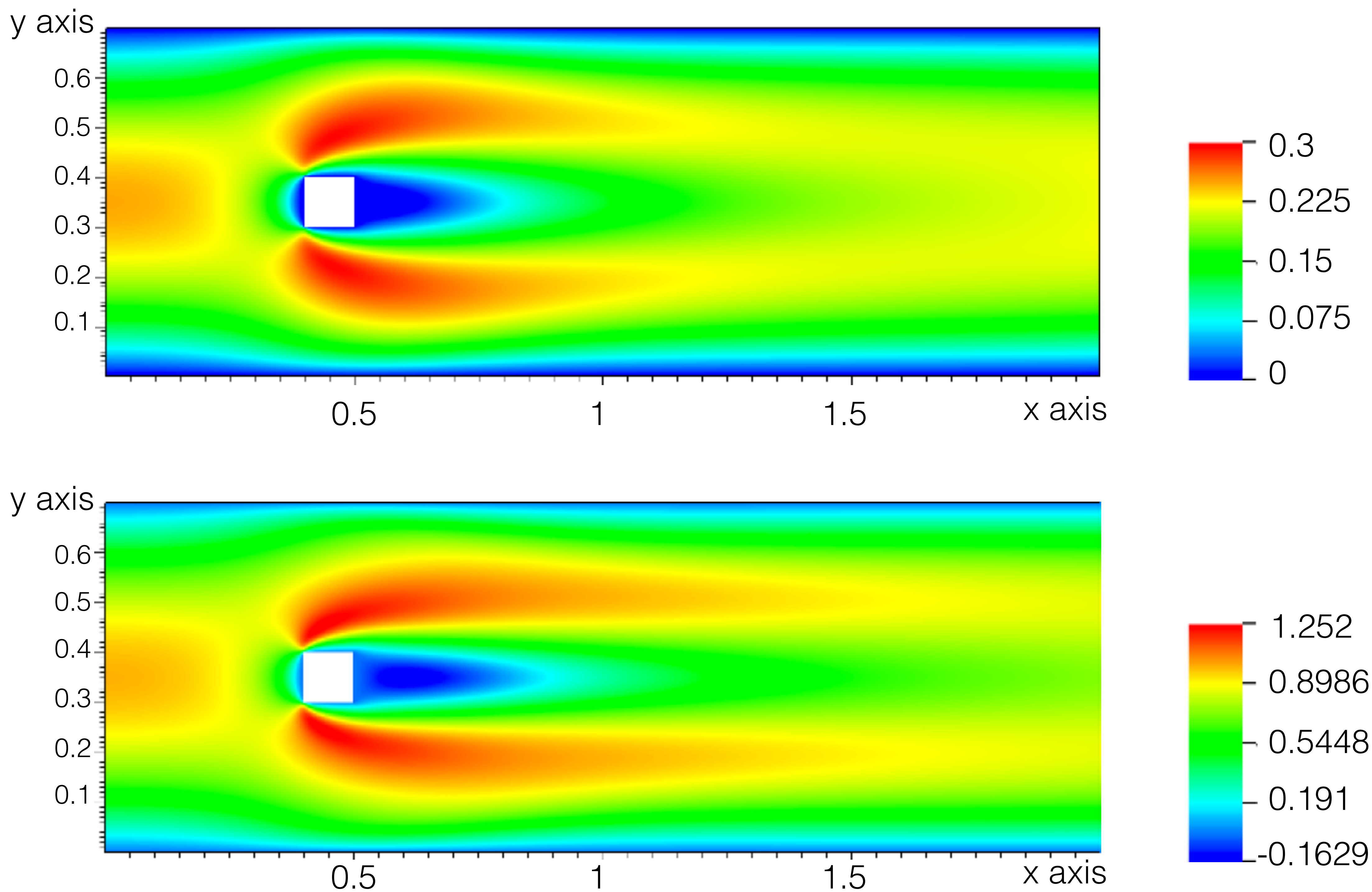


Steady test case

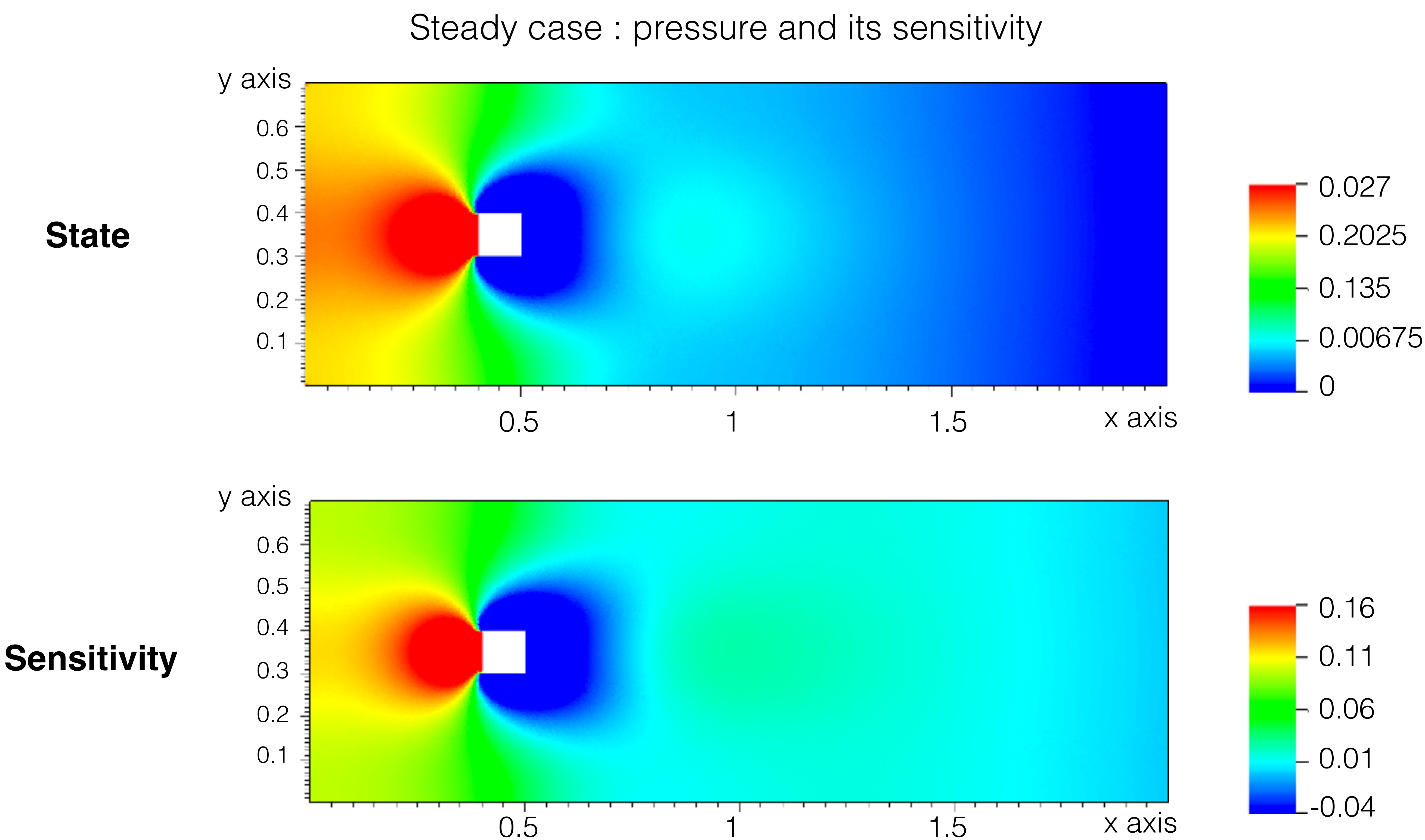
Steady case : x-component of the velocity and its sensitivity

State

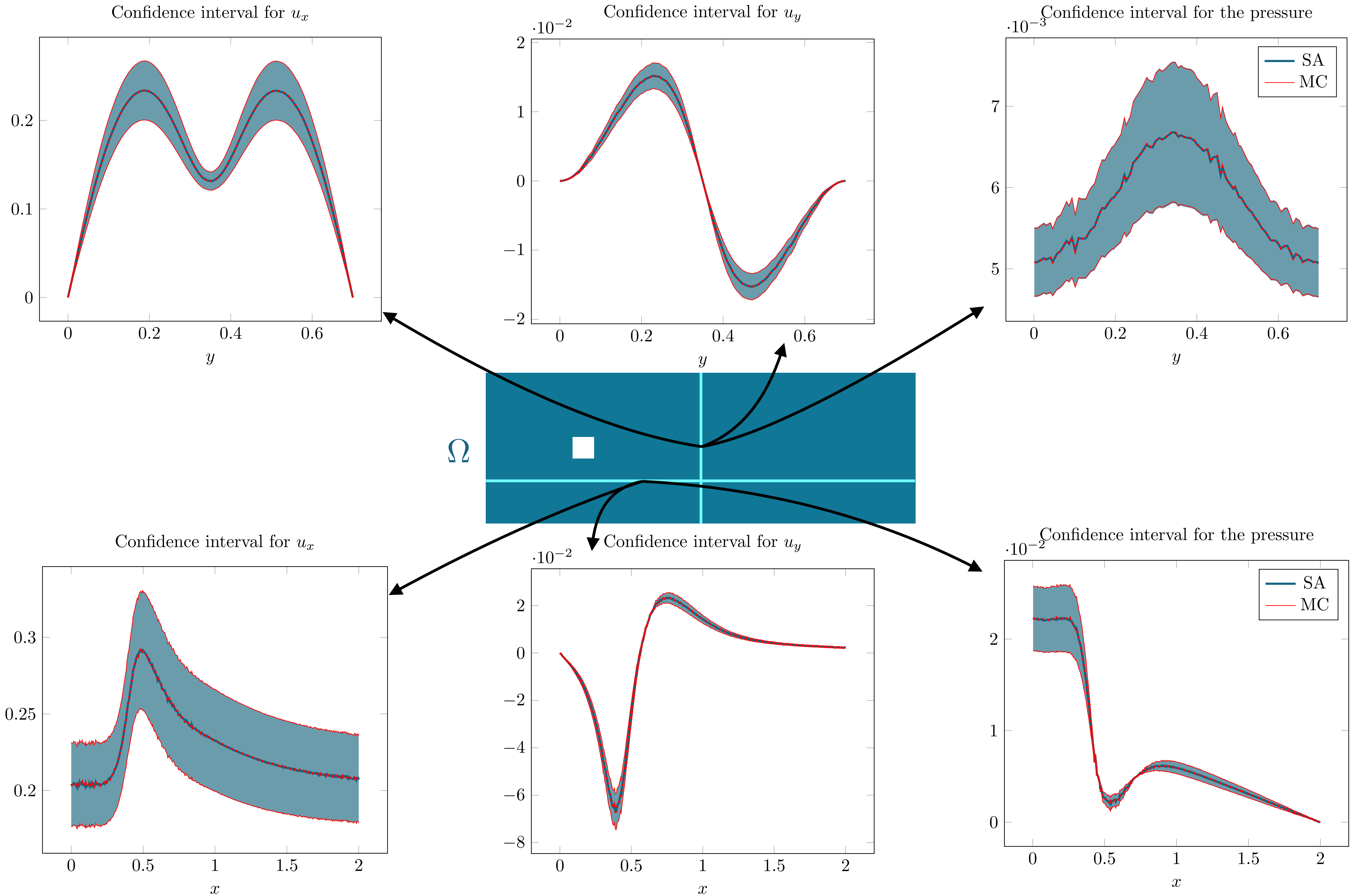
Sensitivity



Steady test case

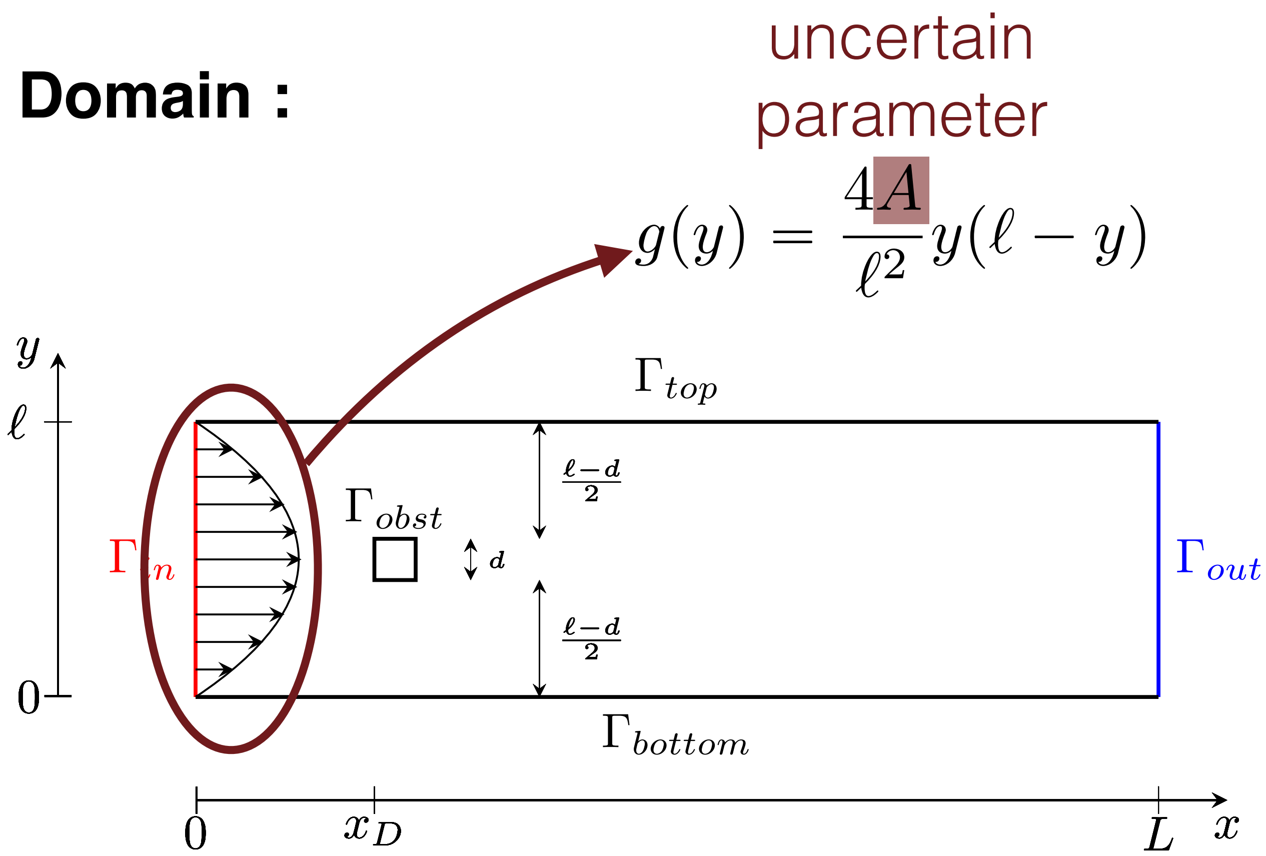


Steady test case

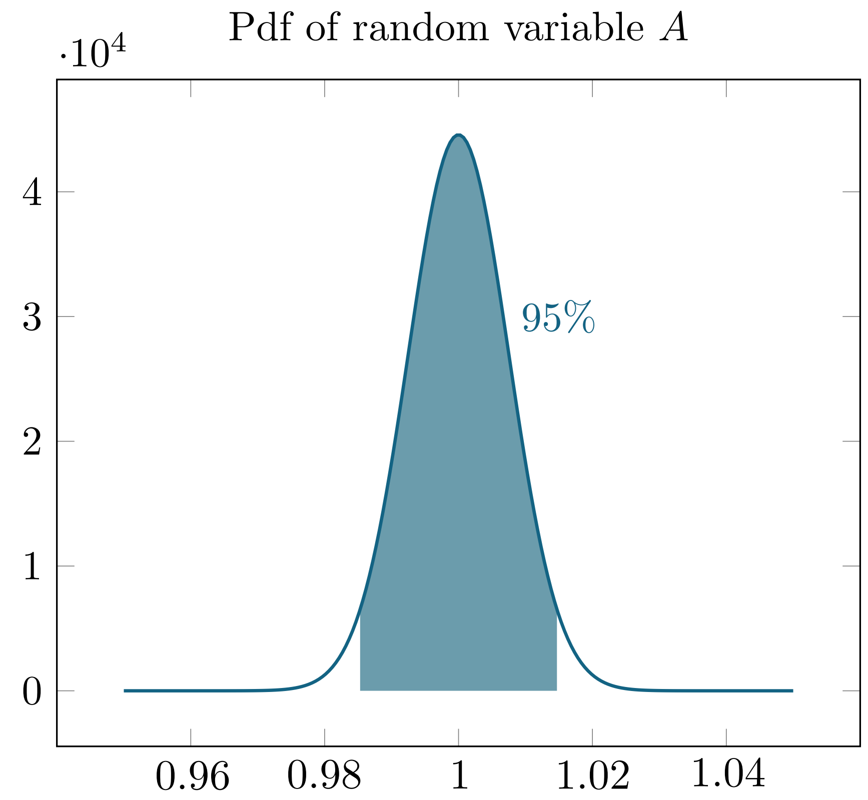


Unsteady test case

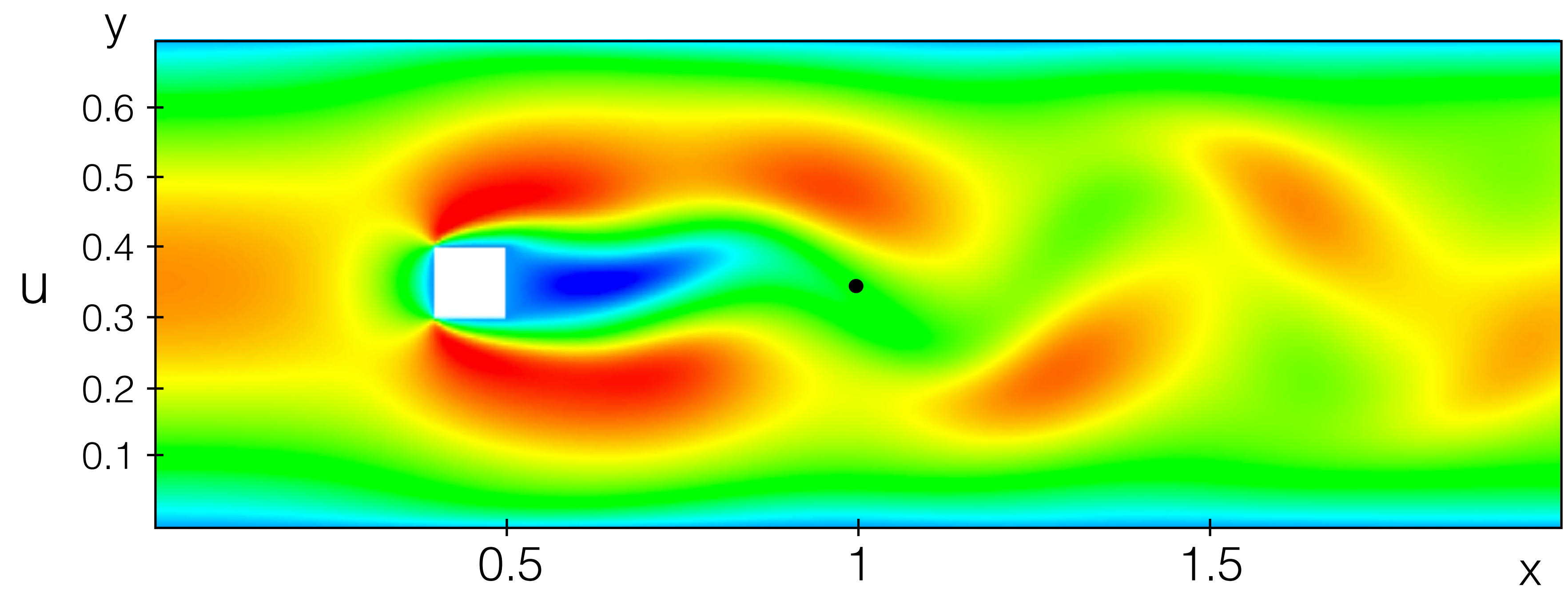
Domain :

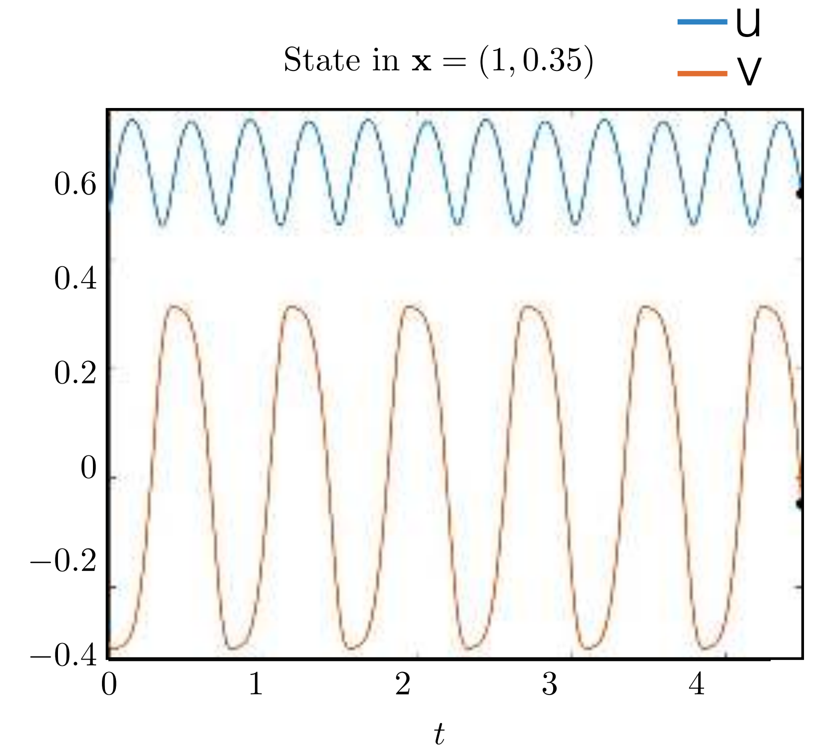


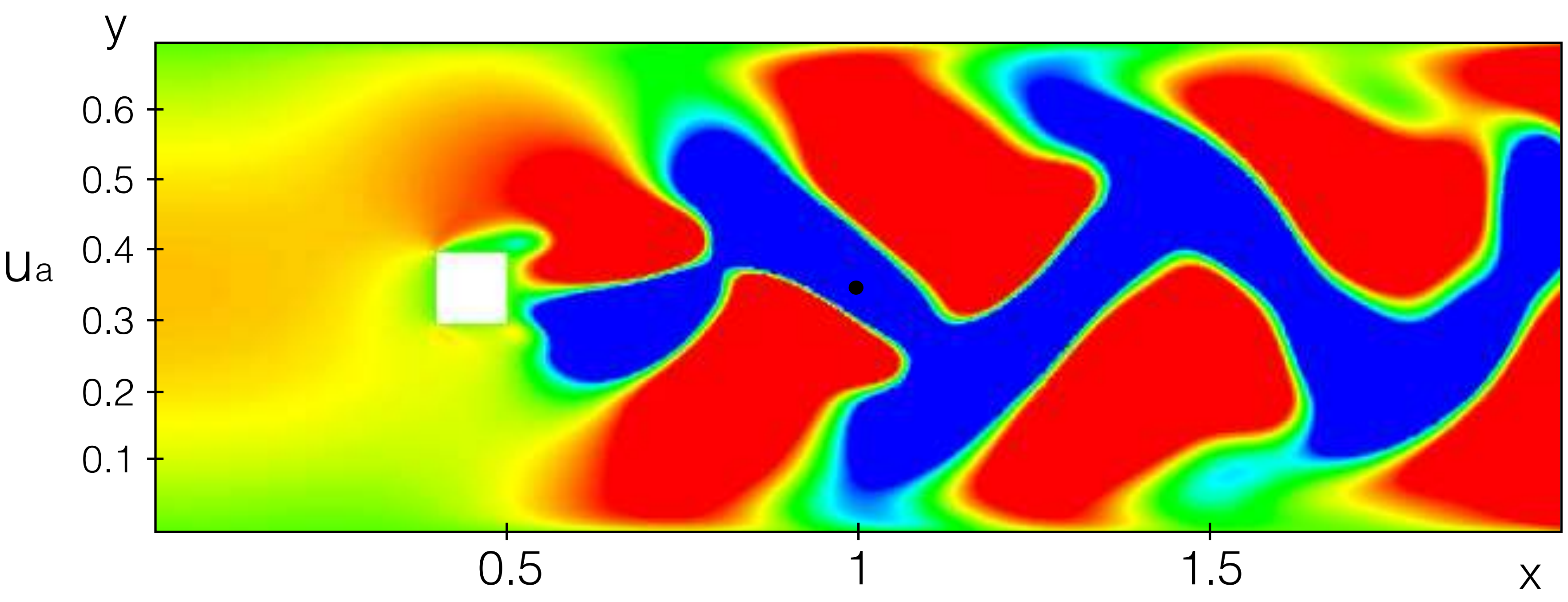
$$\Gamma_w = \Gamma_{obst} \cup \Gamma_{top} \cup \Gamma_{bottom}$$

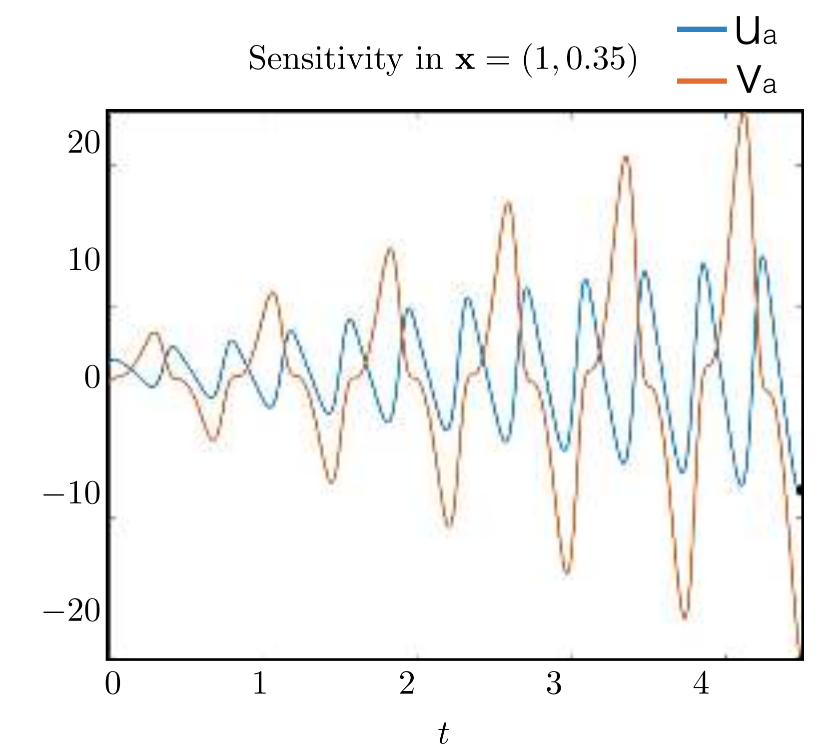


Unsteady test case









Filtering procedure

It is reasonable to assume that the velocity behaves as follows:

$$\mathbf{u}(\mathbf{x},t;a) \simeq \sum_{k=0}^{N} \mathbf{u}_{0,k}(\mathbf{x};a)\cos(\omega_k(a)t).$$

By differentiating this with respect to the parameter of interest, one obtains the following behaviour for the sensitivity:

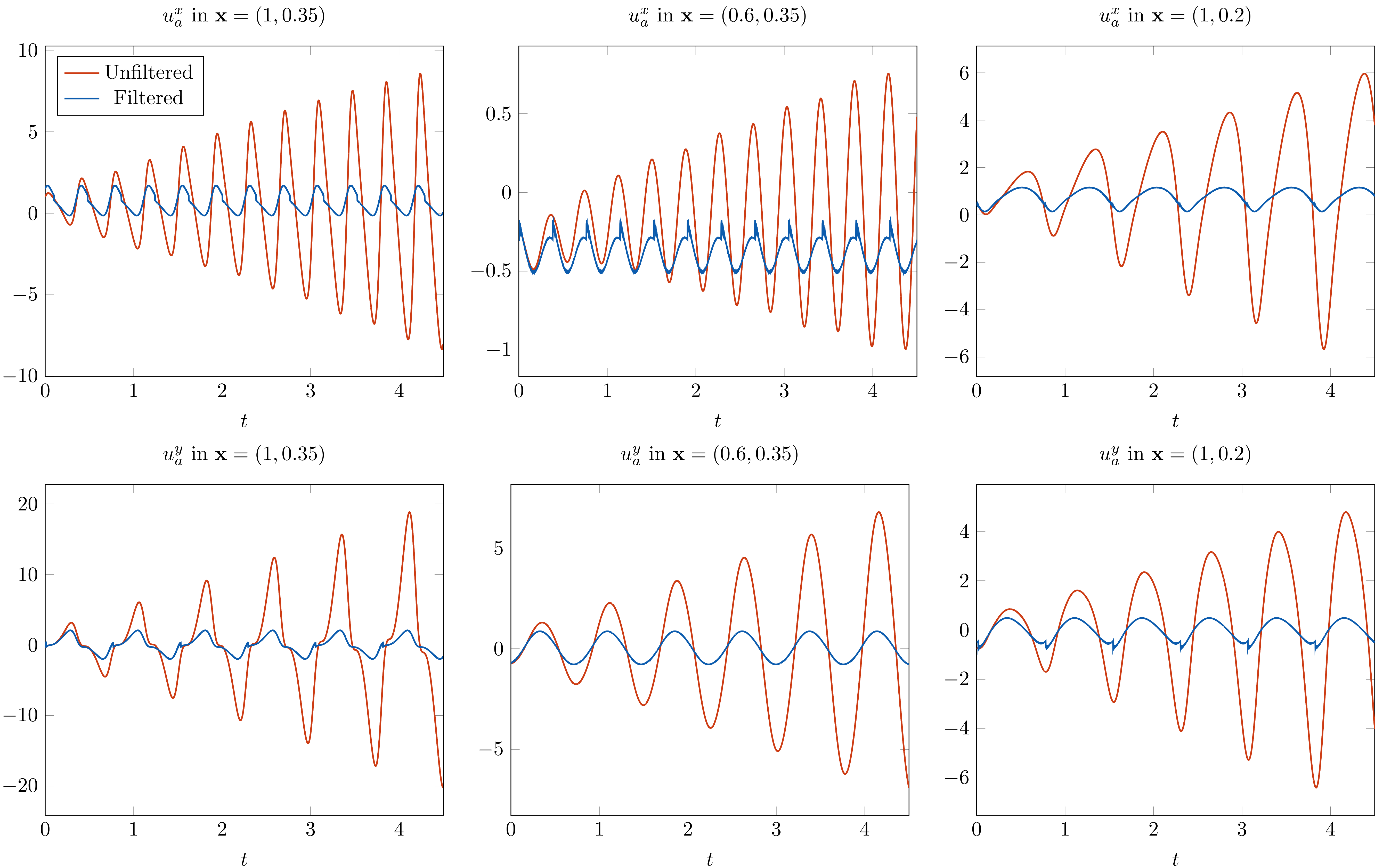
Bornée — Non bornée

$$\mathbf{u}_a(\mathbf{x},t;a) = \underbrace{\sum_{k=0}^{N} \boxed{\mathbf{u}_{0,a,k}(\mathbf{x};a)}\cos(\omega_k(a)t)}_{\text{Bornée}} - \underbrace{t\sum_{k=0}^{N} \mathbf{u}_{0,k}(\mathbf{x};a)\boxed{\omega_k'(a)}\sin(\omega_k(a)t)}_{\text{Non bornée}}$$

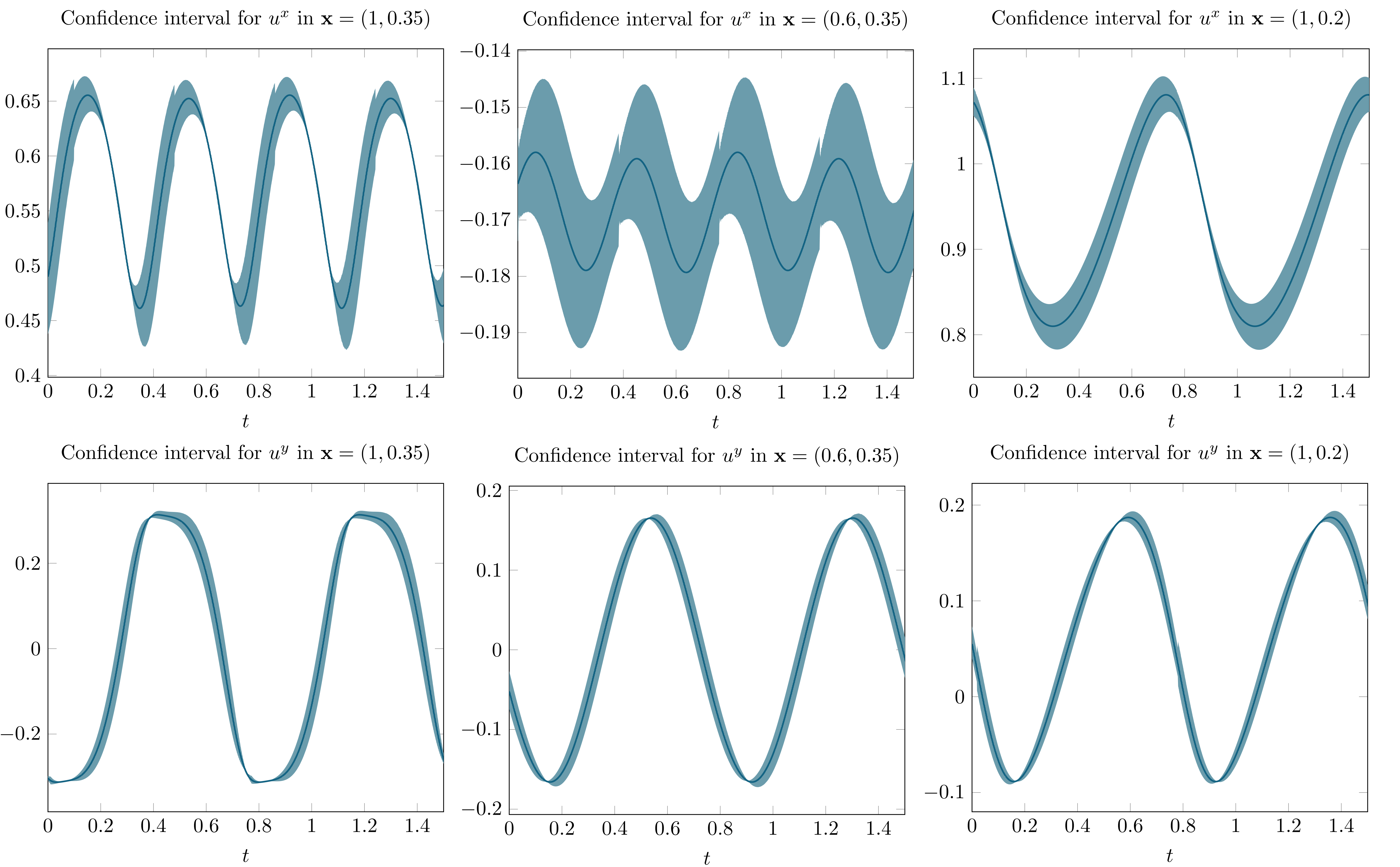
sensibilité de l'amplitude — **sensibilité de la fréquence**

We propose a filter to recover the first part of the sensitivity.

Filtering procedure



Confidence intervals with filtered sensitivities



Analyse de sensibilité pour des équations hyperboliques non linéaires

Standard techniques of sensitivity analysis

Standard techniques of sensitivity analysis call for the differentiation of the state system:

$$\begin{cases} \partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = 0 & \Omega \times (0,T), \\ \mathbf{U}(x,0) = \mathbf{g}(x) & \Omega, \end{cases}$$

[1] Bardos, C., Pironneau, O. (2002). A formalism for the differentiation of conservation laws. *Comptes Rendus Mathematique*, *335*(10), 839-845.

Standard techniques of sensitivity analysis

Standard techniques of sensitivity analysis call for the differentiation of the state system:

$$\begin{cases} \partial_a(\partial_t \mathbf{U}) + \partial_a(\partial_x \mathbf{F}(\mathbf{U})) = 0 & \Omega \times (0,T), \\ \partial_a \mathbf{U}(x,0) = \partial_a \mathbf{g}(x) & \Omega, \end{cases}$$

[1] Bardos, C., Pironneau, O. (2002). A formalism for the differentiation of conservation laws. *Comptes Rendus Mathematique*, *335*(10), 839-845.

Standard techniques of sensitivity analysis

Standard techniques of sensitivity analysis call for the differentiation of the state system:

$$\begin{cases} \partial_a(\partial_t \mathbf{U}) + \partial_a(\partial_x \mathbf{F}(\mathbf{U})) = 0 & \Omega \times (0, T), \\ \partial_a \mathbf{U}(x, 0) = \partial_a \mathbf{g}(x) & \Omega, \end{cases}$$

[1] Bardos, C., Pironneau, O. (2002). A formalism for the differentiation of conservation laws. *Comptes Rendus Mathematique*, *335*(10), 839-845.

Standard techniques of sensitivity analysis

Standard techniques of sensitivity analysis call for the differentiation of the state system:

$$\begin{cases} \partial_t \mathbf{U}_a + \partial_x \mathbf{F}_a(\mathbf{U}, \mathbf{U}_a) = 0 & \Omega \times (0, T), \\ \mathbf{U}_a(x, 0) = \mathbf{g}_a(x) & \Omega, \end{cases}$$

[1] Bardos, C., Pironneau, O. (2002). A formalism for the differentiation of conservation laws. *Comptes Rendus Mathematique*, *335*(10), 839-845.

Standard techniques of sensitivity analysis

Standard techniques of sensitivity analysis call for the differentiation of the state system:

$$\begin{cases} \partial_t \mathbf{U}_a + \partial_x \mathbf{F}_a(\mathbf{U}, \mathbf{U}_a) = 0 & \Omega \times (0,T), \\ \mathbf{U}_a(x,0) = \mathbf{g}_a(x) & \Omega, \end{cases}$$

For the **Burgers' equation**:

$$\mathbf{F}(\mathbf{U}) = \frac{u^2}{2} \qquad \mathbf{F}_a(\mathbf{U}, \mathbf{U}_a) = u u_a$$

This can be done under **hypotheses of regularity** of the state ***U*** [1].

If these techniques are applied to hyperbolic equations, **Dirac delta functions** will appear in the sensitivity.

[1] Bardos, C., Pironneau, O. (2002). A formalism for the differentiation of conservation laws. *Comptes Rendus Mathematique*, *335*(10), 839-845.

Definition of the source term

In order to have a sensitivity system which is valid also when the state is discontinuous, we add a correction term [2]:

$$\begin{cases} \partial_t \mathbf{U}_a + \partial_x \mathbf{F}_a(\mathbf{U}, \mathbf{U}_a) = \mathbf{S} & \Omega \times (0, T), \\ \mathbf{U}_a(x, 0) = \mathbf{g}_a(x) & \Omega, \end{cases}$$

defined as follows:

$$\mathbf{S} = \sum_{k=1}^{N_s} \boldsymbol{\rho}_k \delta(x - x_{k,s}(t)),$$

- N_s: number of discontinuities
- $x_{k,s}$: position of the k-th discontinuity
- $\boldsymbol{\rho}_k$: amplitude of the k-th correction (to be computed)

Remark: a **shock detector** is necessary to discretise such source term.

[2] Guinot, V., Delenne, C., Cappelaere, B. (2009). An approximate Riemann solver for sensitivity equations with discontinuous solutions. *Advances in Water Resources*, *32*(1), 61-77.

The Riemann problem for Euler equations

The Euler equations are:

$$\begin{cases} \partial_t \rho + \partial_x(\rho u) = 0, \\ \partial_t(\rho u) + \partial_x(\rho u^2 + p) = 0, \\ \partial_t(\rho E) + \partial_x(u(\rho E + p)) = 0, \end{cases}$$

Eigenvalues: Eigenvectors:

$$\lambda_1(\mathbf{U}) = u - c, \qquad \mathbf{r}_1(\mathbf{U}) = (1, u - c, H - uc)^t,$$
$$\lambda_2(\mathbf{U}) = u, \qquad \mathbf{r}_2(\mathbf{U}) = (1, u, \tfrac{u^2}{2})^t,$$
$$\lambda_3(\mathbf{U}) = u + c. \qquad \mathbf{r}_3(\mathbf{U}) = (1, u + c, H + uc)^t.$$

Genuinely nonlinear

The Riemann problem for Euler equations

The Euler equations are:

$$\begin{cases} \partial_t \rho + \partial_x(\rho u) = 0, \\ \partial_t(\rho u) + \partial_x(\rho u^2 + p) = 0, \\ \partial_t(\rho E) + \partial_x(u(\rho E + p)) = 0, \end{cases}$$

Eigenvalues:

$$\lambda_1(\mathbf{U}) = u - c,$$
$$\lambda_2(\mathbf{U}) = u,$$
$$\lambda_3(\mathbf{U}) = u + c.$$

Eigenvectors:

$$\mathbf{r}_1(\mathbf{U}) = (1, u - c, H - uc)^t,$$
$$\mathbf{r}_2(\mathbf{U}) = (1, u, \tfrac{u^2}{2})^t,$$
$$\mathbf{r}_3(\mathbf{U}) = (1, u + c, H + uc)^t.$$

Linearly degenerate

The Riemann problem for Euler equations

The Euler equations are:

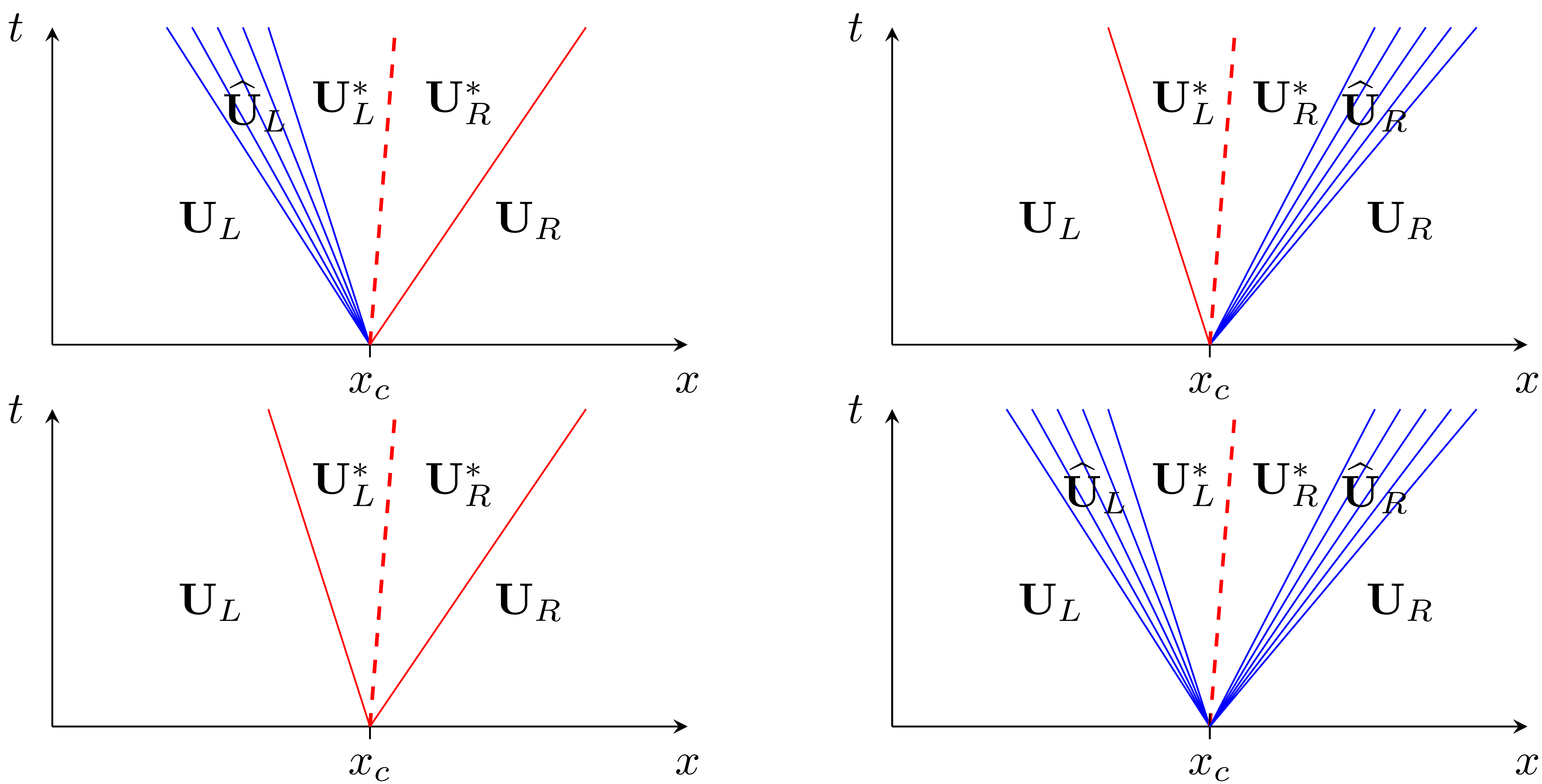
$$\begin{cases} \partial_t \rho + \partial_x(\rho u) = 0, \\ \partial_t(\rho u) + \partial_x(\rho u^2 + p) = 0, \\ \partial_t(\rho E) + \partial_x(u(\rho E + p)) = 0, \end{cases}$$

Eigenvalues:

$$\lambda_1(\mathbf{U}) = u - c, \quad \lambda_2(\mathbf{U}) = u, \quad \lambda_3(\mathbf{U}) = u + c.$$

Eigenvectors:

$$\mathbf{r}_1(\mathbf{U}) = (1, u - c, H - uc)^t, \quad \mathbf{r}_2(\mathbf{U}) = (1, u, \tfrac{u^2}{2})^t, \quad \mathbf{r}_3(\mathbf{U}) = (1, u + c, H + uc)^t.$$



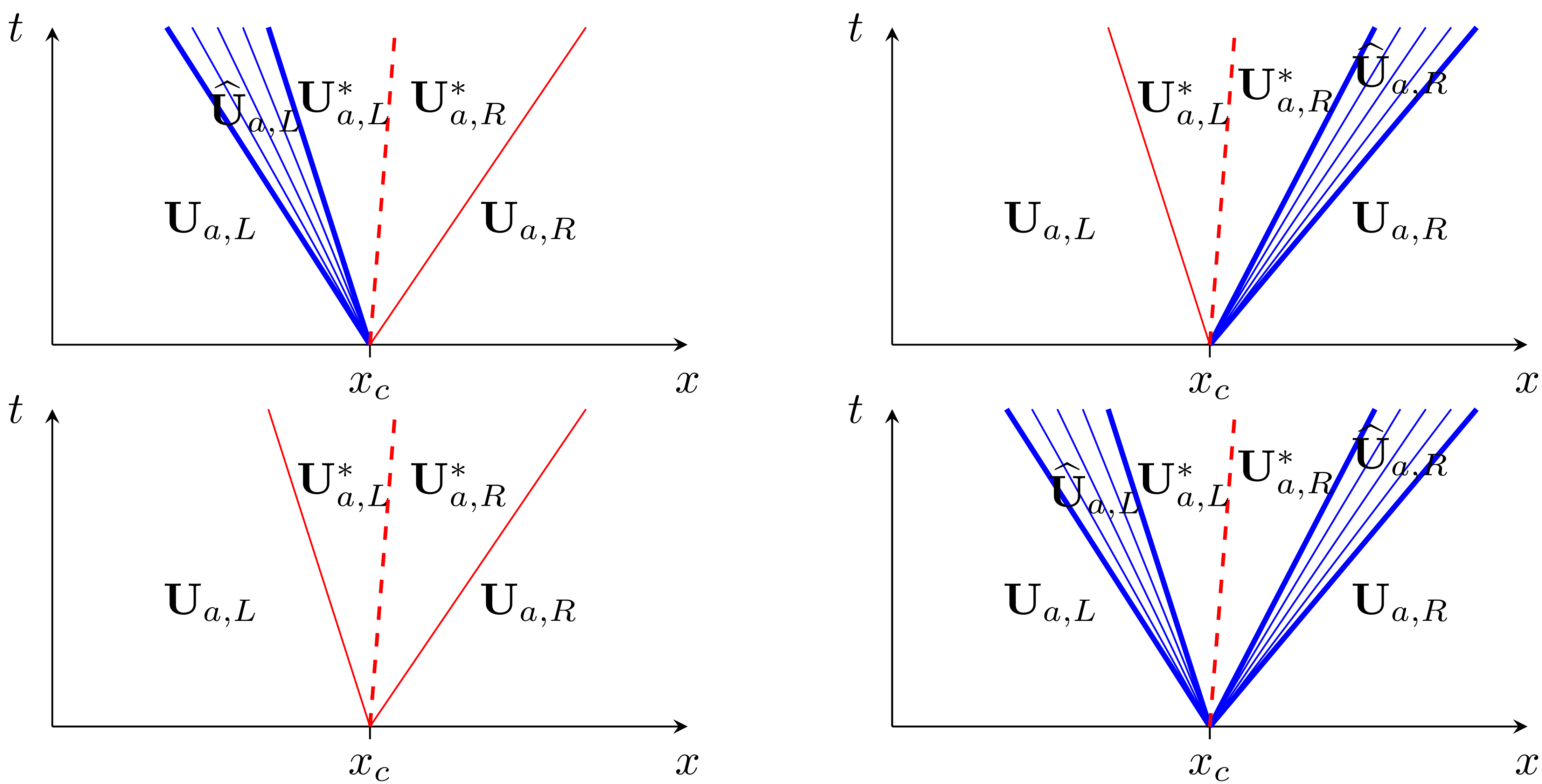
The Riemann problem for the sensitivity equations

The sensitivity system is:

$$\begin{cases} \partial_t \rho_a + \partial_x (\rho u)_a = S_1, \\ \partial_t (\rho u)_a + \partial_x (\rho_a u^2 + 2\rho u u_a + p_a) = S_2, \\ \partial_t (\rho E)_a + \partial_x (u_a(\rho E + p) + u((\rho E)_a + p_a)) = S_3, \end{cases}$$

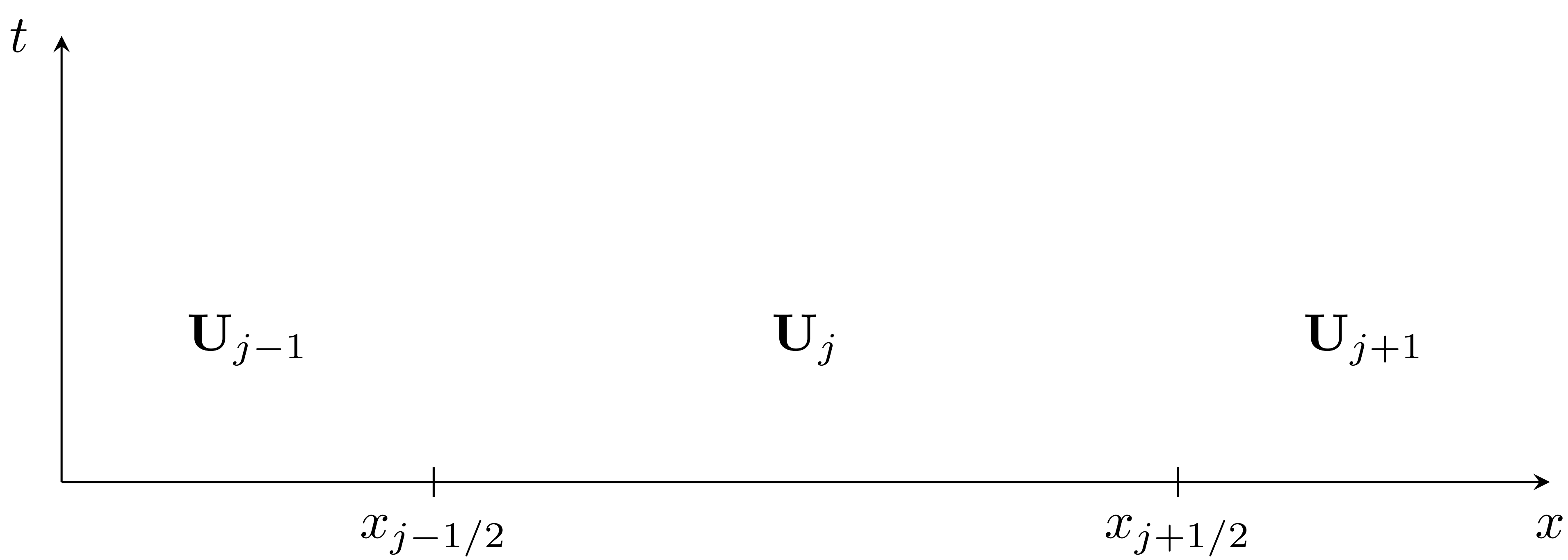
Eigenvalues:

$$\lambda_1(\mathbf{U}) = u - c,$$
$$\lambda_2(\mathbf{U}) = u,$$
$$\lambda_3(\mathbf{U}) = u + c.$$



Godunov-type schemes

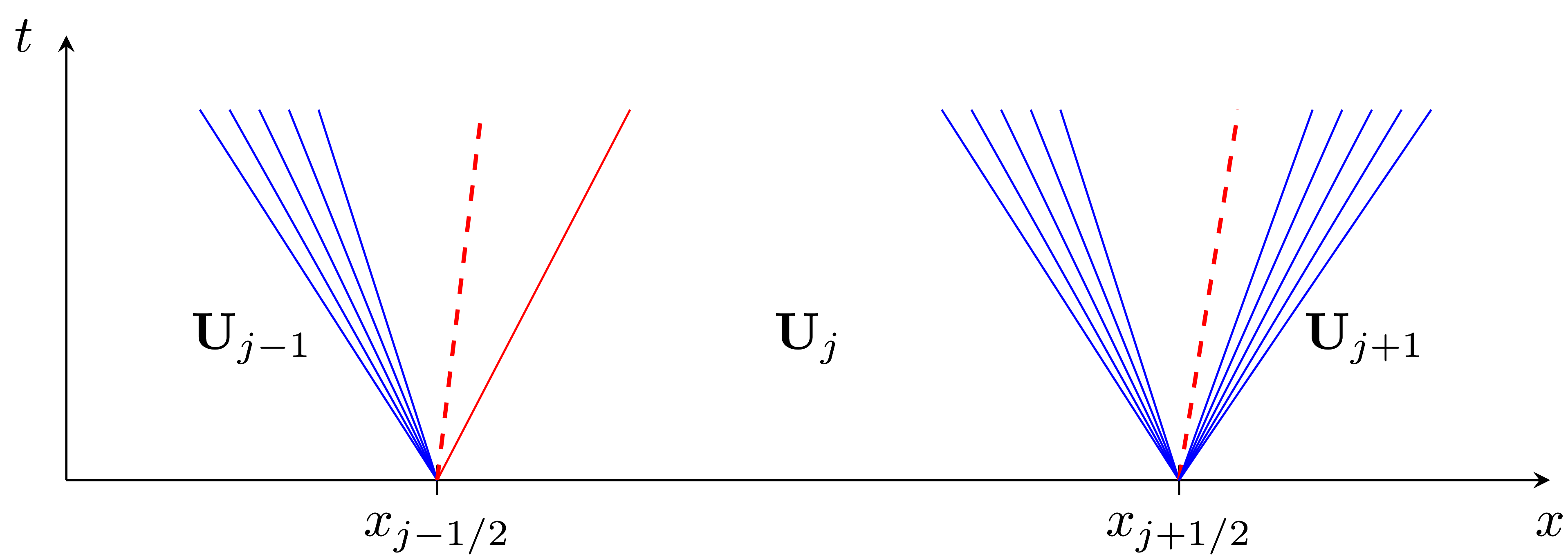
Step 0 : initial data discretisation



Godunov-type schemes

Step 0 : initial data discretisation

Step 1 : solution of a Riemann problem for each interface $x_{j-1/2}$ obtaining $\mathbf{v}(x, t^{n+1})$

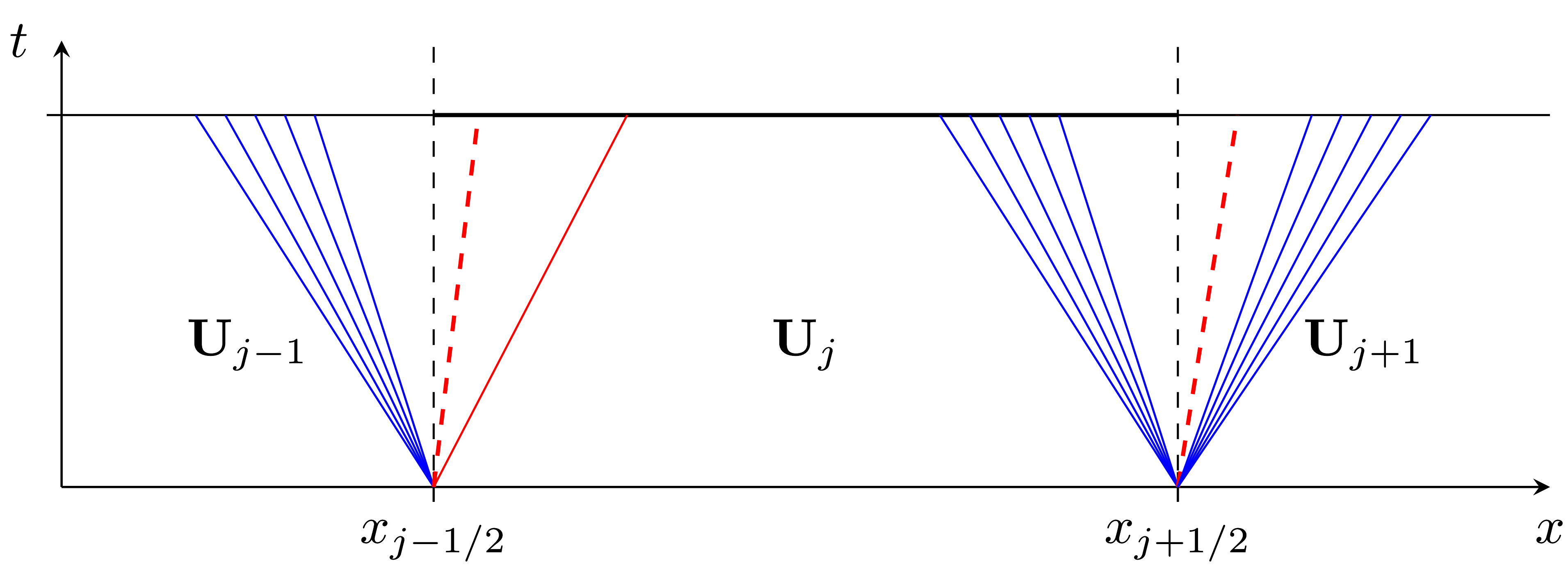


Godunov-type schemes

Step 0 : initial data discretisation

Step 1 : solution of a Riemann problem for each interface $x_{j-1/2}$ obtaining $\mathbf{v}(x, t^{n+1})$

Step 2 : average $\mathbf{V}_j^{n+1} = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \mathbf{v}(x, t^{n+1}) \mathrm{d}x$



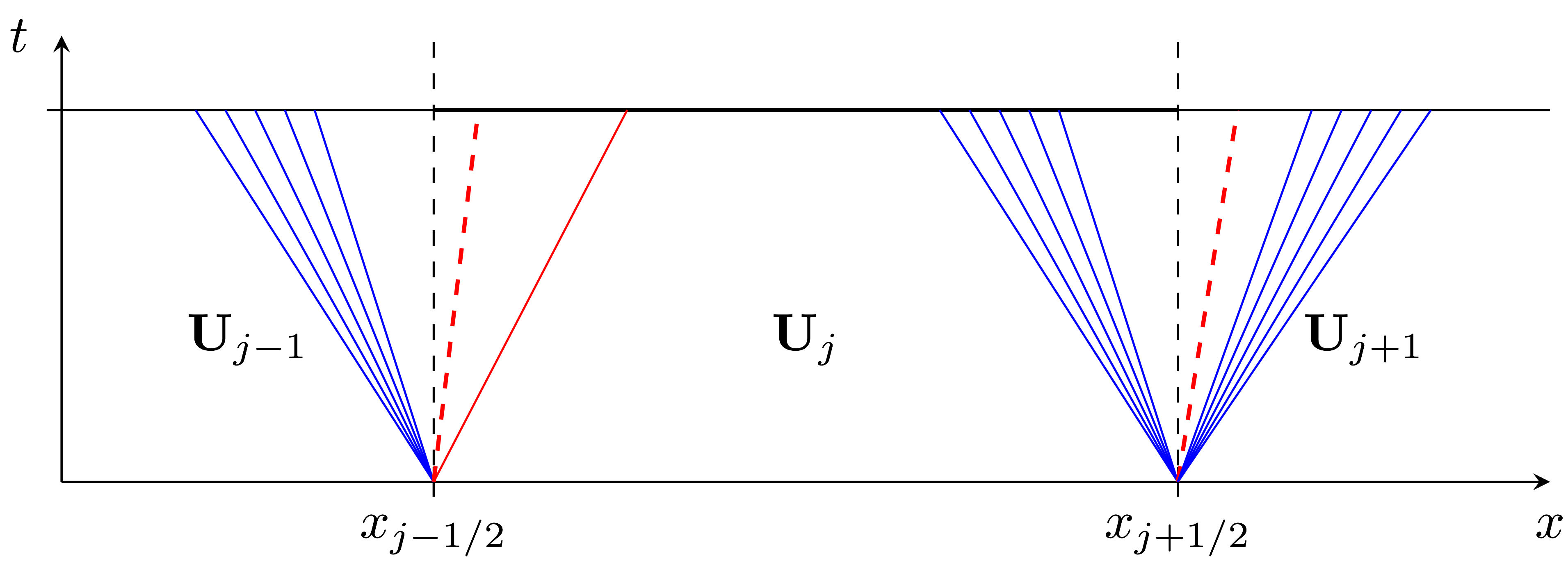
Godunov-type schemes

approximate Riemann solvers are used

Step 0 : initial data discretisation

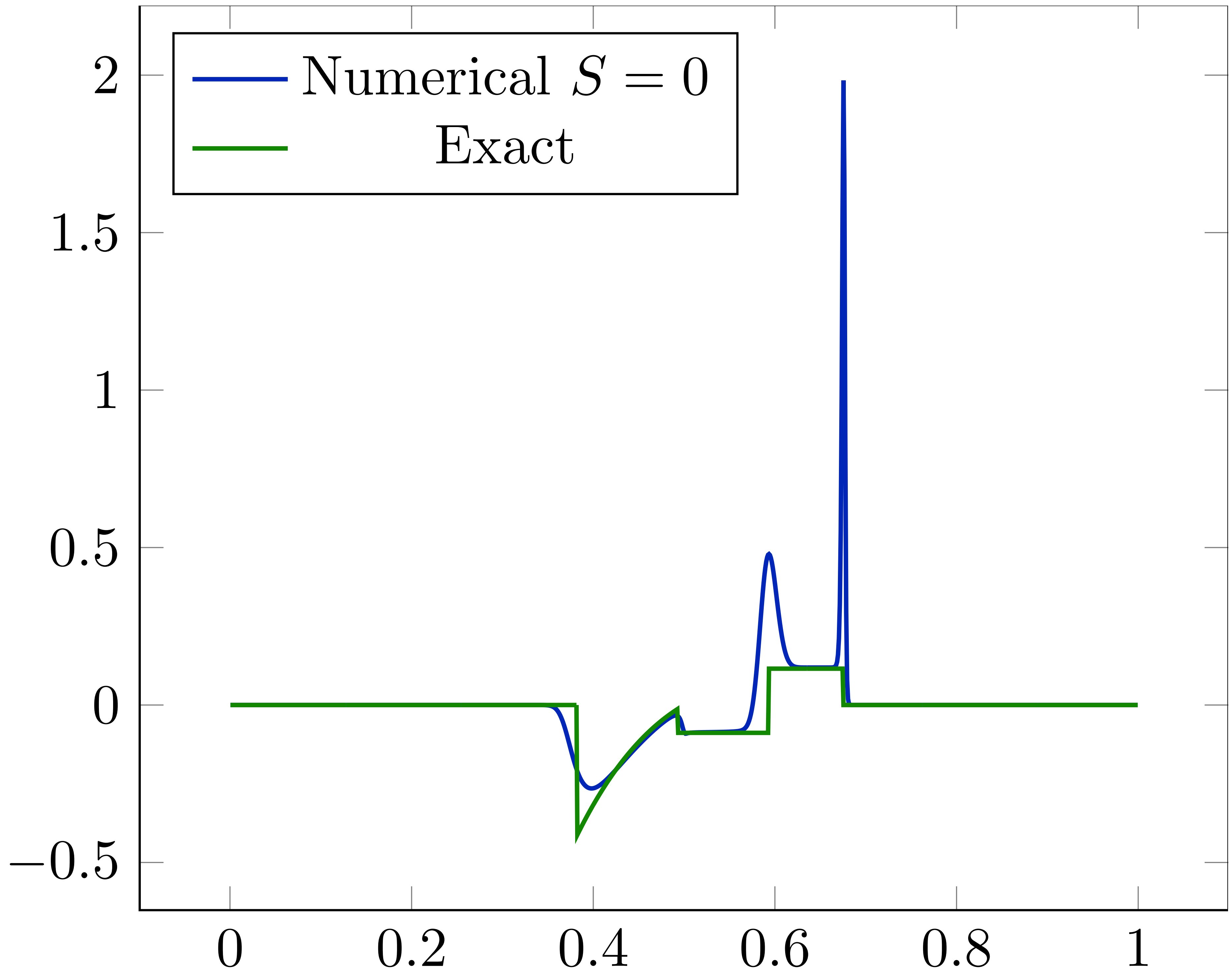
Step 1 : solution of a Riemann problem for each interface $x_{j-1/2}$ obtaining $\mathbf{v}(x, t^{n+1})$

Step 2 : average $\quad \mathbf{V}_j^{n+1} = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \mathbf{v}(x, t^{n+1}) \mathrm{d}x$

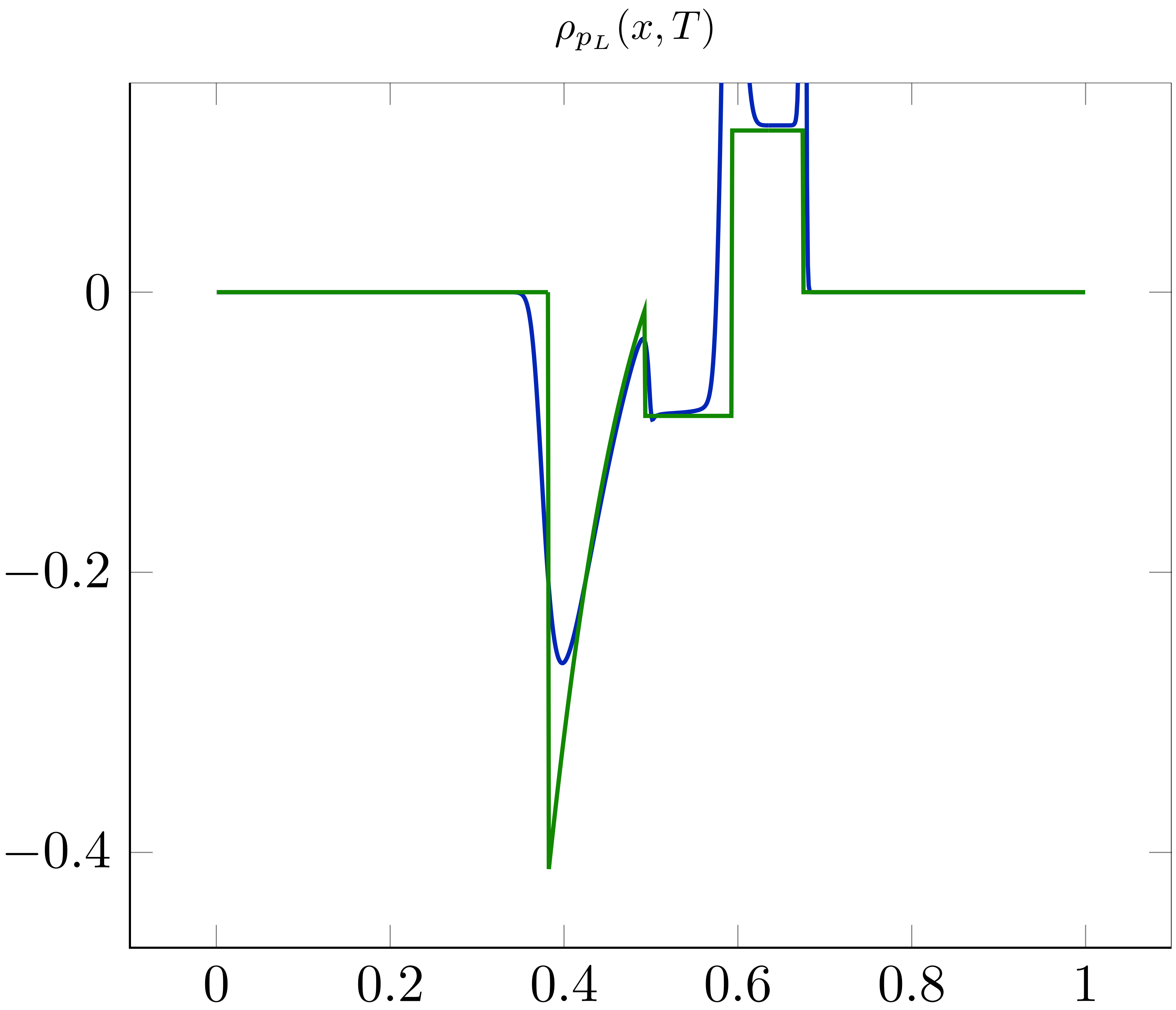


Classical numerical schemes

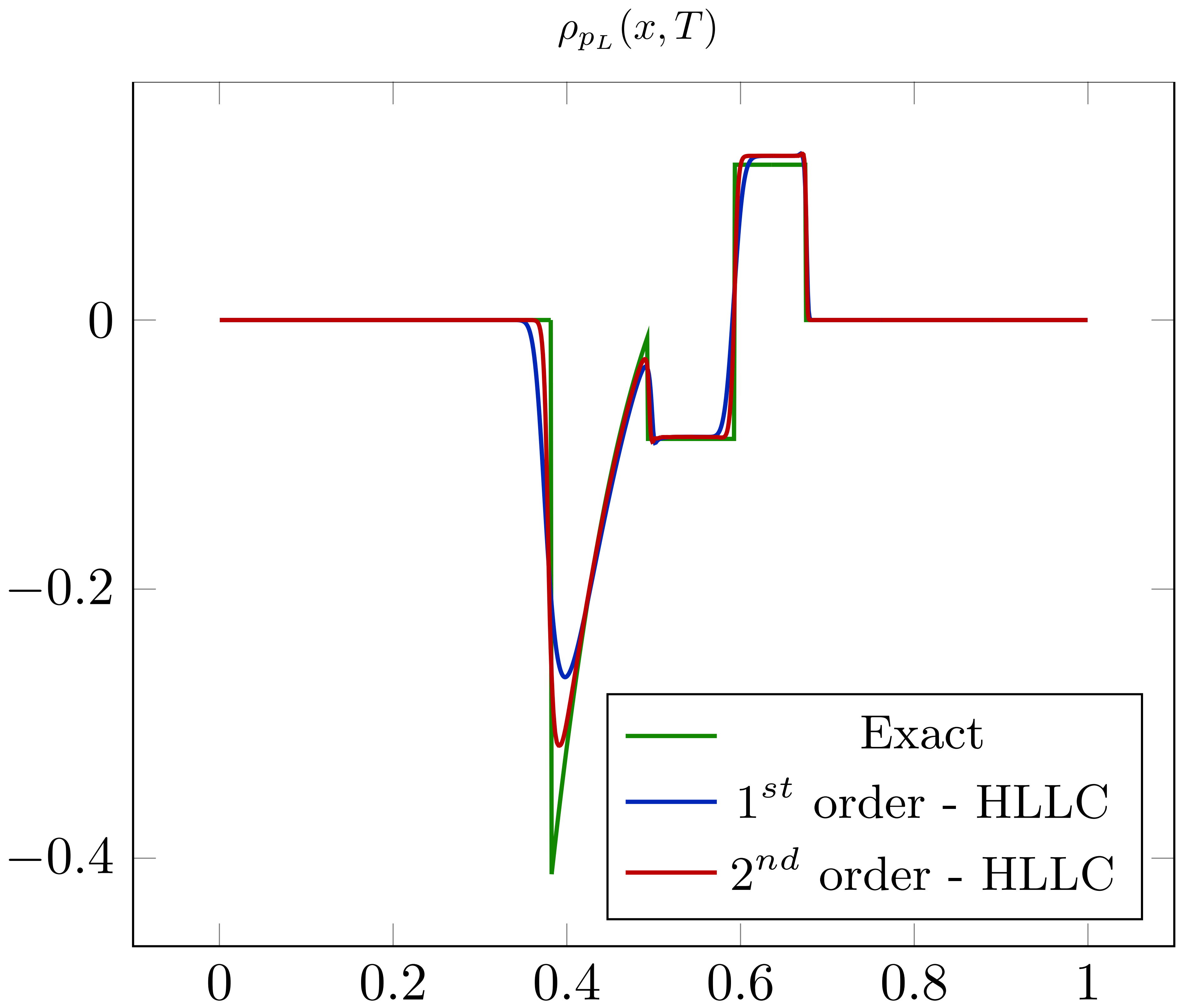
$\rho_{p_L}(x,T)$



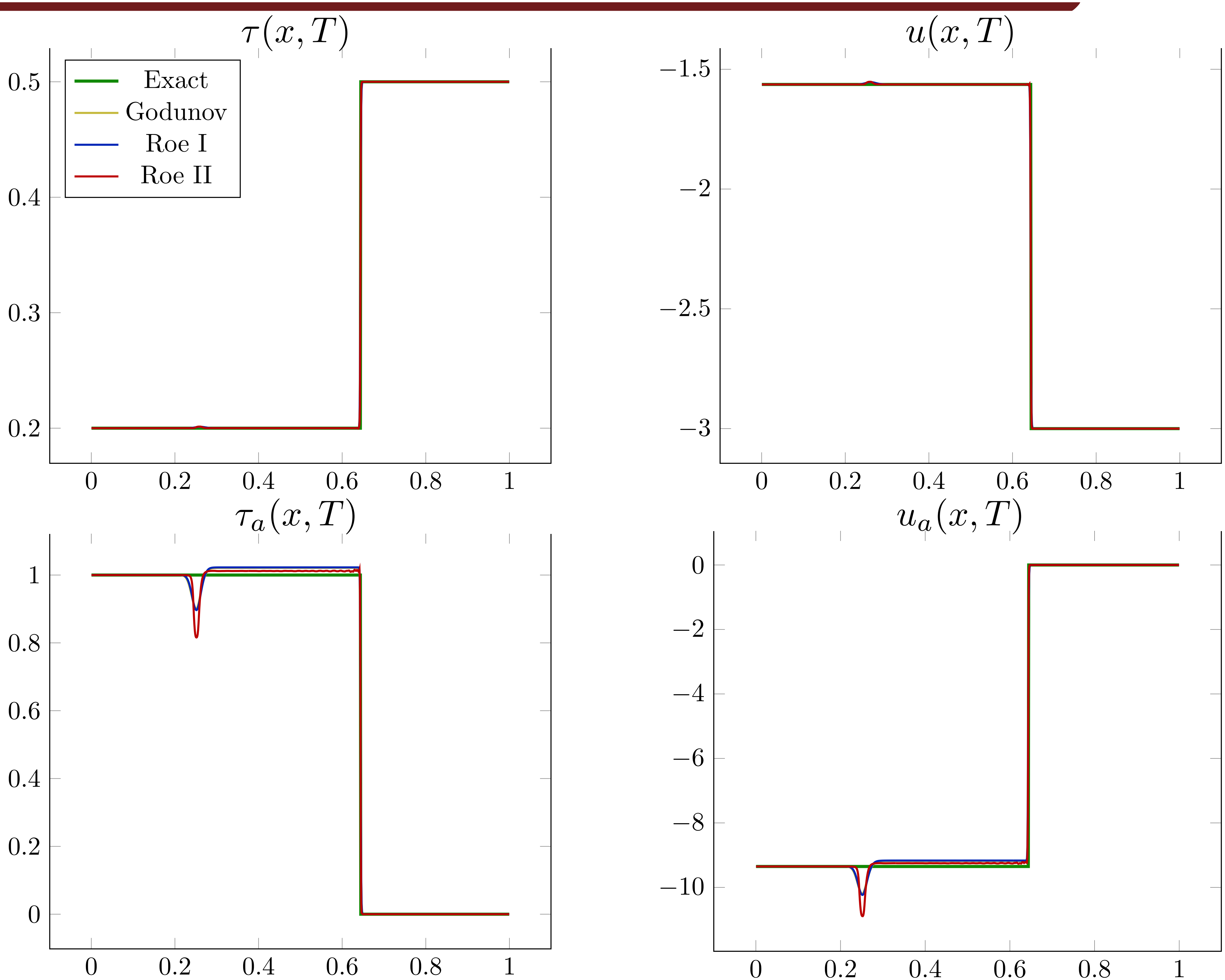
Classical numerical schemes



Classical numerical schemes



Isolated shock for the *p*-system

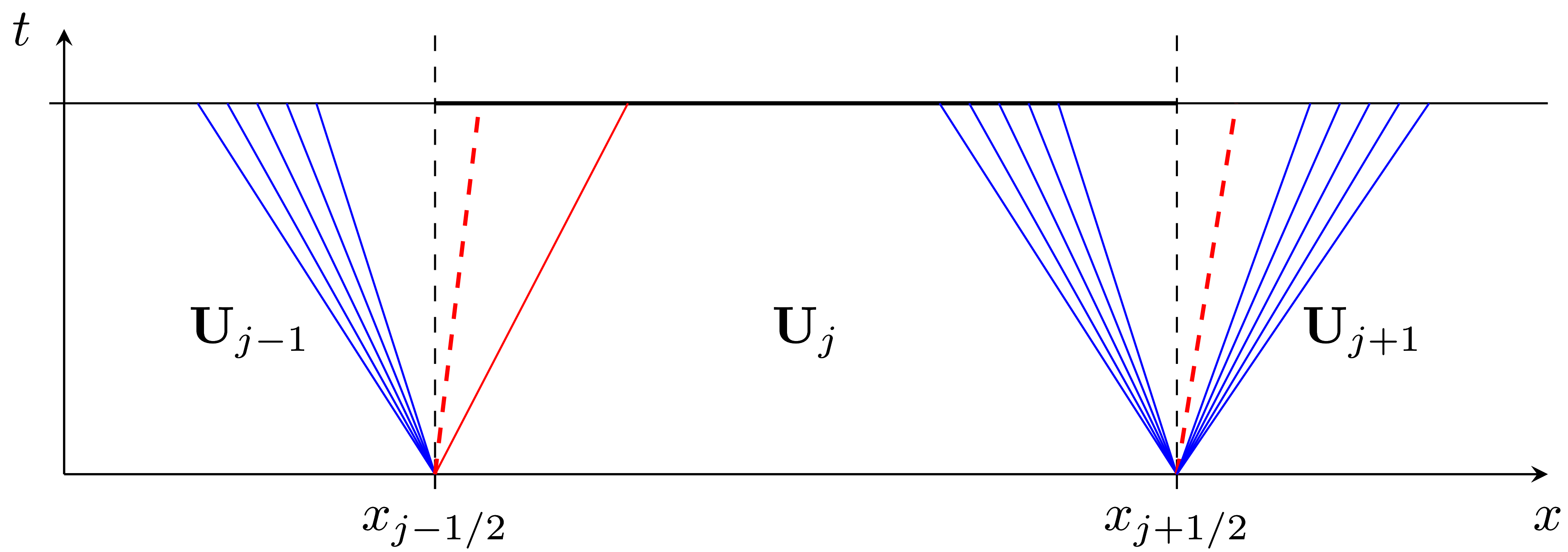


Scheme without numerical diffusion

Step 0 : initial data discretisation

Step 1 : solution of the Riemann problems, one for each interface

Step 2 : ~~average~~

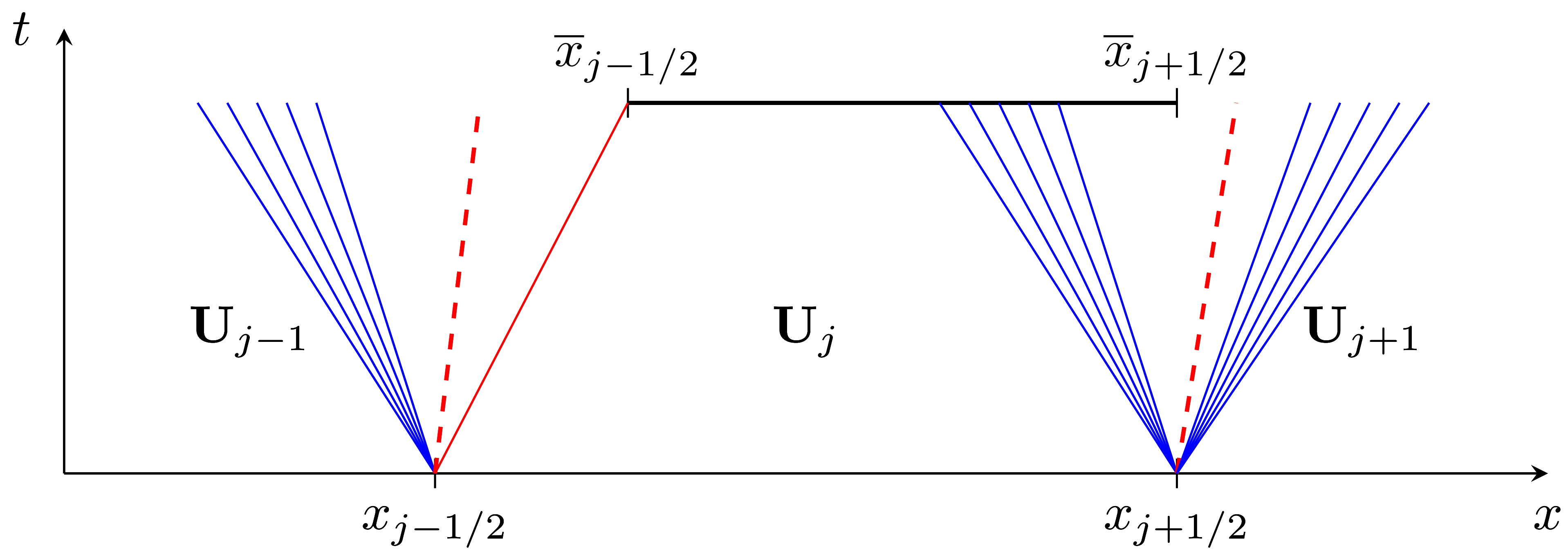


Scheme without numerical diffusion

Step 0 : initial data discretisation

Step 1 : solution of the Riemann problems, one for each interface

Step 2 : definition of a staggered mesh on which the average is performed [4]



$$\overline{x}_{j-1/2} = x_{j-1/2} + \sigma_{j-1/2}\Delta t$$

[4] Chalons, C., Goatin, P. (2008). Godunov scheme and sampling technique for computing phase transitions in traffic flow modeling. *Interfaces and Free Boundaries*, *10*(2), 197-221.

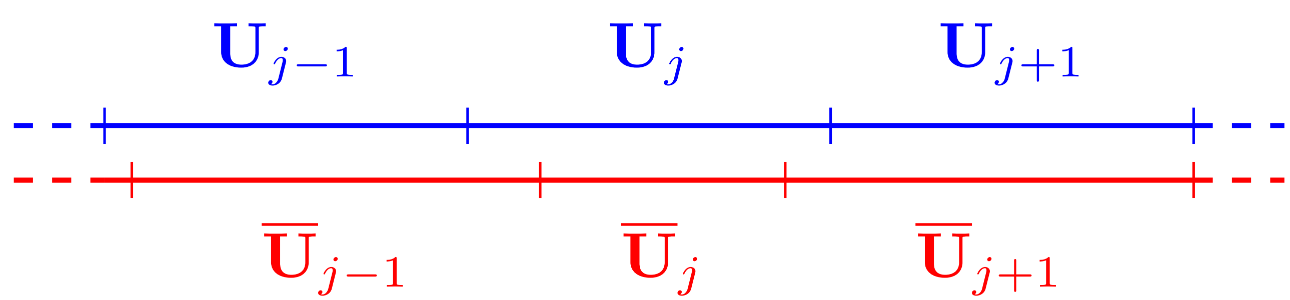
Scheme without numerical diffusion

Step 0 : initial data discretisation

Step 1 : solution of the Riemann problems, one for each interface

Step 2 : definition of a staggered mesh on which the average is performed

Step 3 : projection on the initial mesh [5]

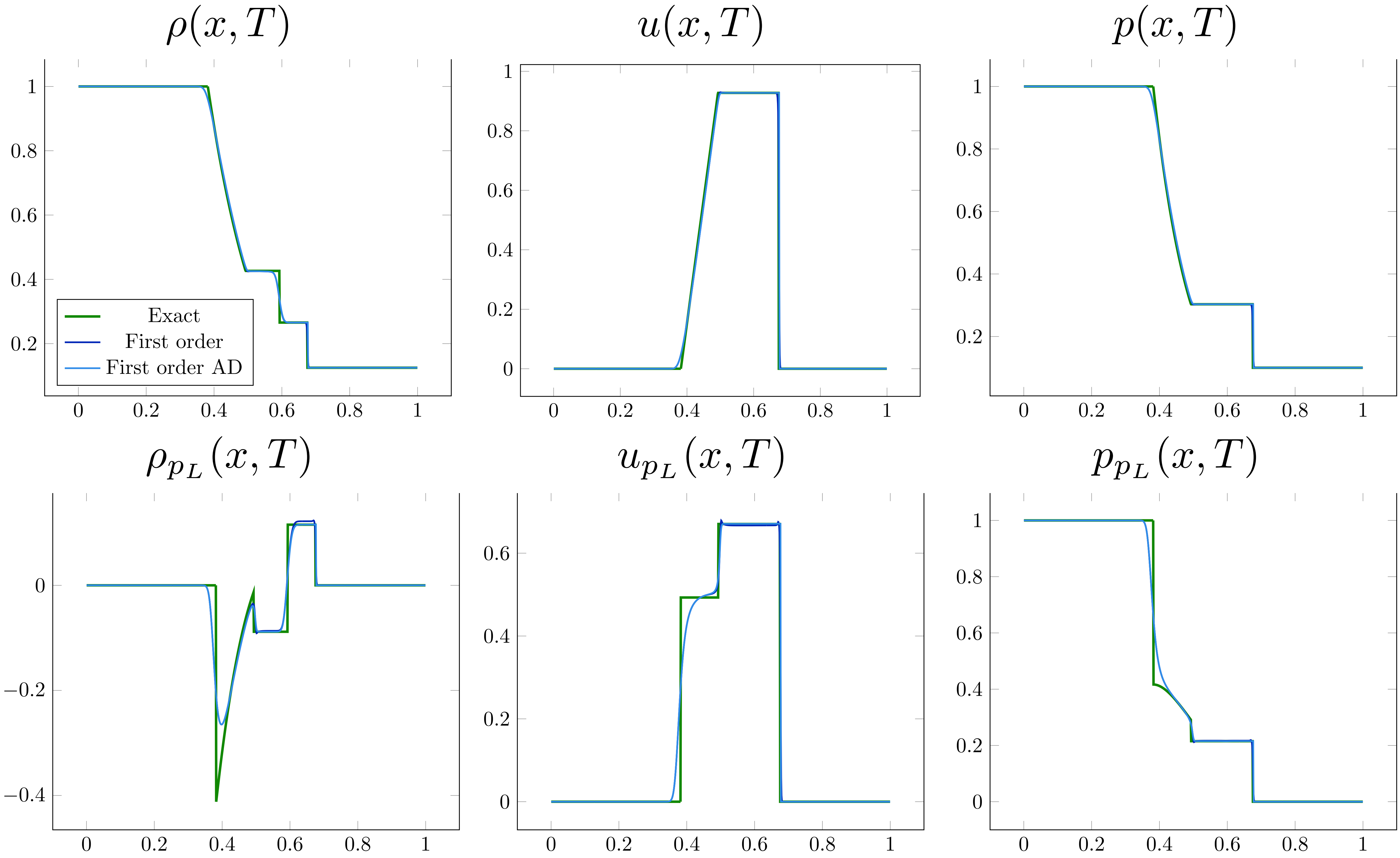


$$\mathbf{U}_j = \begin{cases} \overline{\mathbf{U}}_{j-1} & \text{if } \alpha \in \left(0, \frac{\Delta t}{\Delta x}\max(\sigma_{j-1/2}, 0)\right), \\ \overline{\mathbf{U}}_{j} & \text{if } \alpha \in \left[\frac{\Delta t}{\Delta x}\max(\sigma_{j-1/2}, 0), 1 + \frac{\Delta t}{\Delta x}\min(\sigma_{j+1/2}, 0)\right), \\ \overline{\mathbf{U}}_{j+1} & \text{if } \alpha \in \left[1 + \frac{\Delta t}{\Delta x}\min(\sigma_{j+1/2}, 0), 1\right). \end{cases}$$

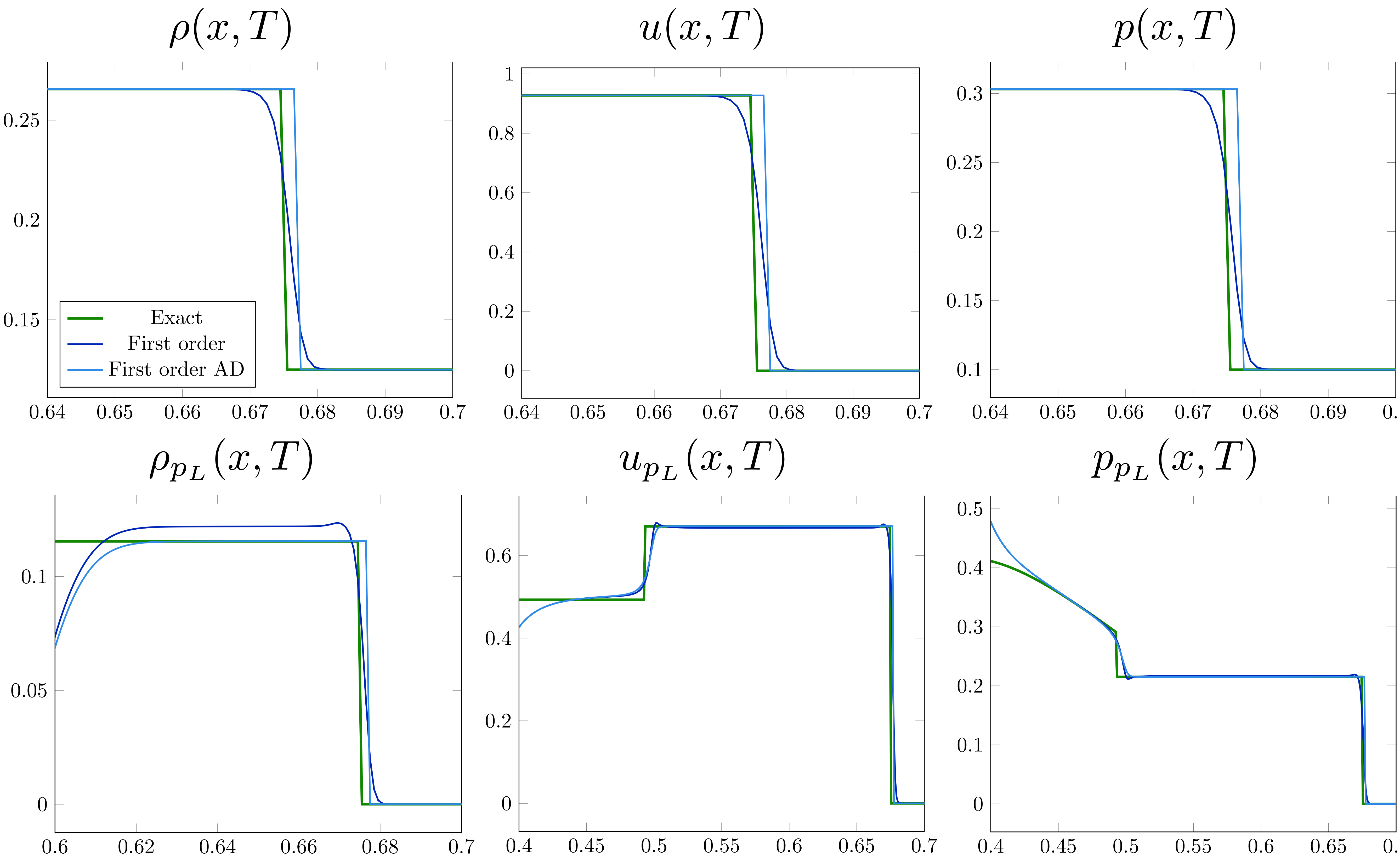
$$\alpha \sim \mathcal{U}([0,1])$$

[5] Glimm, J. (1965). Solutions in the large for nonlinear hyperbolic systems of equations. *Communications on pure and applied mathematics*, *18*(4), 697-715.

Numerical results

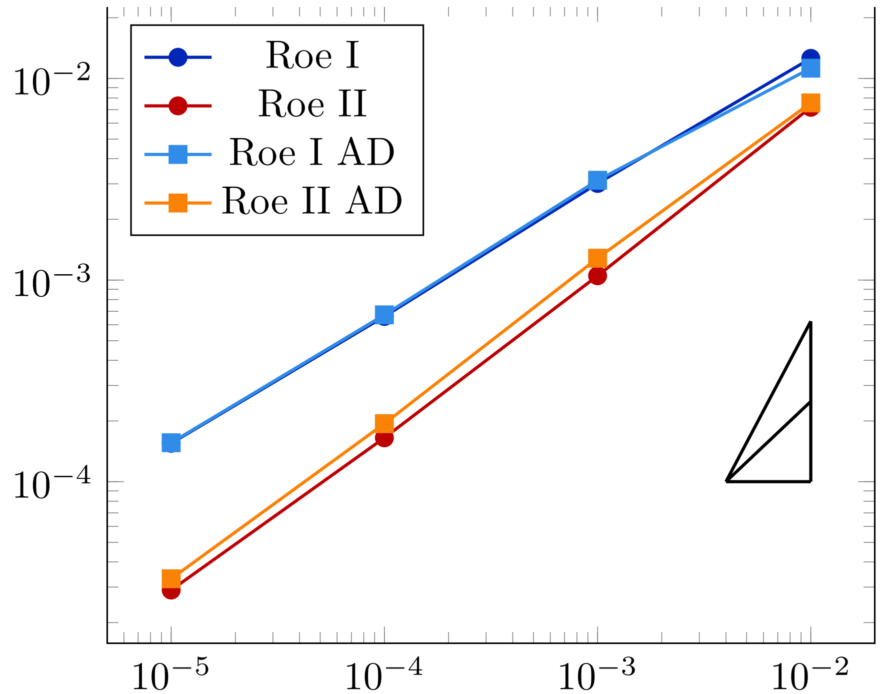


Numerical results

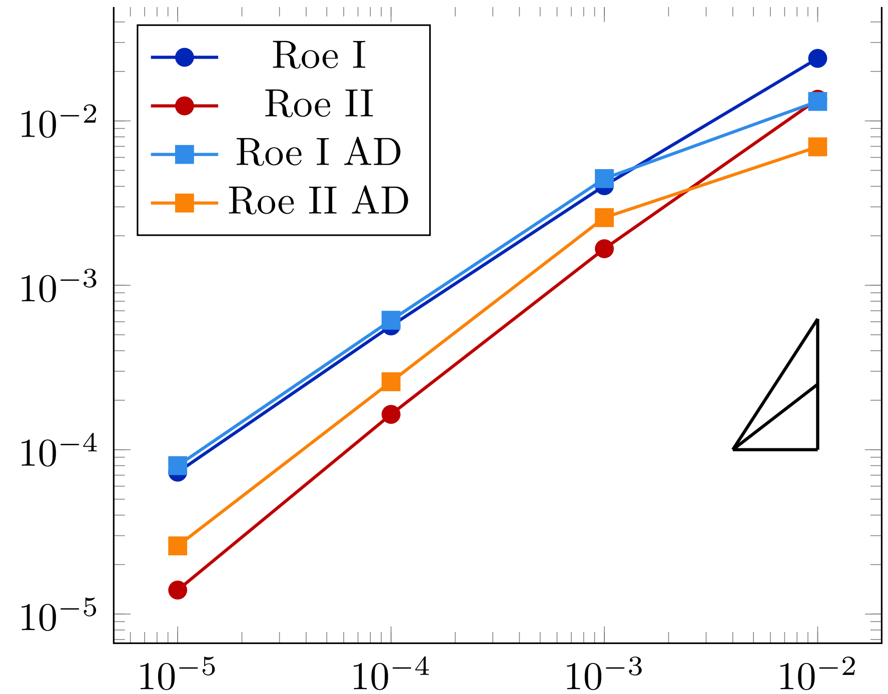


Convergence

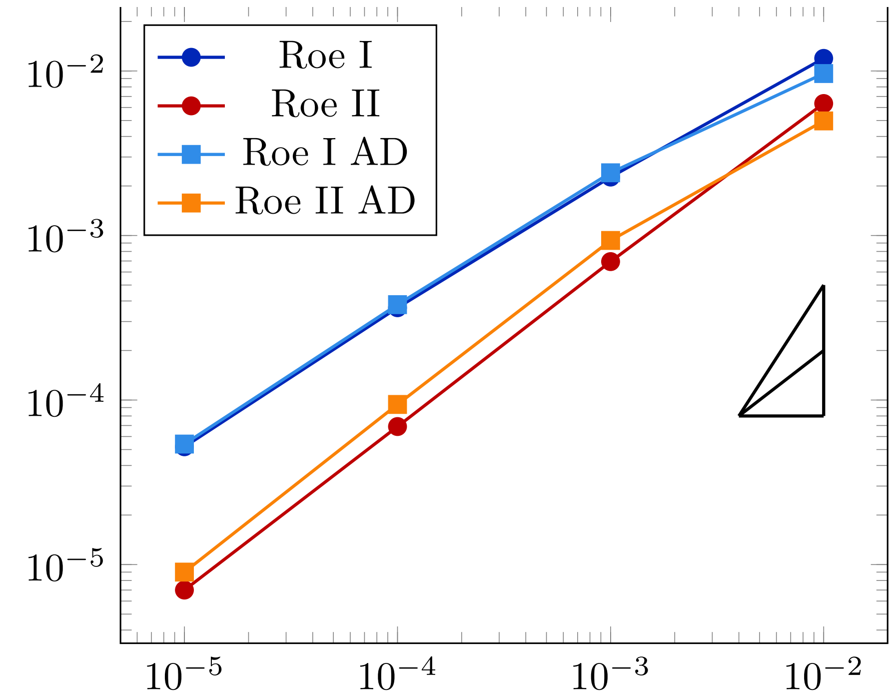
$$\|\rho^{ex}(x,T) - \rho(x,T)\|_{L^1(0,1)}$$



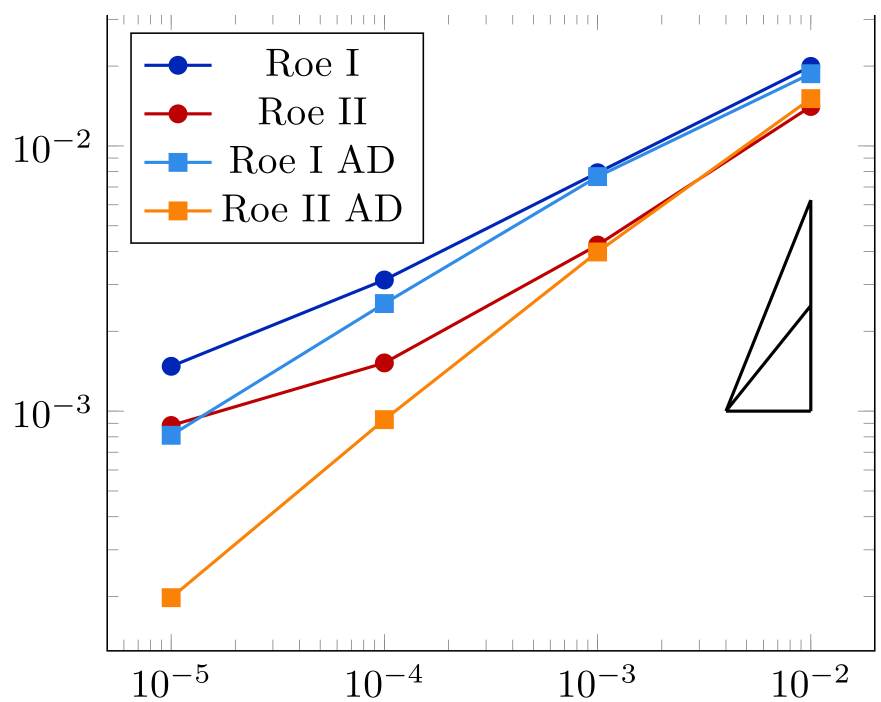
$$\|u^{ex}(x,T) - u(x,T)\|_{L^1(0,1)}$$



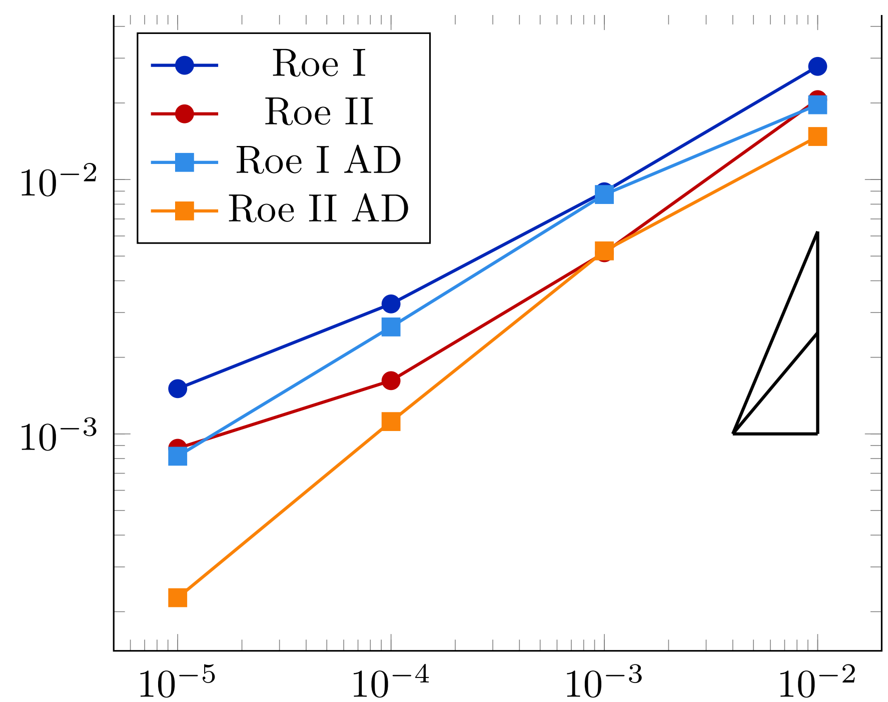
$$\|p^{ex}(x,T) - p(x,T)\|_{L^1(0,1)}$$



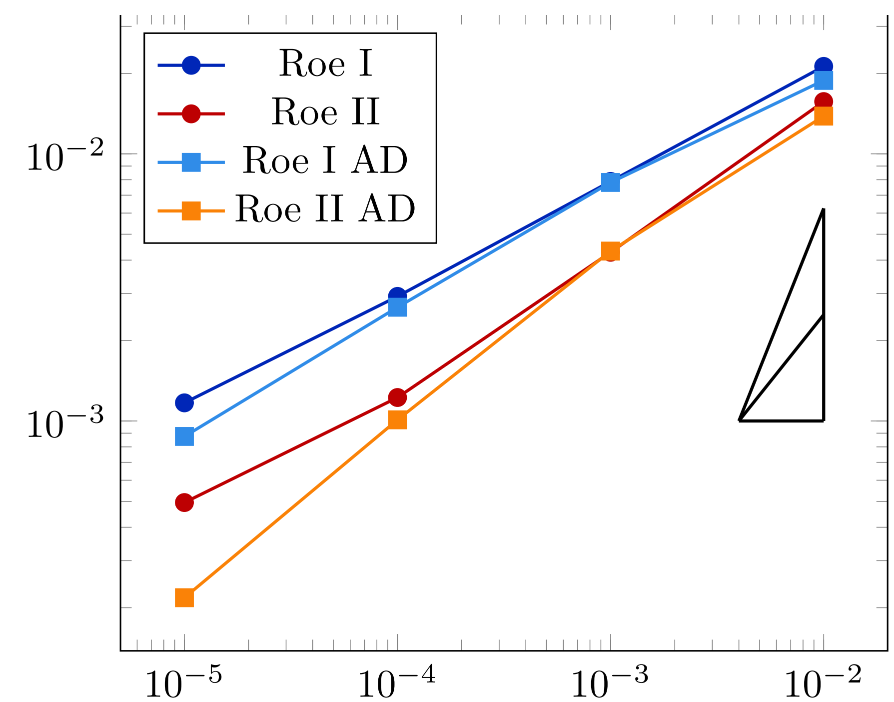
$$\|\rho^{ex}_{p_L}(x,T) - \rho_{p_L}(x,T)\|_{L^1(0,1)}$$



$$\|u^{ex}_{p_L}(x,T) - u_{p_L}(x,T)\|_{L^1(0,1)}$$



$$\|p^{ex}_{p_L}(x,T) - p_{p_L}(x,T)\|_{L^1(0,1)}$$



Optimisation

The quasi-1D Euler equations are:

$$(1)\quad \begin{cases} \partial_t(h\rho) + \partial_x(h\rho u) = 0, \\ \partial_t(h\rho u) + \partial_x(h\rho u^2 + p) = p\partial_x h, \\ \partial_t(h\rho E) + \partial_x(hu(\rho E + p)) = 0, \end{cases}$$

+b.c.

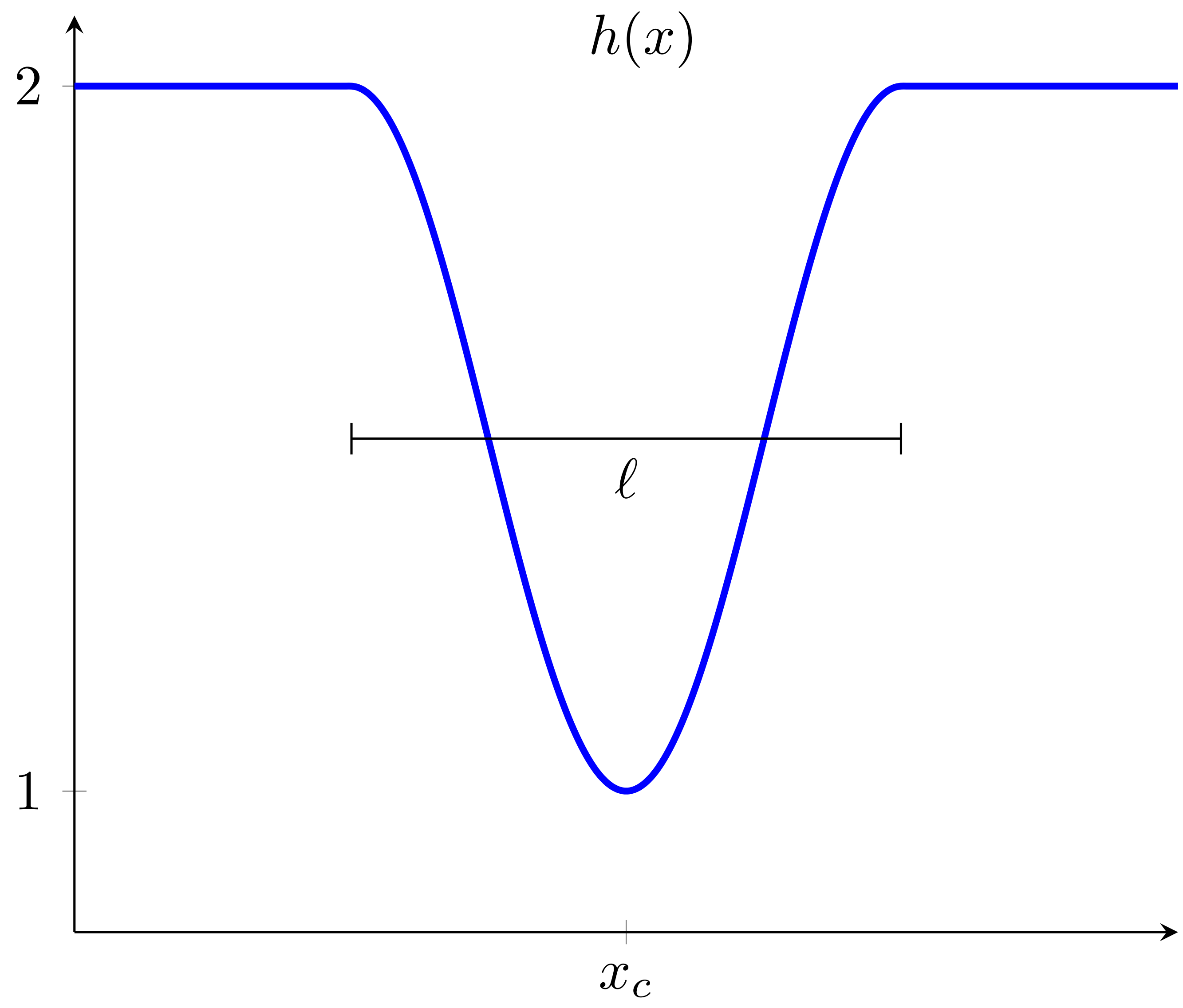
Cost functional: $J(\mathbf{U}) = \frac{1}{2}\|p - p^*\|^2_{L^2}$

Parameters: $\mathbf{a} = (x_c, \ell)^t$

Target pressure: $p^* = p(x_c = 0.5, \ell = 0.5)$

Gradient: $\nabla_\mathbf{a} J(\mathbf{U}) = \begin{bmatrix} (p - p^*, p_{x_c})_{L^2} \\ (p - p^*, p_\ell)_{L^2} \end{bmatrix}$

Optimisation problem: $\min_{\mathbf{a} \in \mathcal{A}} J(\mathbf{U})$ subject to (1).



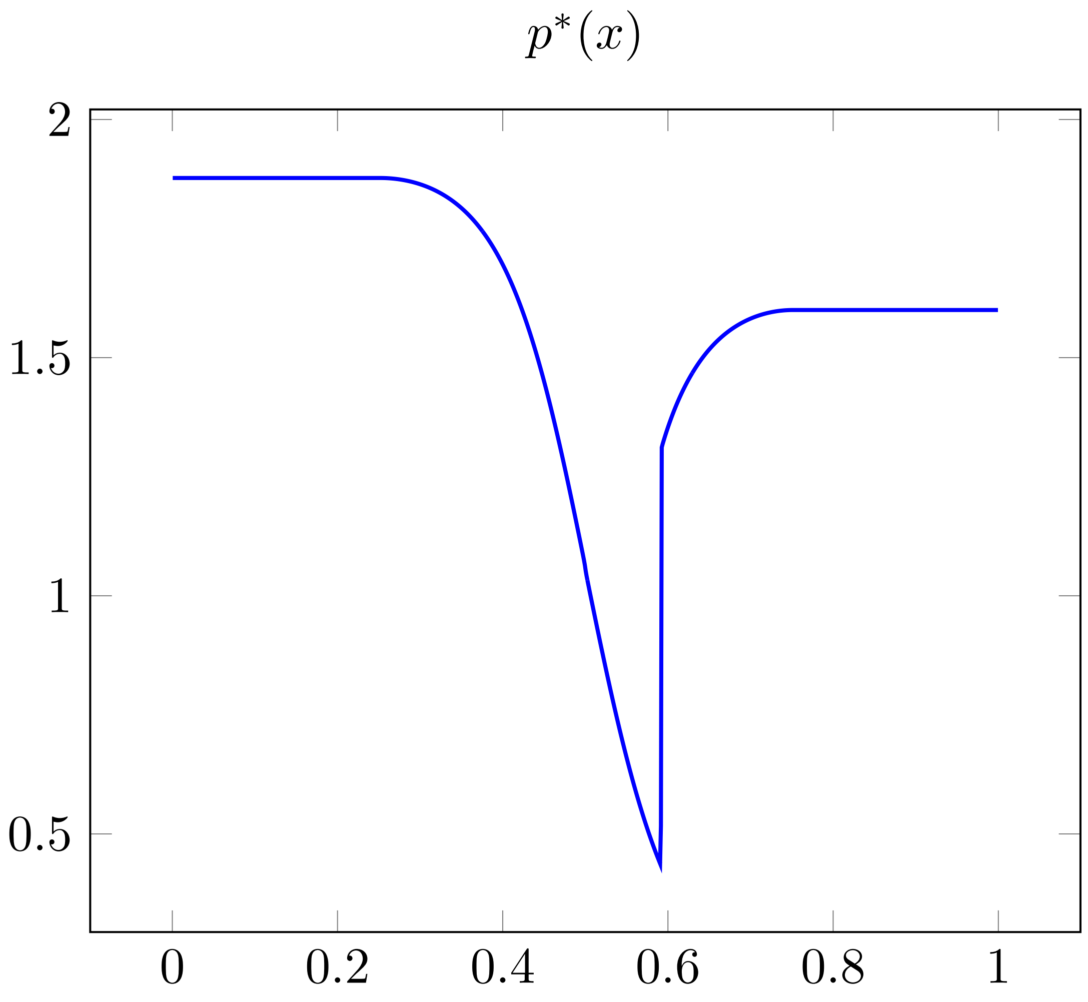
Optimisation

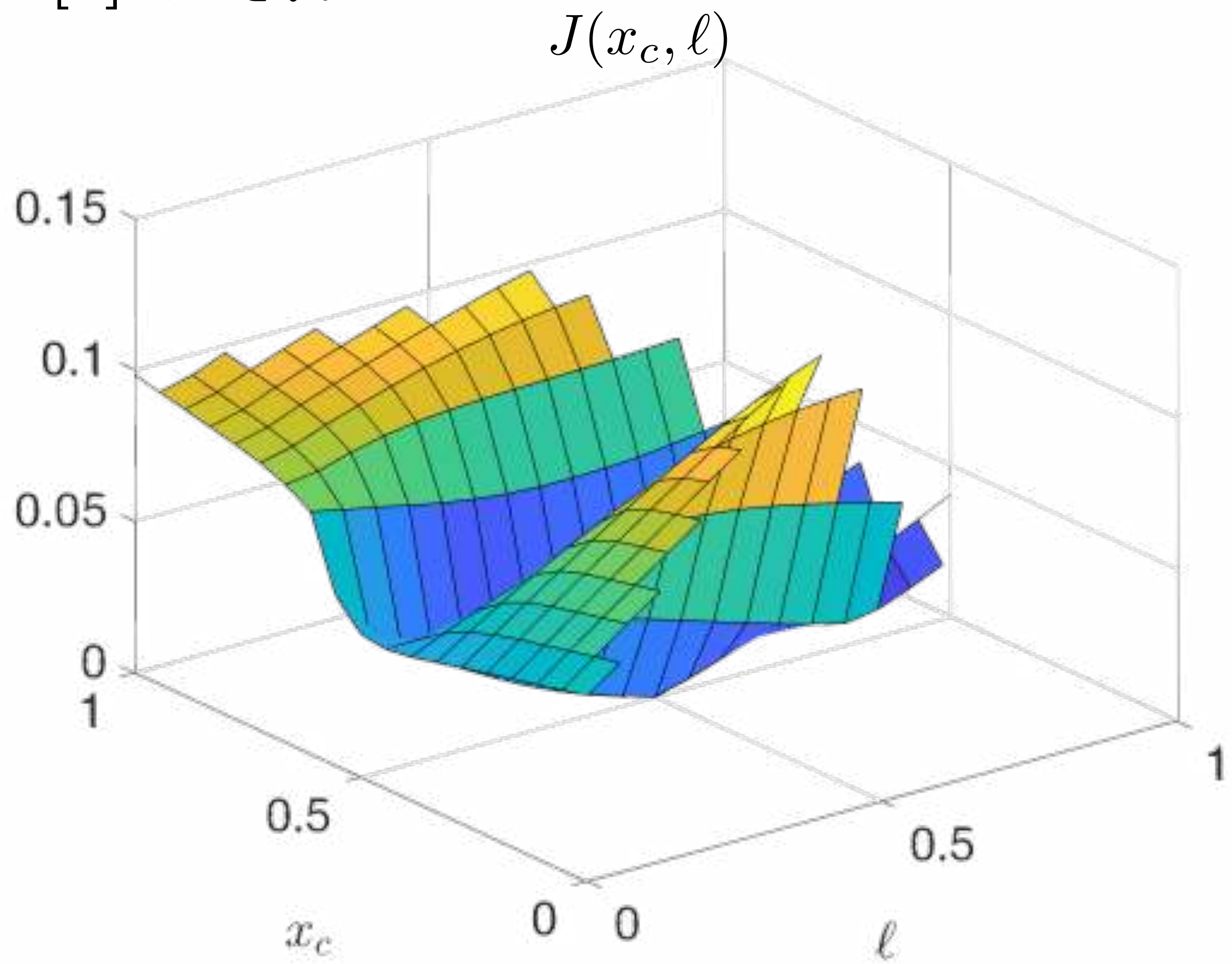
Boundary conditions:

- inlet: enthalpy H_L and total pressure $p_{tot,L}$
- outlet: pressure p_R

$$\begin{cases} H &= E + \dfrac{p}{\rho}, \\ p &= (\gamma - 1)\left(\rho E - \dfrac{1}{2}\rho u^2\right), \\ p_{tot} &= p + \dfrac{1}{2}\rho u^2. \end{cases}$$

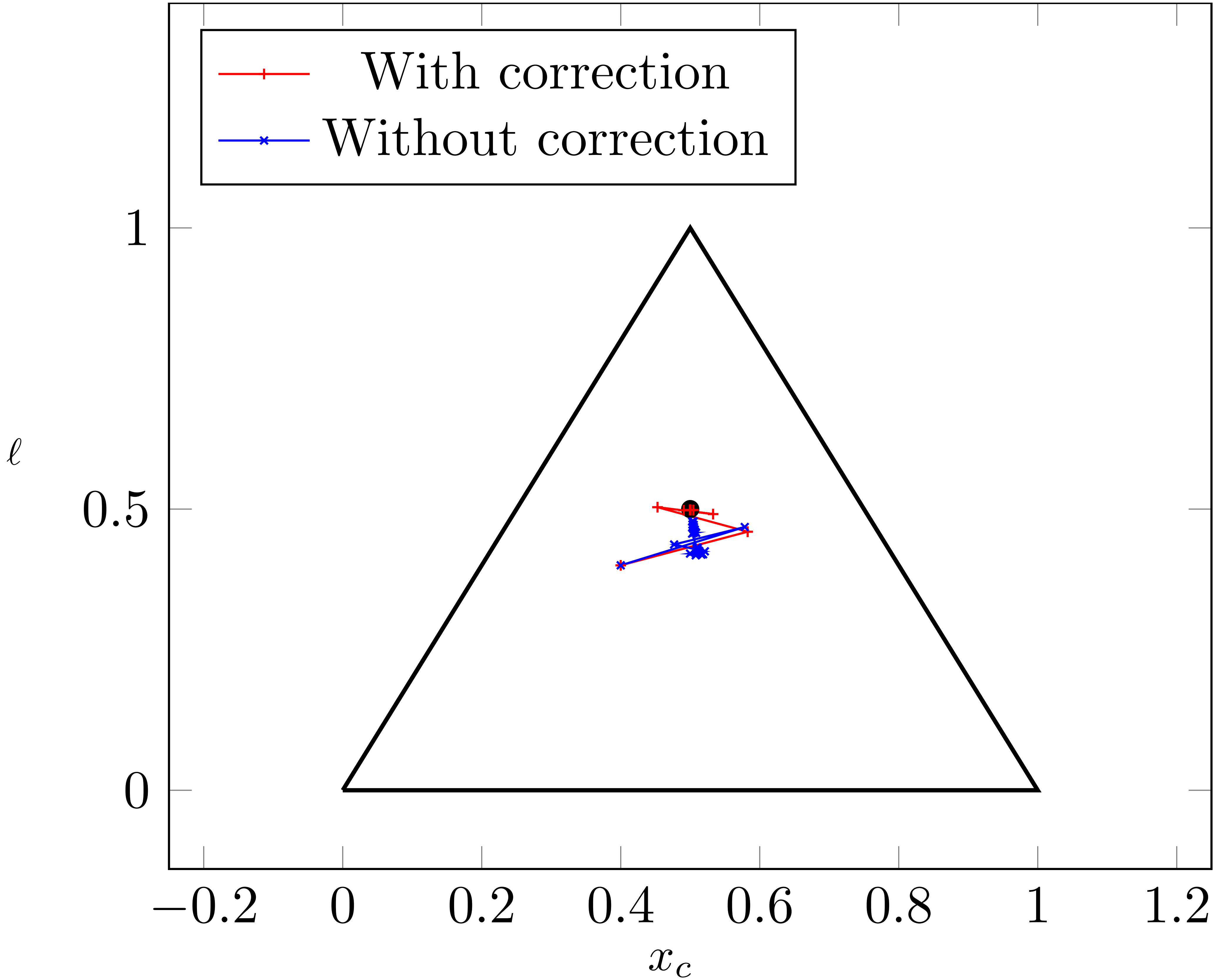
These b.c. provide a discontinuous solution [6] $\forall \mathbf{a} \in \mathcal{A}$.



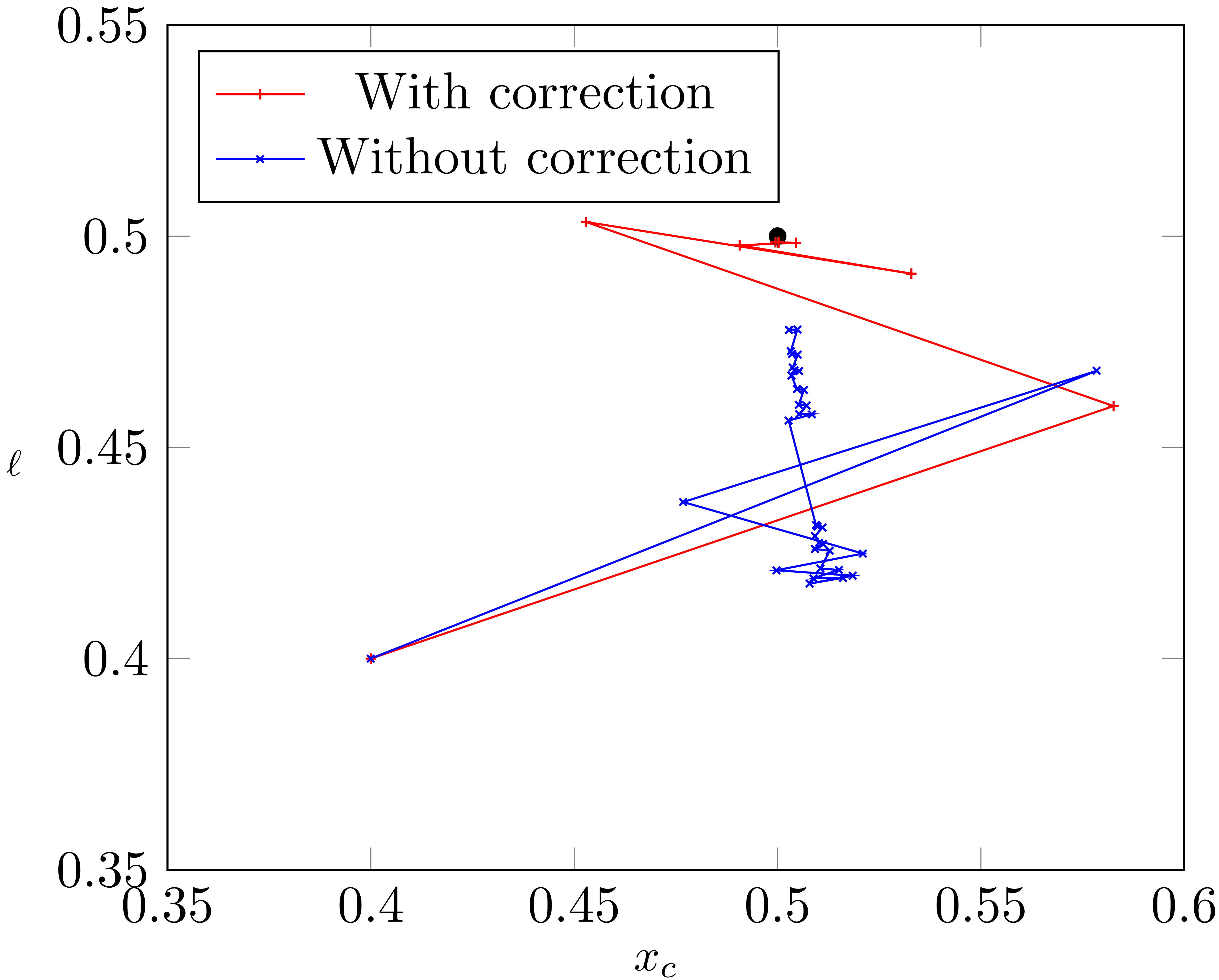


[6] Giles, M. B., & Pierce, N. A. (2001). Analytic adjoint solutions for the quasi-one-dimensional Euler equations. *Journal of Fluid Mechanics*, *426*, 327-345.

Optimisation



Optimisation



Higher order schemes in time for stochastic PDEs modelling ocean dynamics

Modelling under location uncertainty

The **LU framework**, introduced in [1], is based on the following decomposition of the Lagrangian velocity in two components

$$\mathrm{d}\mathbf{X}_t = \mathbf{u}(\mathbf{X}_t, t)\mathrm{d}t + \sigma(\mathbf{X}_t, t)\mathrm{d}\mathbf{B}_t$$

$\mathrm{d}\mathbf{X}_t(\mathbf{x})$ is the Lagrangian displacement of the fluid particle starting at point **x** at time *t = 0*.

u is the large scale, smooth, resolved component of the velocity field.

$\mathbf{B}_t$ is a standard brownian motion, i.e.:

- $\mathbf{B}_0 = 0$;
- **B** is almost certainly continuous;
- $\forall\, t_1 \leq t_2 \leq t_3 \leq t_4,\ \mathbf{B}_{t_4} - \mathbf{B}_{t_3}$ and $\mathbf{B}_{t_2} - \mathbf{B}_{t_1}$ are independent;
- $\forall\, t, s\ \mathbf{B}_t - \mathbf{B}_s \sim \mathcal{N}(0, |t-s|)$.

σ is the correlation operator, whose definition is based on a kernel:

$$\sigma[f](\mathbf{x}, t) := \int_\Omega \breve{\sigma}(\mathbf{x}, \mathbf{y}, t) f(\mathbf{y})\mathrm{d}\mathbf{y}$$

[1] É. Mémin. Fluid flow dynamics under location uncertainty. *Geophysical & Astrophysical Fluid Dynamics*, 2014. *108*(2), 119-146.

Modelling under location uncertainty

Starting from this decomposition, one can compute the **stochastic transport operator**:

$$\mathbb{D}_t b := \mathrm{d}_t b + \mathbf{v}^* \cdot \nabla b \, \mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t \cdot \nabla b - \frac{1}{2} \nabla \cdot (a \nabla b) \mathrm{d}t,$$

where

$$\mathbf{v}^* = \mathbf{u} - \frac{1}{2} \nabla \cdot a - \sigma(\nabla \cdot \sigma)$$ and a is the variance tensor.

We will use the following spectral representation of the noise:

$$\sigma \mathrm{d}\mathbf{B}_t = \sum_m \varphi_m \mathrm{d}\beta_t^m$$

β_t^m: Independent, scalar brownian motions

φ_m: Basis functions, computed from high resolution data

Thanks to this representation, we can give a simple expression for the variance:

$$a = \sum_m \varphi_m \varphi_m^T$$

SQG system under LU

We recall the definition of the **stochastic transport operator**:

$$\mathbb{D}_t b := \mathrm{d}_t b + \mathbf{v}^* \cdot \nabla b \ \mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t \cdot \nabla b - \frac{1}{2} \nabla \cdot (a \nabla b) \mathrm{d}t,$$

Starting from this, one can write the **LU version** of any classical fluid dynamical system. In this work we will focus on the **surface quasi geostrophic** (SQG) system, whose LU version is:

$$\begin{cases} \mathbb{D}_t b = 0, \\ b = N(-\Delta)^{1/2} \psi, \\ \mathbf{u} = \nabla^{\perp} \psi, \end{cases}$$

Stochastic transport of the buoyancy

Linear operator that links velocity and buoyancy through the streamline function

Numerical schemes for stochastic PDEs

For stochastic PDEs, two different notions of convergence exist:

- **Strong convergence**

$$\text{err} = \mathbb{E}\left[\|b_{ex} - b_h\|^2_{L^2(\Omega)}\right]^{1/2}$$

- **Weak convergence**

$$\text{err} = \|\mathbb{E}(b_{ex}) - \mathbb{E}(b_h)\|_{L^2(\Omega)}$$

For stochastic PDEs, the generalisation of the **Euler scheme**, known as the Euler-Maruyama scheme, has **order of strong convergence 0.5**.

Our aim is to develop a scheme with higher order of strong convergence.

Milstein scheme

By replacing everything in the Itō formulas and then into the main equation, keeping only first order a lower order terms one finds:

$$b_t = b_{t_0} + f(b_{t_0})\Delta t - \sum_m g^m(b_{t_0})\Delta\beta^m + \int_{t_0}^{t}\int_{t_0}^{s}\sum_{m,k} g^m(g^k(b_\tau))\mathrm{d}\beta_\tau^k \mathrm{d}\beta_s^m \qquad (1)$$

Euler-Maruyama

We define the following quantities:

$$G^{m,k} := g^m(g^k(b_{t_0})) \qquad I^{m,k} := \int_{t_0}^{t}\int_{t_0}^{s} \mathrm{d}\beta_\tau^k \mathrm{d}\beta_s^m$$

Then the double integral in (1) can be approximated with:

$$\sum_{m,k} G^{m,k} I^{m,k} = \sum_{m,k} G^{m,k}\frac{I^{m,k}+I^{k,m}}{2} + G^{m,k}\frac{I^{m,k}-I^{k,m}}{2}$$

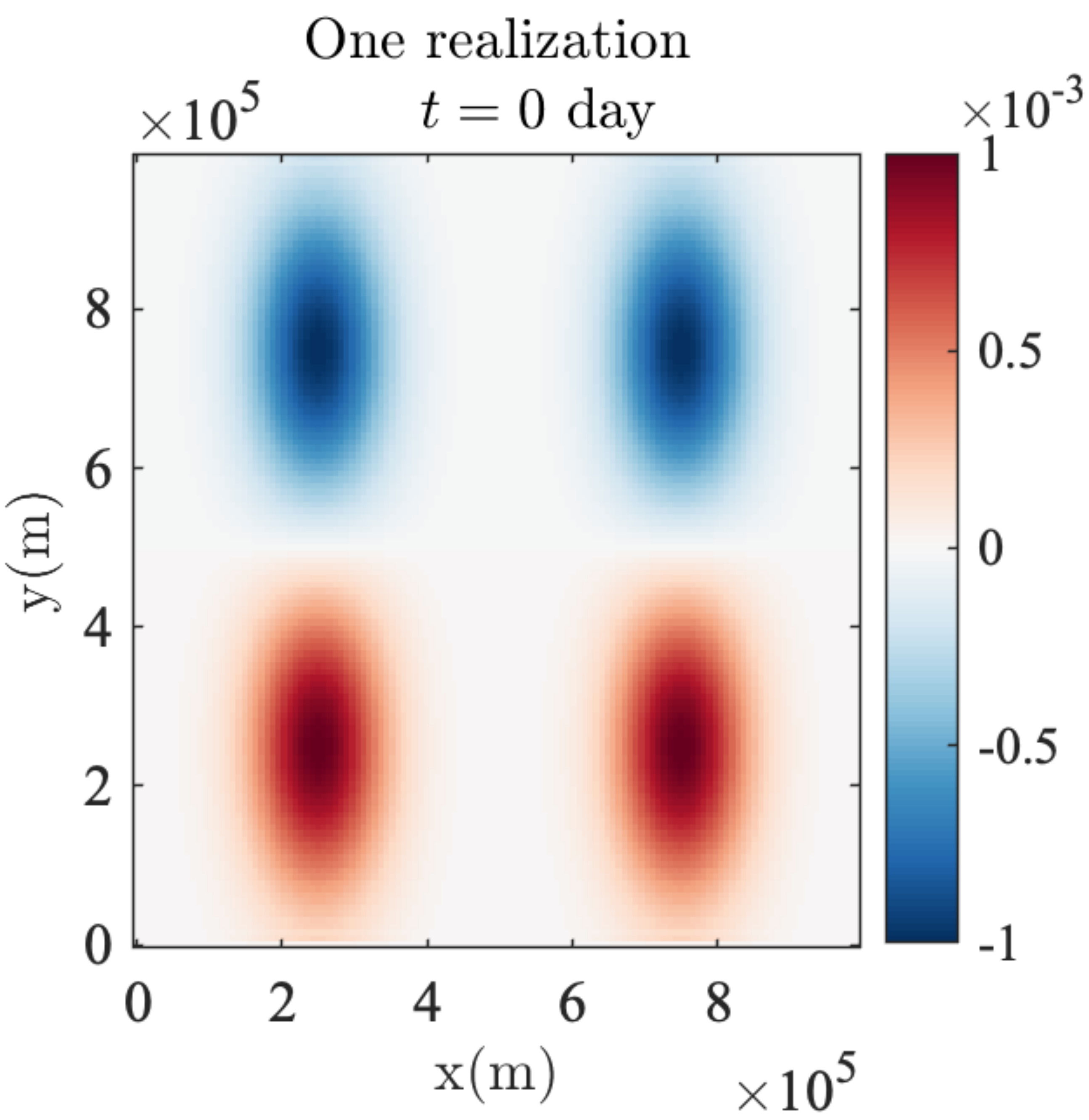
where $\frac{I^{m,k}+I^{k,m}}{2}$: $= \Delta\beta^m\Delta\beta^k - \delta_{m,k}\Delta t$

$\frac{I^{m,k}-I^{k,m}}{2}$: Lévy area, which can be simulated → conditional (recursive approximation) / weak approximation / neglected

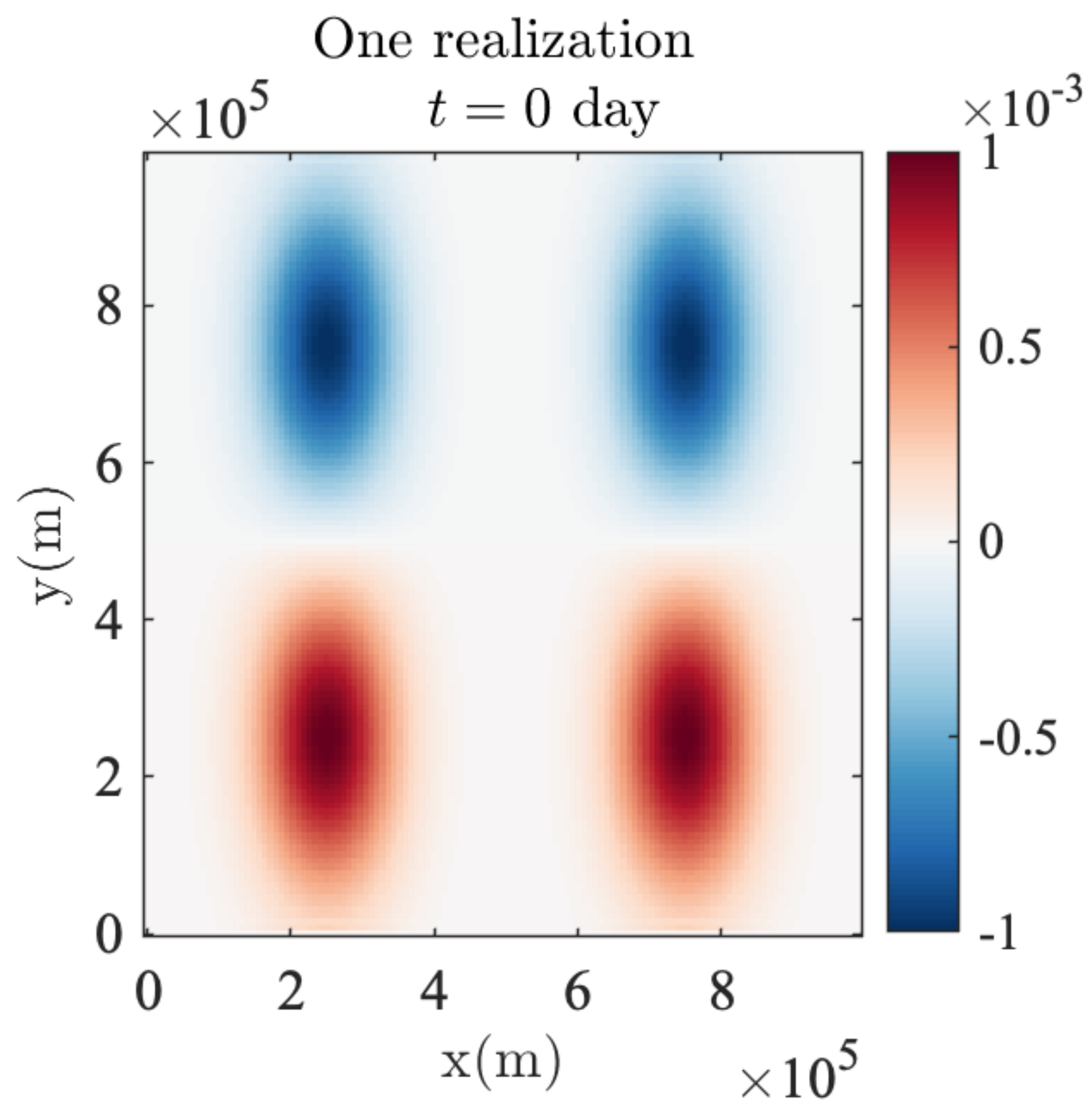
Remark: if G is symmetric (i.e. $G^{m,k} = G^{k,m}$), then the Lévy area is not necessary:

$$\sum_{m,k} G^{m,k} I^{m,k} = \frac{1}{2}\sum_{m,k} G^{m,k}I^{m,k} + G^{k,m}I^{k,m} = \sum_{m,k} G^{m,k}\frac{I^{m,k}+I^{k,m}}{2}$$

Euler Maruyama

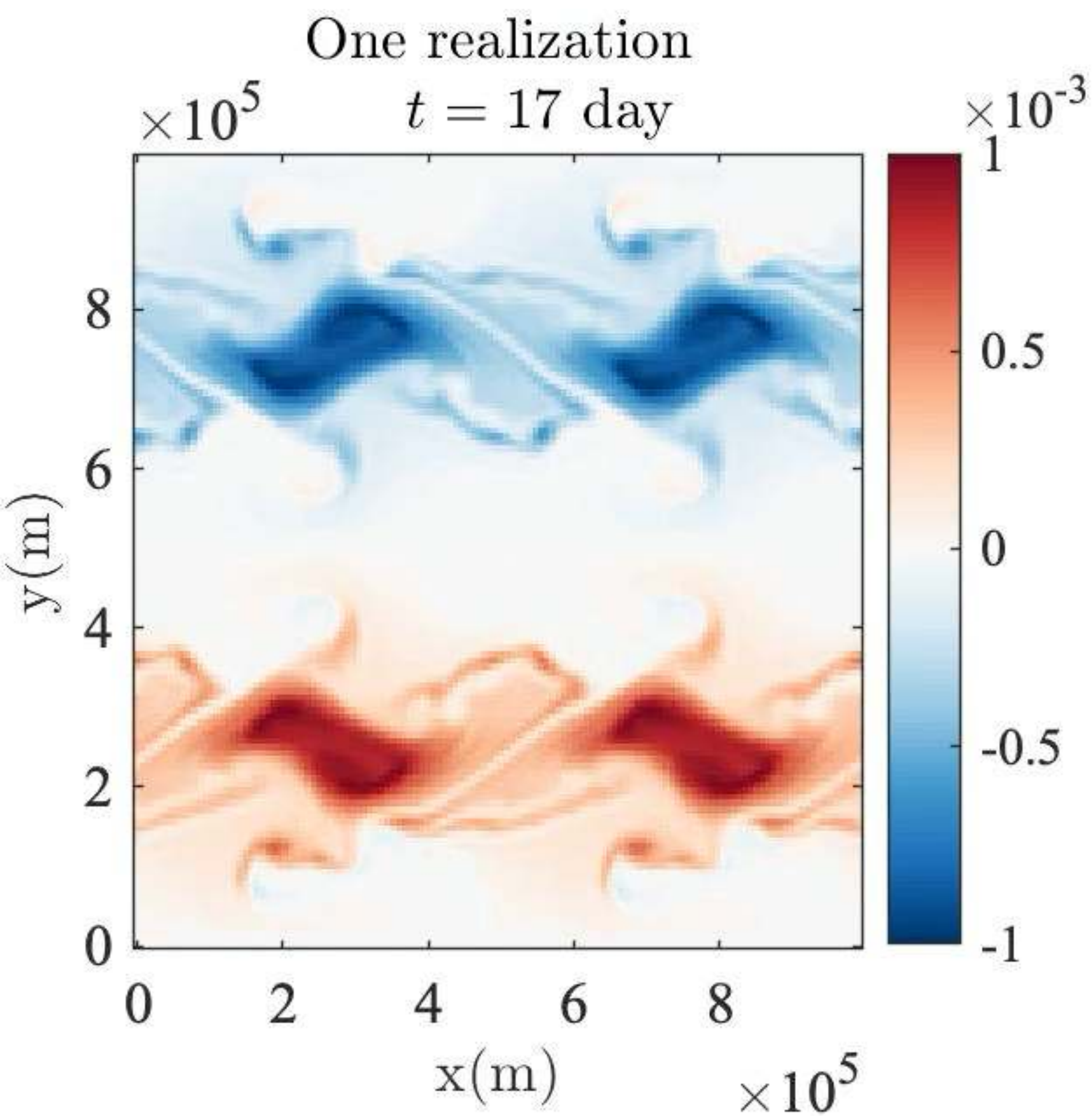


Milstein - weak

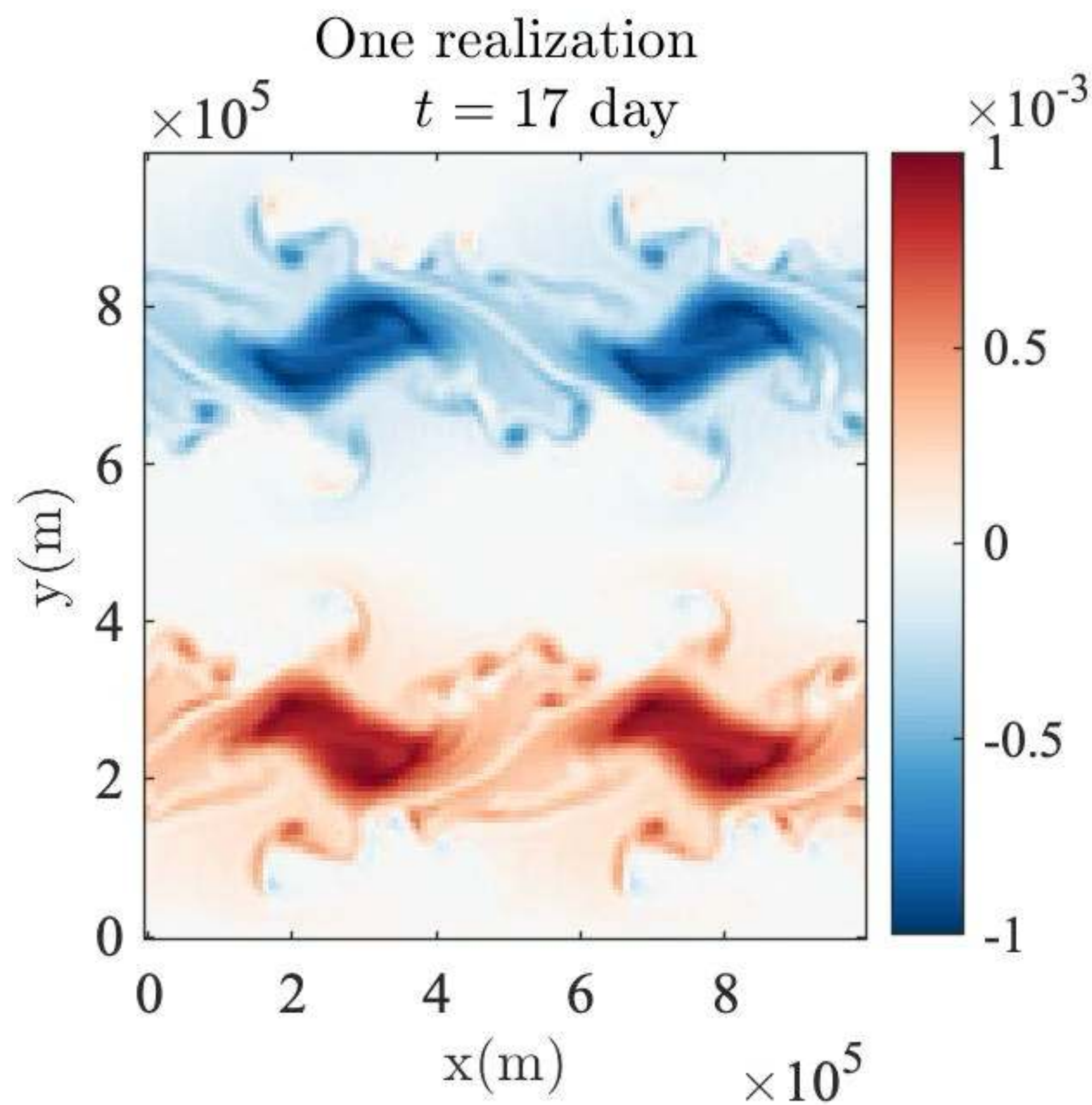


Numerical results

Euler Maruyama



Milstein - weak

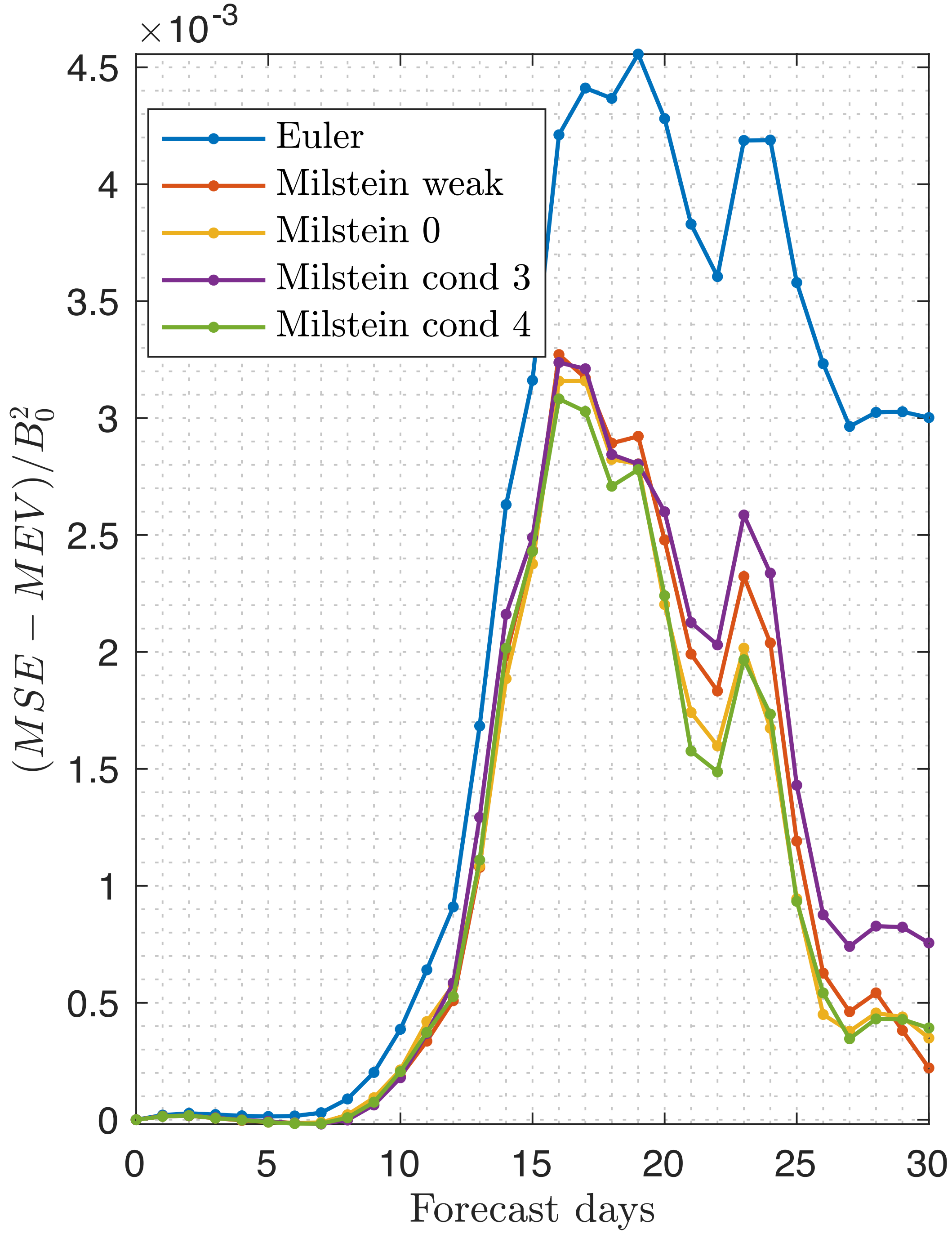


Numerical results

A necessary condition for ensemble reliability is that the mean squared error (MSE) of the ensemble mean forecast is close to the mean ensemble variance (MEV):

$$\text{MSE} = \sum_{n=1}^{N} \left(\hat{\mathbb{E}}(b_h(\mathbf{x}_n, t)) - b^{obs}(\mathbf{x}_n, t) \right)^2$$

$$\simeq \text{MEV} = \left(\frac{1}{N} \sum_{n=1}^{N} \hat{Var}(b_h(\mathbf{x}_n, t)) \right)$$



Multi-step schemes

The final aim being to use Milstein scheme in a multi-step Runge-Kutta type method, we started studying Runge-Kutta methods in the stochastic framework, starting with SSPRK3 [3] and Heun [4].

First, we rewrite the system in Stratonovich form:

$$\begin{cases} \mathrm{d}_t b = f_s(b,u) + g_s(b) \circ \mathrm{d}B_t \\ u = -\kappa \nabla^{\perp} \Delta^{-1/2} b =: \mathscr{H}(b) \end{cases}$$

SSPRK3 [3]

$$\begin{cases} b^{(1)} = b^n + f_s(b^n, u^n)\Delta t + g_s(b^n)\Delta B^n \\ u^{(1)} = \mathscr{H}(b^{(1)}) \\ b^{(2)} = \frac{3}{4} b^n + \frac{1}{4}\left(b^{(1)} + f_s(b^{(1)}, u^{(1)})\Delta t + g_s(b^{(1)})\Delta B^n\right) \\ u^{(2)} = \mathscr{H}(b^{(2)}) \\ b^{n+1} = \frac{1}{3} b^n + \frac{2}{3}\left(b^{(2)} + f_s(b^{(2)}, u^{(2)})\Delta t + g_s(b^{(2)})\Delta B^n\right) \end{cases}$$

Heun [4]

$$\begin{cases} b^{(1)} = b^n + f_s(b^n, u^n)\Delta t + g_s(b^n)\Delta B^n \\ u^{(1)} = \mathscr{H}(b^{(1)}) \\ b^{n+1} = \frac{1}{2} b^n + \frac{1}{2}\left(b^{(1)} + f_s(b^{(1)}, u^{(1)})\Delta t + g_s(b^{(1)})\Delta B^n\right) \end{cases}$$

[3] Numerically modeling stochastic Lie transport in fluid dynamics, Multiscale Modeling & Simulation 17.1 (2019): 192-232. C. Cotter, D. Crisan, D. Holm, W. Pan and I. Shevchenko.
[4] Modelling uncertainty using stochastic transport noise in a 2-layer quasi-geostrophic model. Foundations of Data Science, 2.2 (2020). C. Cotter, D. Crisan, D. Holm, W. Pan and I. Shevchenko.

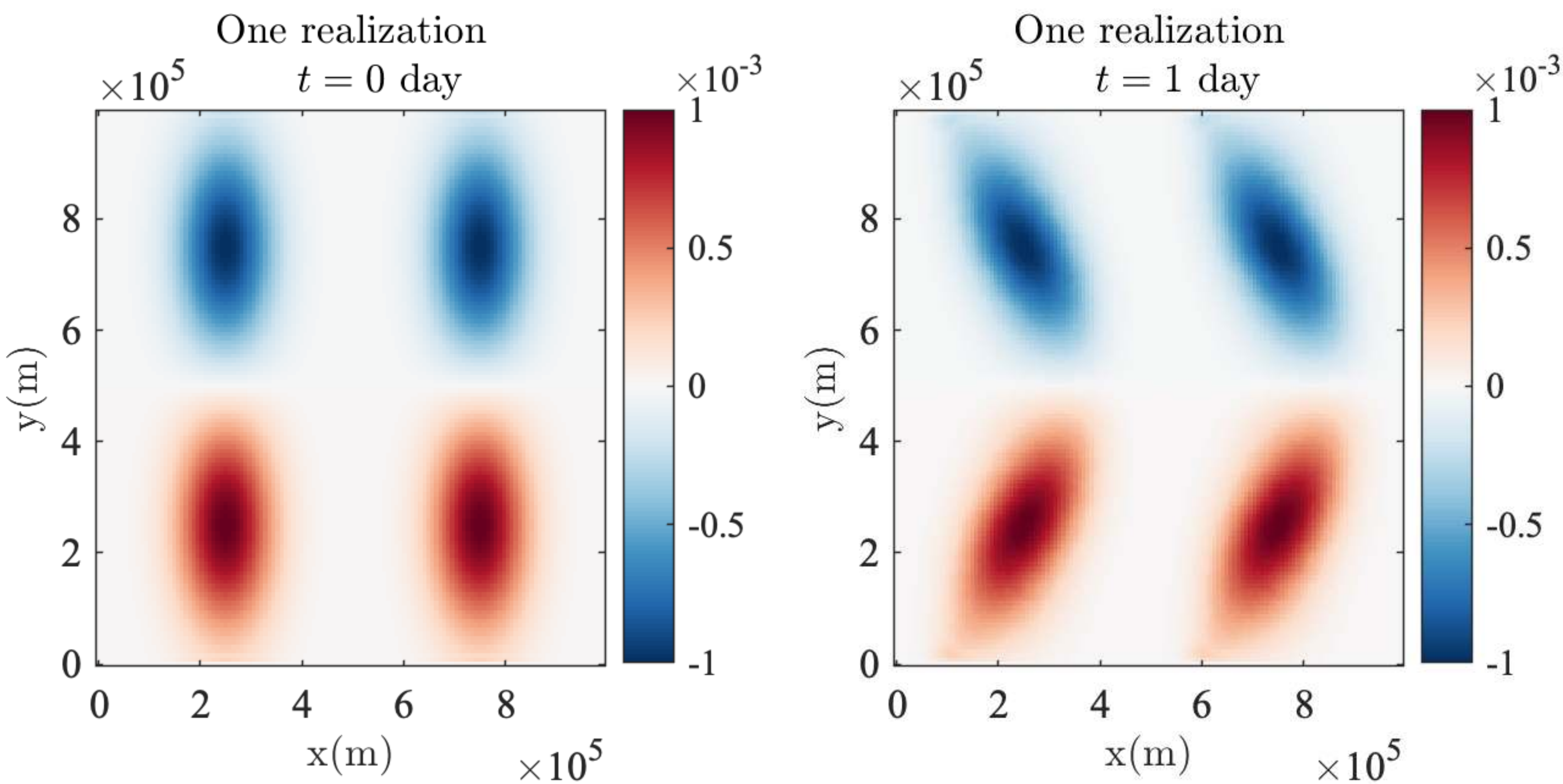
Multi-step scheme based on Milstein

In this work, we propose a two-step scheme, SRK2-EM, where the first step is done with a Milstein-0 scheme, and the second one with an Euler-Maruyama method:

$$\begin{cases} b^{(1)} = b^n + f(b^n, u^n)\Delta t + \sum_m g^m(b^n)\Delta B_n^m + \sum_{m,k} g^m(g^k(b^n))(\Delta\beta_n^m \Delta\beta_n^k - \Delta t \delta_{mk}) \\ u^{(1)} = \mathscr{H}(b^{(1)}) \\ b^{n+1} = \frac{1}{2}b^n + \frac{1}{2}\left(b^{(1)} + f(b^{(1)}, u^{(1)})\Delta t + \sum_m g^m(b^{(1)})\Delta B_n^m \right) \end{cases}$$

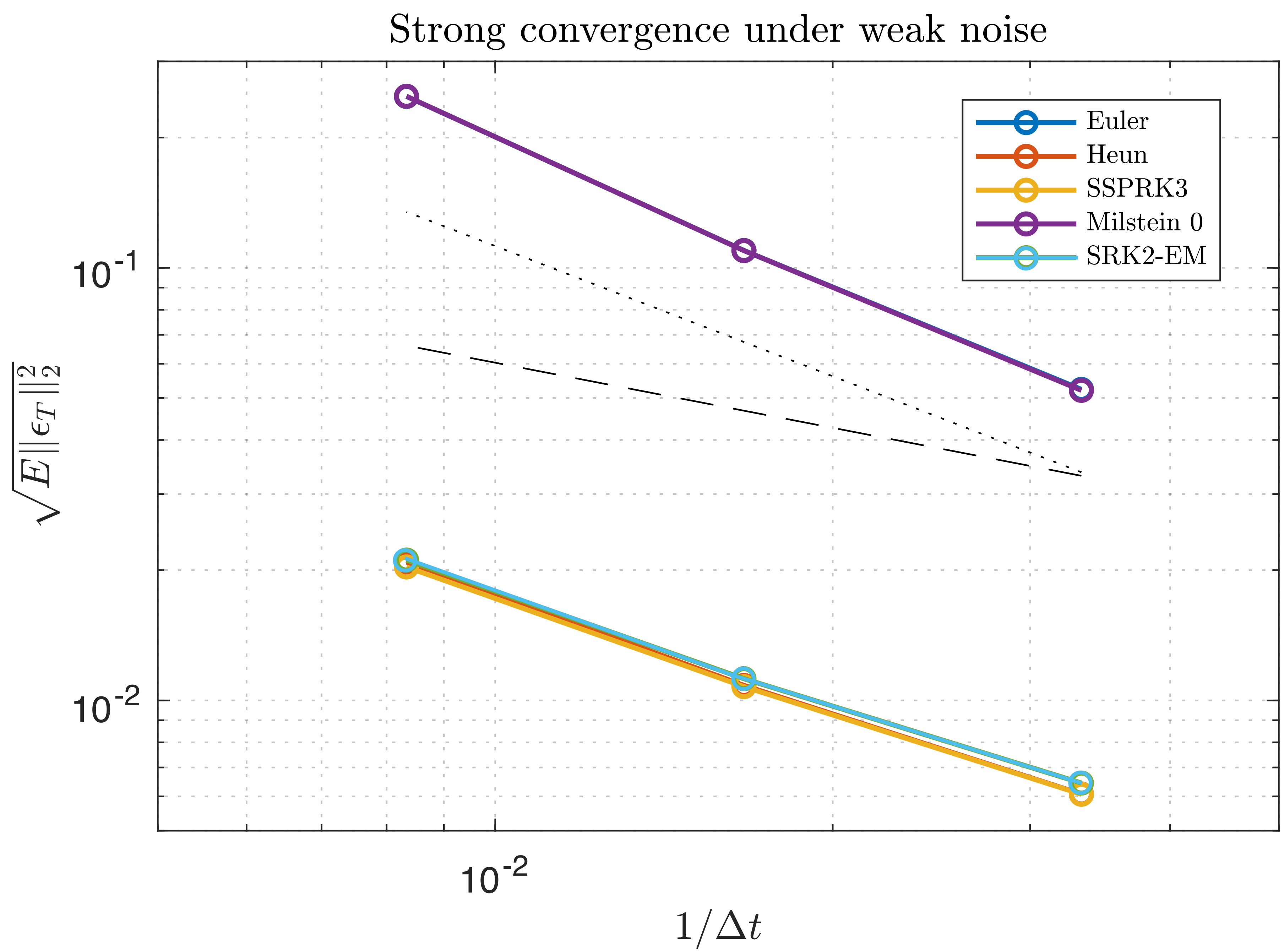
Numerical results

Euler Maruyama

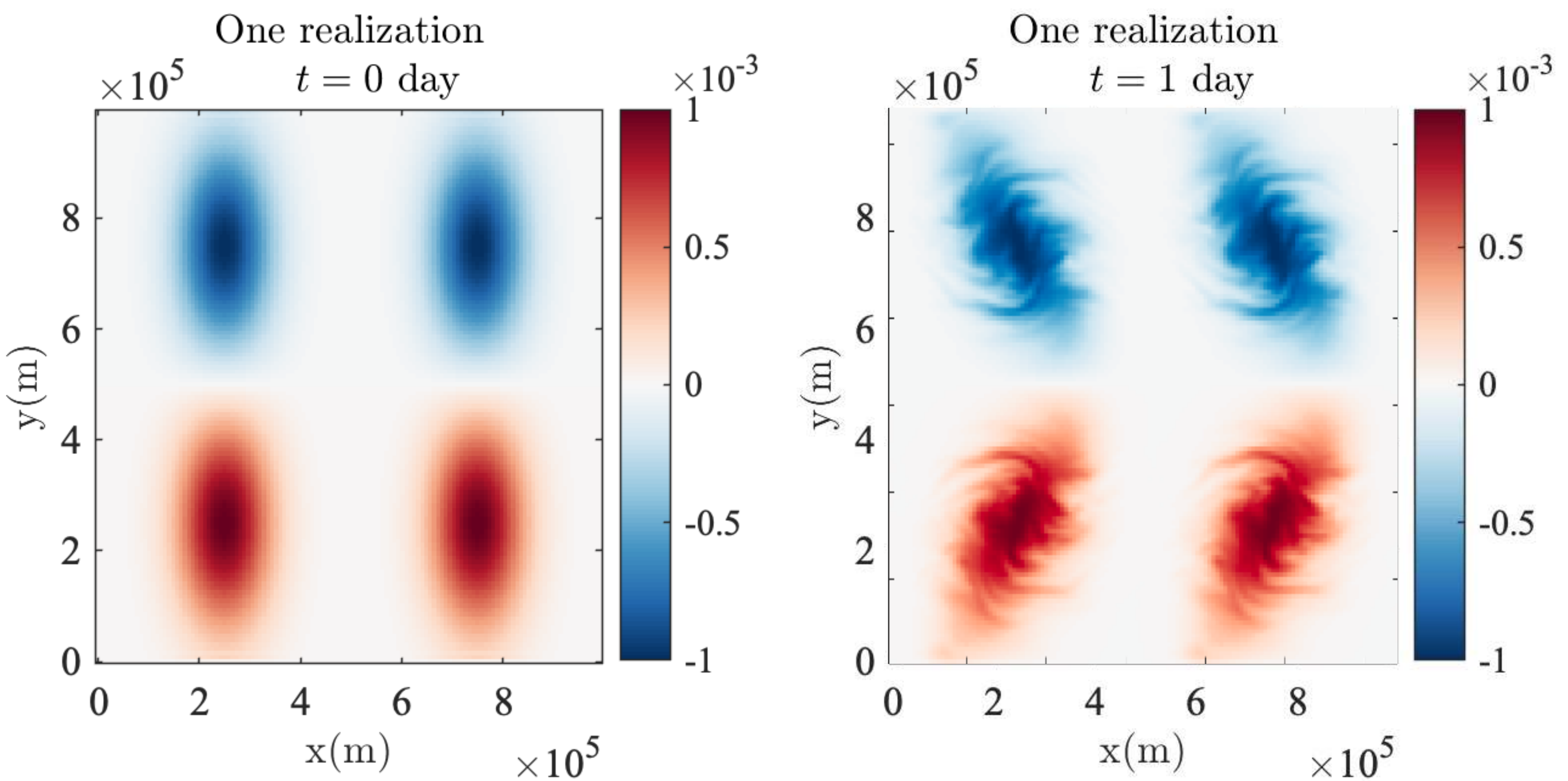


Numerical results

$$\mathbb{E}\left[\|b(T,\mathbf{x}) - b_h(n\Delta t,\mathbf{x})\|\right] \leq C\Delta t^{\gamma}$$

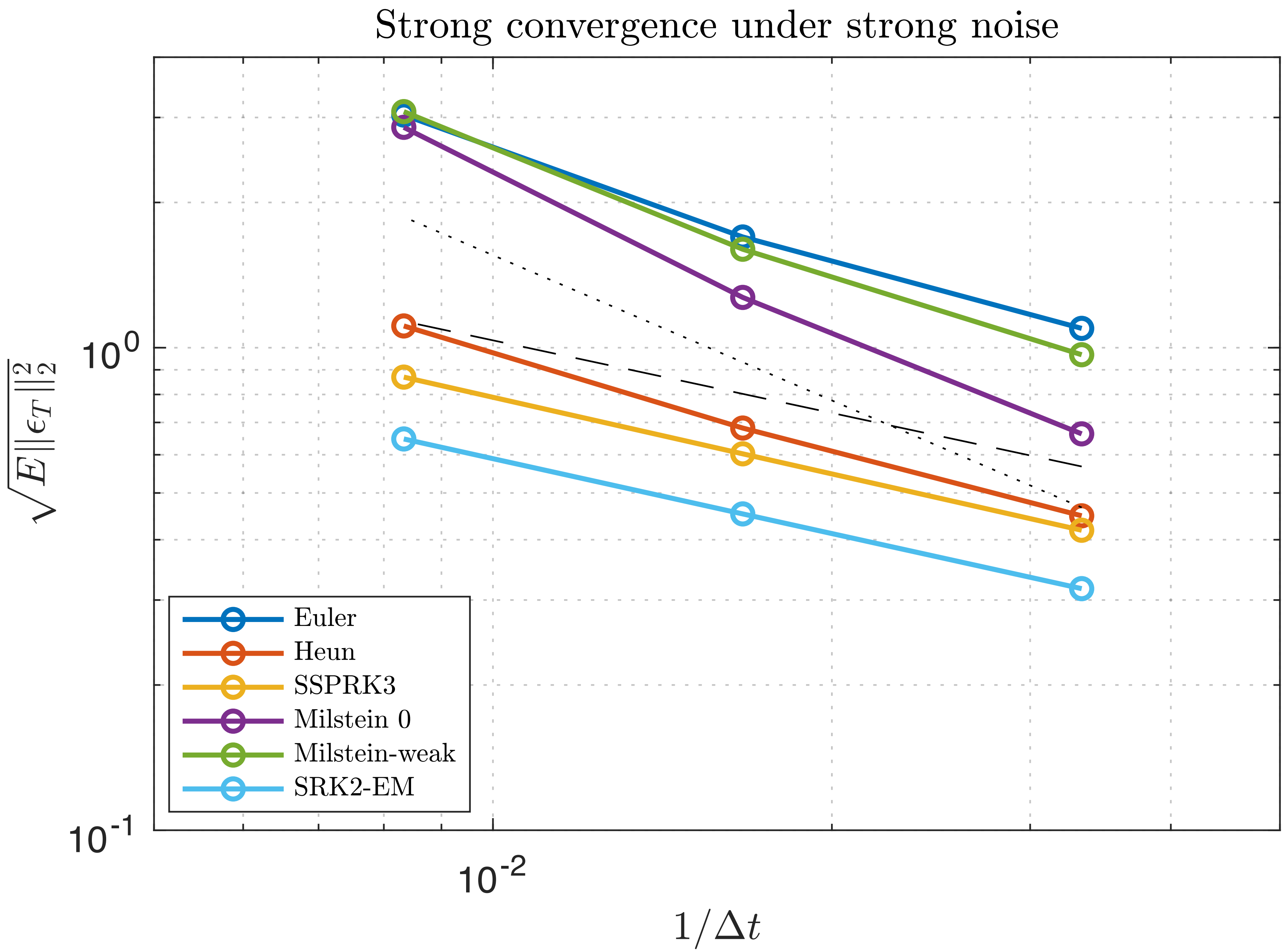


Euler Maruyama



Numerical results

$$\mathbb{E}\left[\|b(T,\mathbf{x}) - b_h(n\Delta t,\mathbf{x})\|\right] \leq C\Delta t^\gamma$$



Merci de votre attention !