A modified ROM-based parareal method for the simulation of urban floods

J. G. Caldas Steinstraesser [1,2]   V. Guinot [1,3]   A. Rousseau [1,2]

[1] Inria - LEMON team   [2] IMAG, Univ. Montpellier   [3] HSM, Univ. Montpellier

14 January 2021

Table of Contents

Table of Contents

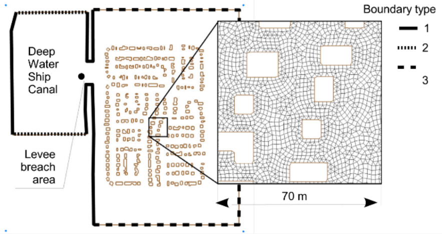
Introduction

- Numerical simulation of urban floods using the shallow water equations (SWE);

Introduction

- Numerical simulation of urban floods using the shallow water equations (SWE);
- Accurate results: **high computational cost**;

[Guinot et al. (2017)]



[Guinot et al. (2017)]

Introduction

- Alternative: coarser models

Introduction

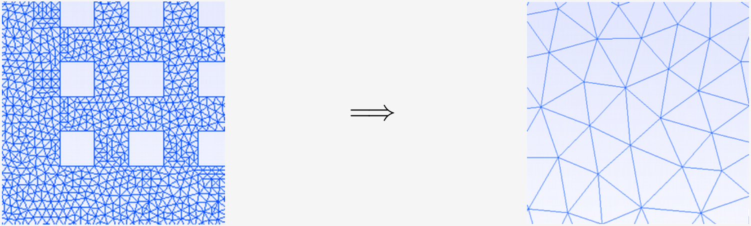
- Alternative: coarser models
  - Coarser SWE

Introduction

- Alternative: coarser models
  - Coarser SWE
  - Porosity-based SWE [Defina (2000); Guinot and Soares-Frazão (2006), ...]

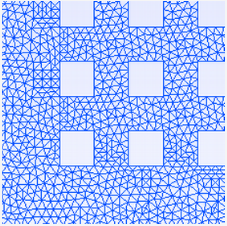
Introduction

- Alternative: coarser models
  - Coarser SWE
  - Porosity-based SWE [Defina (2000); Guinot and Soares-Frazão (2006), ...]
- Coarser mesh, larger time step $\Longrightarrow$ smaller computational cost (2-3 orders of magnitude);

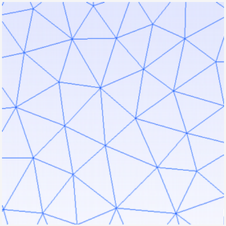


Introduction

- Alternative: coarser models
  - Coarser SWE
  - Porosity-based SWE [Defina (2000); Guinot and Soares-Frazão (2006), ...]
- Coarser mesh, larger time step $\Longrightarrow$ smaller computational cost (2-3 orders of magnitude);
- Good global approximations, but less accurate inside the urban zone (zones of interest)



$\Longrightarrow$



Objectives and methodology

- Objective: **coupled model** between the SWE at different scales:

Objectives and methodology

- Objective: **coupled model** between the SWE at different scales:
  - More accurate results inside the urban zone;
  - Reduced computational cost compared to SWE.

Objectives and methodology

- Objective: **coupled model** between the SWE at different scales:
  - More accurate results inside the urban zone;
  - Reduced computational cost compared to SWE.
- **Parareal method**:

Objectives and methodology

- Objective: **coupled model** between the SWE at different scales:
  - More accurate results inside the urban zone;
  - Reduced computational cost compared to SWE.
- **Parareal method**:
  - Objective: reduce the computational cost of a fine model;

Objectives and methodology

- Objective: **coupled model** between the SWE at different scales:
  - More accurate results inside the urban zone;
  - Reduced computational cost compared to SWE.
- **Parareal method**:
  - Objective: reduce the computational cost of a fine model;
  - Simultaneous use of a fine and a coarse model;

Objectives and methodology

- Objective: **coupled model** between the SWE at different scales:
  - More accurate results inside the urban zone;
  - Reduced computational cost compared to SWE.
- **Parareal method**:
  - Objective: reduce the computational cost of a fine model;
  - Simultaneous use of a fine and a coarse model;
  - Parallelize the fine simulation in time;

Objectives and methodology

- Objective: **coupled model** between the SWE at different scales:
  - More accurate results inside the urban zone;
  - Reduced computational cost compared to SWE.
- **Parareal method**:
  - Objective: reduce the computational cost of a fine model;
  - Simultaneous use of a fine and a coarse model;
  - Parallelize the fine simulation in time;
  - Adaptations for solving hyperbolic problems: use of **reduced-order models (ROMs)**;

Objectives and methodology

- Objective: **coupled model** between the SWE at different scales:
  - More accurate results inside the urban zone;
  - Reduced computational cost compared to SWE.
- **Parareal method**:
  - Objective: reduce the computational cost of a fine model;
  - Simultaneous use of a fine and a coarse model;
  - Parallelize the fine simulation in time;
  - Adaptations for solving hyperbolic problems: use of **reduced-order models (ROMs)**;
  - Further improvements

Table of Contents

The parareal method: definitions

[Lions et al. (2001)]

- A simple problem:

$$\begin{cases} \frac{d}{dt}\boldsymbol{y}(t) + A\boldsymbol{y}(t) = 0, \quad \text{in } [0, T] \\ \boldsymbol{y}(0) = \boldsymbol{y}_0 \end{cases} \tag{1}$$

The parareal method: definitions

[Lions et al. (2001)]

- A simple problem:

$$\begin{cases} \frac{d}{dt}\boldsymbol{y}(t) + A\boldsymbol{y}(t) = 0, \quad \text{in } [0, T] \\ \boldsymbol{y}(0) = \boldsymbol{y}_0 \end{cases} \tag{1}$$

- $\mathcal{F}_{\delta t}$: A **fine discretization (propagator)** of (1)
  - Time step δt
  - Accurate but too expensive

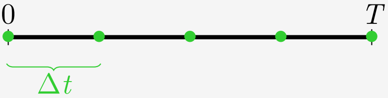
The parareal method: definitions

[Lions et al. (2001)]

- A simple problem:

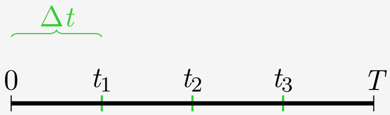
$$\begin{cases} \frac{d}{dt}\boldsymbol{y}(t) + A\boldsymbol{y}(t) = 0, & \text{in } [0, T] \\ \boldsymbol{y}(0) = \boldsymbol{y}_0 \end{cases} \tag{1}$$

- $\mathcal{F}_{\delta t}$: A **fine discretization (propagator)** of (1)
  - Time step δt
  - Accurate but too expensive
- $\mathcal{G}_{\Delta t}$: A **coarser discretization (propagator)** of (1)
  - Time step $\Delta t > \delta t$
  - Much cheaper but inaccurate

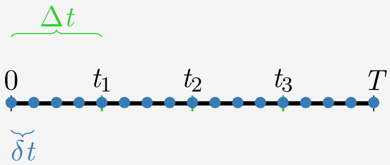


The parareal method: construction

The parareal method: construction

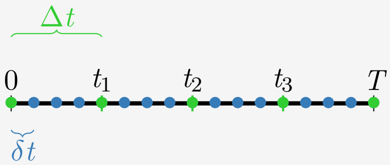


The parareal method: construction



The parareal method: construction

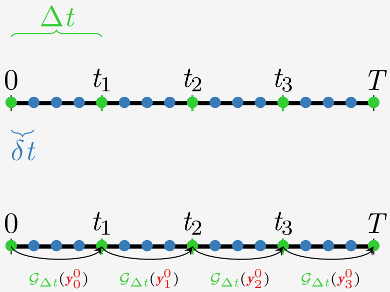
- Predictor-corrector iterative method;
- $\boldsymbol{y}_n^k$: solution at instant t_n and iteration k.



The parareal method: construction

- Predictor-corrector iterative method;
- $\boldsymbol{y}_n^k$: solution at instant t_n and iteration k.
- **Initial prediction** $(k = 0)$:

$$\boldsymbol{y}_{n+1}^0 = \mathcal{G}_{\Delta t}(\boldsymbol{y}_n^0) \qquad \text{(seq.)}$$



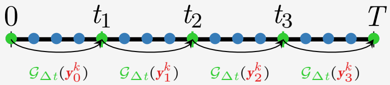
The parareal method: construction

- **Iteration $k + 1$:**
  - $\boldsymbol{y}_n^k$ available for all $n = 0, \ldots, N_{\Delta t}$

The parareal method: construction

- **Iteration $k+1$**:
  - $\boldsymbol{y}_n^k$ available for all $n = 0, \ldots, N_{\Delta t}$
  - Coarse prediction (sequentially):

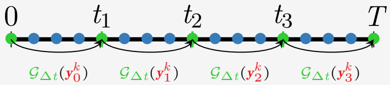
$$\boldsymbol{y}_{n+1}^{k+1} = \underbrace{\mathcal{G}_{\Delta t}(\boldsymbol{y}_n^{k+1})}_{\text{prediction (seq.)}}$$



The parareal method: construction

- **Iteration $k+1$**:
  - $\boldsymbol{y}_n^k$ available for all $n = 0, \ldots, N_{\Delta t}$
  - Coarse prediction (sequentially):

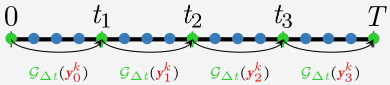
$$\boldsymbol{y}_{n+1}^{k+1} = \underbrace{\mathcal{G}_{\Delta t}(\boldsymbol{y}_n^{k+1})}_{\text{prediction (seq.)}} + \underbrace{\mathcal{F}_{\delta t}(\boldsymbol{y}_n^k) - \mathcal{G}_{\Delta t}(\boldsymbol{y}_n^k)}_{\text{correction (par.)}}$$



The parareal method: construction

- **Iteration $k+1$:**
  - y_n^k available for all $n = 0, \ldots, N_{\Delta t}$
  - Coarse prediction (sequentially):

$$y_{n+1}^{k+1} = \underbrace{\mathcal{G}_{\Delta t}(y_n^{k+1})}_{\text{prediction (seq.)}} + \underbrace{\mathcal{F}_{\delta t}(y_n^k) - \mathcal{G}_{\Delta t}(y_n^k)}_{\text{correction (par.)}}$$

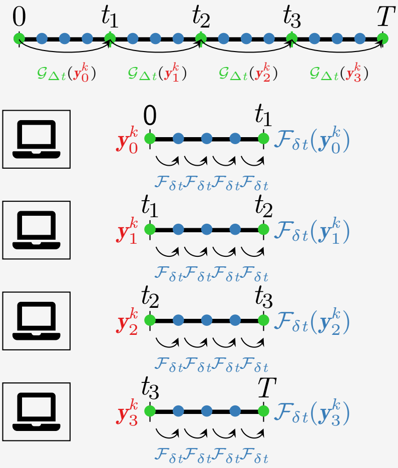


The parareal method: construction

- **Iteration $k+1$**:

$$\boldsymbol{y}_{n+1}^{k+1} = \underbrace{\mathcal{G}_{\Delta t}(\boldsymbol{y}_n^{k+1})}_{\text{prediction (seq.)}} + \underbrace{\mathcal{F}_{\delta t}(\boldsymbol{y}_n^{k}) - \mathcal{G}_{\Delta t}(\boldsymbol{y}_n^{k})}_{\text{correction (par.)}}$$

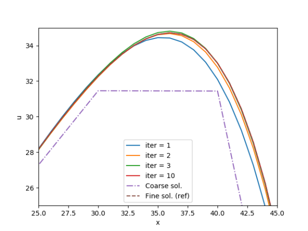
  - $\boldsymbol{y}_n^k$ available for all $n = 0, \ldots, N_{\Delta t}$
  - Coarse prediction (sequentially):
  - Fine correction (in parallel):



Parareal method: performance

- Fast convergence for many problems;
  - Parabolic, diffusive problems

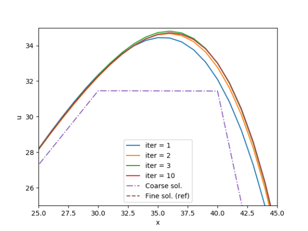
$$u_t + u_{xx} = f$$



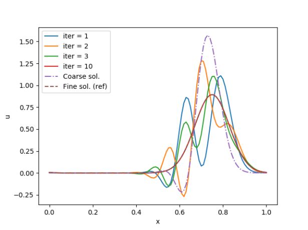
Parareal method: performance

- Fast convergence for many problems;
  - Parabolic, diffusive problems
- In the case of hyperbolic problems:
  - Slow convergence
  - Instabilities

$$u_t + u_{xx} = f$$



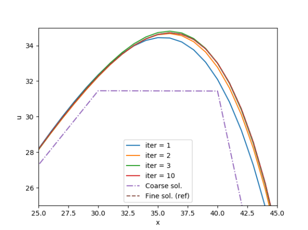
$$u_t + u_x = 0$$



Parareal method: performance

- Fast convergence for many problems;
  - Parabolic, diffusive problems
- In the case of hyperbolic problems:
  - Slow convergence
  - Instabilities
- Causes [Ruprecht (2018)]:
  - Mismatch of discrete phase speeds between $\mathcal{F}_{\delta t}$ and $\mathcal{G}_{\Delta t}$
  - Mainly on high wave numbers (damped in parabolic problems);

$$u_t + u_{xx} = f$$



$$u_t + u_x = 0$$

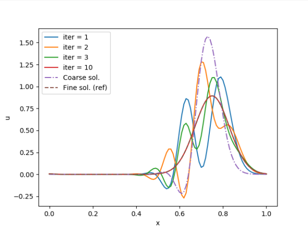


Table of Contents

Parareal methods for nonlinear hyperbolic problems

[Barrault et al. (2004); Chaturantabut and Sorensen (2010); Chen et al. (2014)]

- The coarser model is replaced by a **reduced-order model (ROM)**, constructed from snapshots of the solution.

Parareal methods for nonlinear hyperbolic problems

[Barrault et al. (2004); Chaturantabut and Sorensen (2010); Chen et al. (2014)]

- The coarser model is replaced by a **reduced-order model (ROM)**, constructed from snapshots of the solution.

$$\frac{dy_1}{dt} = Ay_1 + F(y_1)$$
$$\frac{dy_2}{dt} = Ay_2 + F(y_2)$$
$$\vdots$$
$$\vdots$$
$$\frac{dy_N}{dt} = Ay_N + F(y_N)$$

Parareal methods for nonlinear hyperbolic problems

[Barrault et al. (2004); Chaturantabut and Sorensen (2010); Chen et al. (2014)]

- The coarser model is replaced by a **reduced-order model (ROM)**, constructed from snapshots of the solution.

$$\frac{dy_1}{dt} = Ay_1 + F(y_1)$$
$$\frac{dy_2}{dt} = Ay_2 + F(y_2)$$
$$\vdots$$
$$\vdots$$
$$\frac{dy_N}{dt} = Ay_N + F(y_N)$$

Snapshots

$y(t_1),\ y(t_2), \ldots, y(t_s)$ $\Longrightarrow$

Parareal methods for nonlinear hyperbolic problems

[Barrault et al. (2004); Chaturantabut and Sorensen (2010); Chen et al. (2014)]

- The coarser model is replaced by a **reduced-order model (ROM)**, constructed from snapshots of the solution.

$$\frac{dy_1}{dt} = Ay_1 + F(y_1)$$
$$\frac{dy_2}{dt} = Ay_2 + F(y_2)$$
$$\vdots$$
$$\vdots$$
$$\frac{dy_N}{dt} = Ay_N + F(y_N)$$

Snapshots

$y(t_1),\ y(t_2), \dots, y(t_s)$ $\Longrightarrow$

$$\frac{d\widetilde{y}_1}{dt} = \widetilde{A}\widetilde{y}_1 + \widetilde{F}(\widetilde{y}_1)$$
$$\vdots$$
$$\frac{d\widetilde{y}_q}{dt} = \widetilde{A}\widetilde{y}_q + \widetilde{F}(\widetilde{y}_q)$$

$q \ll N$

Parareal methods for nonlinear hyperbolic problems

[Barrault et al. (2004); Chaturantabut and Sorensen (2010); Chen et al. (2014)]

- The coarser model is replaced by a **reduced-order model (ROM)**, constructed from snapshots of the solution.
  - **POD**: proper orthogonal decomposition (reduction of linear term)
    - Approximation in a space $\mathcal{S}_q$ with dimension $q \ll N$ (SVD);

$$\frac{dy_1}{dt} = Ay_1 + F(y_1)$$
$$\frac{dy_2}{dt} = Ay_2 + F(y_2)$$
$$\vdots$$
$$\vdots$$
$$\frac{dy_N}{dt} = Ay_N + F(y_N)$$

Snapshots

$y(t_1),\ y(t_2), \ldots, y(t_s)$ $\Longrightarrow$

$$\frac{d\widetilde{y}_1}{dt} = \widetilde{A}\widetilde{y}_1 + \widetilde{F}(\widetilde{y}_1)$$
$$\vdots$$
$$\frac{d\widetilde{y}_q}{dt} = \widetilde{A}\widetilde{y}_q + \widetilde{F}(\widetilde{y}_q)$$

$q \ll N$

Parareal methods for nonlinear hyperbolic problems

[Barrault et al. (2004); Chaturantabut and Sorensen (2010); Chen et al. (2014)]

- The coarser model is replaced by a **reduced-order model (ROM)**, constructed from snapshots of the solution.
  - **POD**: proper orthogonal decomposition (reduction of linear term)
    - Approximation in a space $\mathcal{S}_q$ with dimension $q \ll N$ (SVD);
  - **EIM/POD-DEIM:** (discrete) empirical interpolation method (reduction of nonlinear term)
    - Approximation in a space $\widehat{\mathcal{S}}_m$ with dimension $m \ll N$;
    - Interpolation from m points in space.

$$\frac{dy_1}{dt} = Ay_1 + F(y_1)$$
$$\frac{dy_2}{dt} = Ay_2 + F(y_2)$$
$$\vdots$$
$$\vdots$$
$$\frac{dy_N}{dt} = Ay_N + F(y_N)$$

Snapshots

$y(t_1),\ y(t_2), \dots, y(t_s)$ $\Longrightarrow$

$$\frac{d\widetilde{y}_1}{dt} = \widetilde{A}\widetilde{y}_1 + \widetilde{F}(\widetilde{y}_1)$$
$$\vdots$$
$$\frac{d\widetilde{y}_q}{dt} = \widetilde{A}\widetilde{y}_q + \widetilde{F}(\widetilde{y}_q)$$

$q \ll N$

The ROM-based parareal method

- How is the ROM introduced in the **parareal method**?

The ROM-based parareal method

- How is the ROM introduced in the **parareal method**?
  - $\mathcal{G}_{\Delta t}$ is replaced by the ROM $\mathcal{F}^k_{\delta t,r}$, solved with a fine time step δt:

The ROM-based parareal method

- How is the ROM introduced in the **parareal method**?
  - $\mathcal{G}_{\Delta t}$ is replaced by the ROM $\mathcal{F}^{k}_{\delta t,r}$, solved with a fine time step δt:

$$\boldsymbol{y}_{n+1}^{k+1} = \mathcal{G}_{\Delta t}(\boldsymbol{y}_n^{k+1}) + \mathcal{F}_{\delta t}(\boldsymbol{y}_n^k) - \mathcal{G}_{\Delta t}(\boldsymbol{y}_n^k)$$

The ROM-based parareal method

- How is the ROM introduced in the **parareal method**?
  - $\mathcal{G}_{\Delta t}$ is replaced by the ROM $\mathcal{F}^k_{\delta t,r}$, solved with a fine time step δt:

$$\boldsymbol{y}^{k+1}_{n+1} = \mathcal{F}^k_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^{k+1}_n) + \mathcal{F}_{\delta t}(\boldsymbol{y}^k_n) - \mathcal{F}^k_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^k_n)$$

The ROM-based parareal method

- How is the ROM introduced in the **parareal method**?
  - $\mathcal{G}_{\Delta t}$ is replaced by the ROM $\mathcal{F}^k_{\delta t,r}$, solved with a fine time step δt:

$$\boldsymbol{y}^{k+1}_{n+1} = \mathcal{F}^k_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^{k+1}_n) + \mathcal{F}_{\delta t}(\boldsymbol{y}^k_n) - \mathcal{F}^k_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^k_n)$$

  - The coarse model $\mathcal{G}_{\Delta t}$ is still used for the initial prediction $(k = 0)$.

The ROM-based parareal method

- How is the ROM introduced in the **parareal method**?
  - $\mathcal{G}_{\Delta t}$ is replaced by the ROM $\mathcal{F}^k_{\delta t,r}$, solved with a fine time step δt:

$$\boldsymbol{y}^{k+1}_{n+1} = \mathcal{F}^k_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^{k+1}_n) + \mathcal{F}_{\delta t}(\boldsymbol{y}^k_n) - \mathcal{F}^k_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^k_n)$$

  - The coarse model $\mathcal{G}_{\Delta t}$ is still used for the initial prediction $(k = 0)$.
  - The ROM is reformulated **on-the-fly** at each iteration, using the solutions from all coarse time steps of all previous iterations.

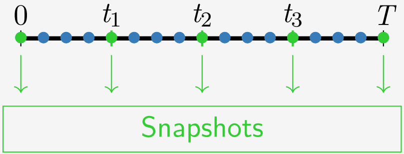


Table of Contents

Improvement of the ROM-based parareal method

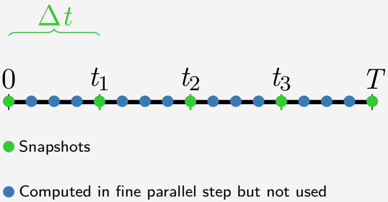
- **Limitation:** quality of the ROM
  - Imprecise snapshots

Improvement of the ROM-based parareal method

- **Limitation:** quality of the ROM
  - Imprecise snapshots
- **Enrichment of the snapshots sets:**

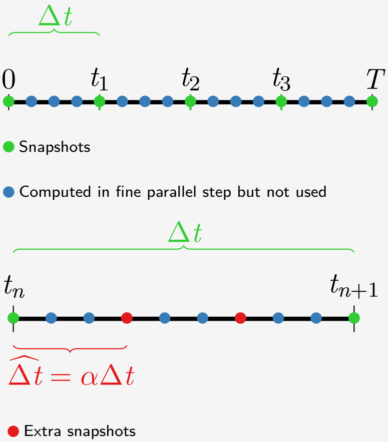
Improvement of the ROM-based parareal method

- **Limitation:** quality of the ROM
  - Imprecise snapshots
- **Enrichment of the snapshots sets:**
  - Extra snapshots are available for free;



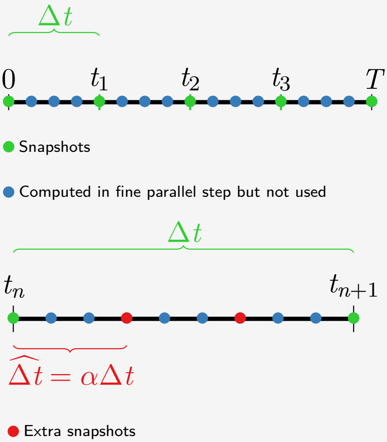
Improvement of the ROM-based parareal method

- **Limitation:** quality of the ROM
  - Imprecise snapshots
- **Enrichment of the snapshots sets:**
  - Extra snapshots are available for free;
  - Take extra snapshots at every $\widehat{\Delta t} = \alpha \Delta t$, $\delta t \leq \widehat{\Delta t} \leq \Delta t$.



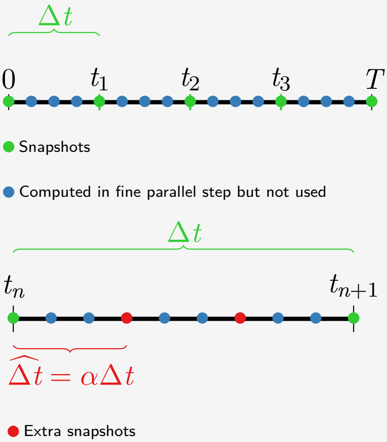
Improvement of the ROM-based parareal method

- **Limitation:** quality of the ROM
  - Imprecise snapshots
- **Enrichment of the snapshots sets:**
  - Extra snapshots are available for free;
  - Take extra snapshots at every $\widehat{\Delta t} = \alpha \Delta t$, $\delta t \leq \widehat{\Delta t} \leq \Delta t$.
- How many extra snapshots?



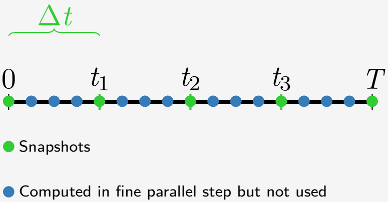
Improvement of the ROM-based parareal method

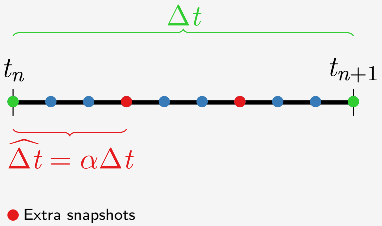
- **Limitation:** quality of the ROM
  - Imprecise snapshots
- **Enrichment of the snapshots sets:**
  - Extra snapshots are available for free;
  - Take extra snapshots at every $\widehat{\Delta t} = \alpha \Delta t$, $\delta t \leq \widehat{\Delta t} \leq \Delta t$.
- How many extra snapshots?
  - POD (SVD): cost $= \mathcal{O}(n^2_{\text{snapshots}}) = \mathcal{O}(1/\alpha^2)$



Improvement of the ROM-based parareal method

- **Limitation:** quality of the ROM
  - Imprecise snapshots
- **Enrichment of the snapshots sets:**
  - Extra snapshots are available for free;
  - Take extra snapshots at every $\widehat{\Delta t} = \alpha \Delta t$, $\delta t \leq \widehat{\Delta t} \leq \Delta t$.
- How many extra snapshots?
  - POD (SVD): cost $= \mathcal{O}(n^2_{\text{snapshots}}) = \mathcal{O}(1/\alpha^2)$
  - Keep α large (*e.g.* $\alpha = 1/2$).





Numerical tests - SWE

$$\frac{\partial}{\partial t}\boldsymbol{U}(t) + \frac{\partial}{\partial x}\boldsymbol{F}(\boldsymbol{U}(t)) + \frac{\partial}{\partial y}\boldsymbol{G}(\boldsymbol{U}(t)) = \boldsymbol{S}(\boldsymbol{U}(t))$$

$$\boldsymbol{U} = \begin{pmatrix} h \\ hu_x \\ hu_y \end{pmatrix}, \ \boldsymbol{F} = \begin{pmatrix} hu_x \\ hu_x^2 + gh^2/2 \\ hu_x u_y \end{pmatrix}, \ \boldsymbol{G} = \begin{pmatrix} hu_y \\ hu_x u_y \\ hu_y^2 + gh^2/2 \end{pmatrix}, \ \boldsymbol{S} = \begin{pmatrix} 0 \\ S_{0,x} + S_{f,x} \\ S_{0,y} + S_{f,y} \end{pmatrix}$$

Numerical tests - SWE

$$\frac{\partial}{\partial t}\boldsymbol{U}(t) + \frac{\partial}{\partial x}\boldsymbol{F}(\boldsymbol{U}(t)) + \frac{\partial}{\partial y}\boldsymbol{G}(\boldsymbol{U}(t)) = \boldsymbol{S}(\boldsymbol{U}(t))$$

$$\boldsymbol{U} = \begin{pmatrix} h \\ hu_x \\ hu_y \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} hu_x \\ hu_x^2 + gh^2/2 \\ hu_x u_y \end{pmatrix}, \quad \boldsymbol{G} = \begin{pmatrix} hu_y \\ hu_x u_y \\ hu_y^2 + gh^2/2 \end{pmatrix}, \quad \boldsymbol{S} = \begin{pmatrix} 0 \\ S_{0,x} + S_{f,x} \\ S_{0,y} + S_{f,y} \end{pmatrix}$$

- Explicit FV scheme;

Numerical tests - SWE

$$\frac{\partial}{\partial t}\boldsymbol{U}(t) + \frac{\partial}{\partial x}\boldsymbol{F}(\boldsymbol{U}(t)) + \frac{\partial}{\partial y}\boldsymbol{G}(\boldsymbol{U}(t)) = \boldsymbol{S}(\boldsymbol{U}(t))$$

$$\boldsymbol{U} = \begin{pmatrix} h \\ hu_x \\ hu_y \end{pmatrix},\ \boldsymbol{F} = \begin{pmatrix} hu_x \\ hu_x^2 + gh^2/2 \\ hu_x u_y \end{pmatrix},\ \boldsymbol{G} = \begin{pmatrix} hu_y \\ hu_x u_y \\ hu_y^2 + gh^2/2 \end{pmatrix},\ \boldsymbol{S} = \begin{pmatrix} 0 \\ S_{0,x} + S_{f,x} \\ S_{0,y} + S_{f,y} \end{pmatrix}$$

- Explicit FV scheme;
- Referential solution:

$$\begin{cases} \boldsymbol{y}_{\text{ref},0} = \boldsymbol{y}_0 \\ \boldsymbol{y}_{\text{ref},n+1} = \mathcal{F}_{\delta t}(\boldsymbol{y}_{\text{ref},n}) \qquad n = 0, \ldots, N_{\Delta t} - 1 \end{cases}$$

Numerical tests - SWE

$$\frac{\partial}{\partial t}\boldsymbol{U}(t) + \frac{\partial}{\partial x}\boldsymbol{F}(\boldsymbol{U}(t)) + \frac{\partial}{\partial y}\boldsymbol{G}(\boldsymbol{U}(t)) = \boldsymbol{S}(\boldsymbol{U}(t))$$

$$\boldsymbol{U} = \begin{pmatrix} h \\ hu_x \\ hu_y \end{pmatrix}, \ \boldsymbol{F} = \begin{pmatrix} hu_x \\ hu_x^2 + gh^2/2 \\ hu_x u_y \end{pmatrix}, \ \boldsymbol{G} = \begin{pmatrix} hu_y \\ hu_x u_y \\ hu_y^2 + gh^2/2 \end{pmatrix}, \ \boldsymbol{S} = \begin{pmatrix} 0 \\ S_{0,x} + S_{f,x} \\ S_{0,y} + S_{f,y} \end{pmatrix}$$

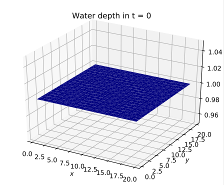
- Explicit FV scheme;
- Referential solution:

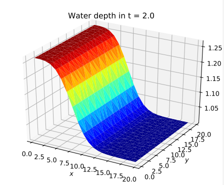
$$\begin{cases} \boldsymbol{y}_{\text{ref},0} = \boldsymbol{y}_0 \\ \boldsymbol{y}_{\text{ref},n+1} = \mathcal{F}_{\delta t}(\boldsymbol{y}_{\text{ref},n}) \qquad n = 0, \ldots, N_{\Delta t} - 1 \end{cases}$$

- Error per instant and iteration:

$$e_n^k := \frac{\sum_{i=1}^{3N} |[\boldsymbol{y}_n^k]_i - [\boldsymbol{y}_{\text{ref, n}}]_i|}{\sum_{i=1}^{3N} |[\boldsymbol{y}_{\text{ref},n}]_i|}$$

First test case (pseudo-2D)



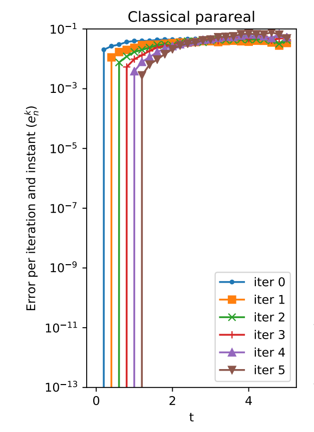


- $\Omega = [0, 20]^2$;
- Initial solution: lake-in-rest, $h(t = 0) \equiv 1$, flat bottom;
- Boundary conditions: inward unitary mass flux at $x = 0$.
- Propagators:

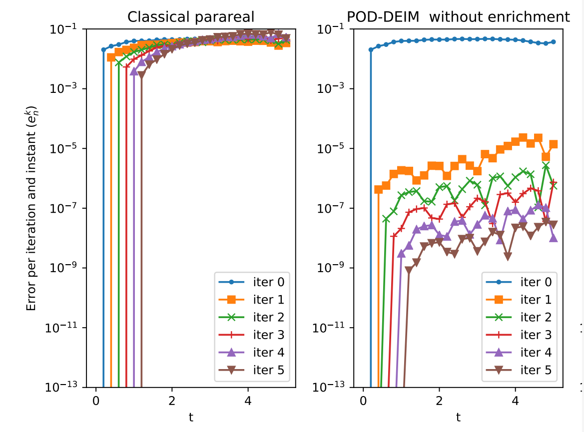
$\mathcal{F}_{\delta t}$	SWE	$\delta t = 0.001$	$\delta x = 1$	$\delta y = 1$
$\mathcal{G}_{\Delta t}$	SWE	$\Delta t = 0.2$	$\Delta x = 1$	$\Delta y = 1$

- $T = 5$, $N_{\Delta t} = 25$, $P = 20$ processors;
- Enriched snapshots: $\widehat{\Delta t} = 0.1$ $(\alpha = 1/2)$;

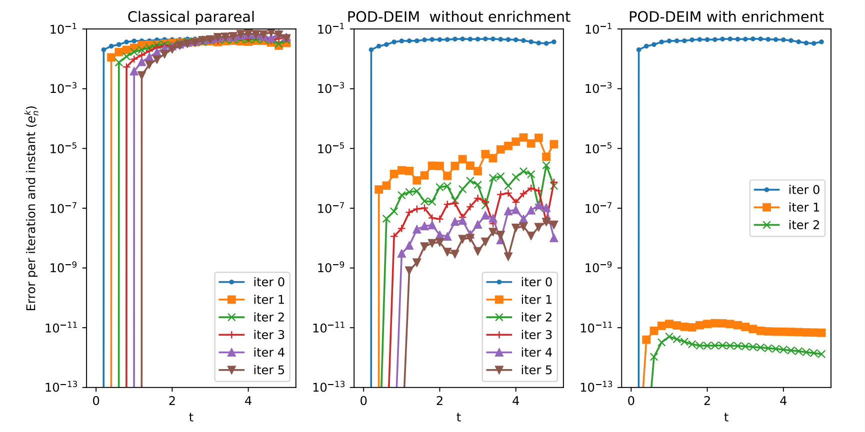
First test case (pseudo-2D): errors



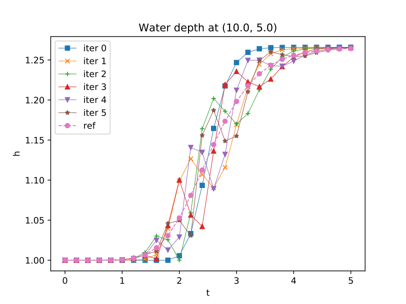
First test case (pseudo-2D): errors

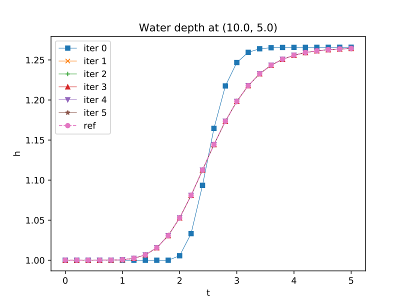


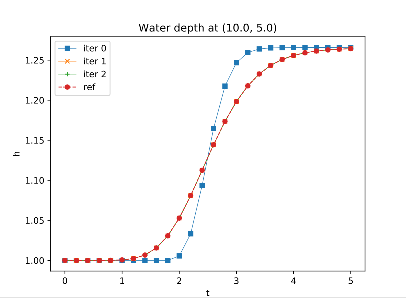
First test case (pseudo-2D): errors



First test case (pseudo-2D): h at $(x, y) = (10, 5)$

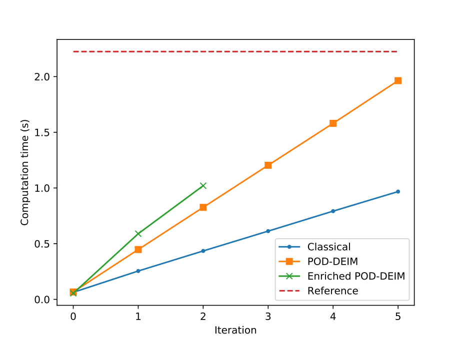


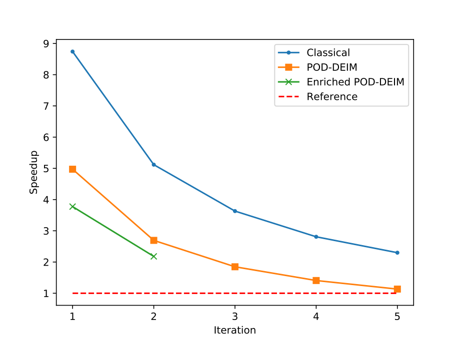




First test case (pseudo-2D): speedup

$$\text{speedup} = \frac{\tau_{\text{ref}}}{\tau_{\text{parareal}}}$$



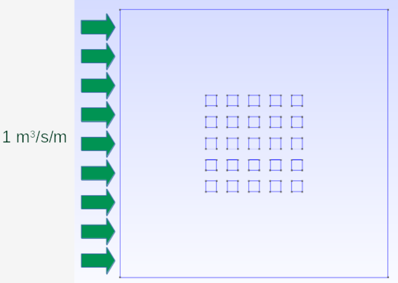


Second test case: a fictional flood simulation

- $\Omega = [0, 100]^2$;
- Initial solution:

$$\begin{cases} h(t=0, x, y) = 0.1 \\ u_x(t=0, x, y) = 0 \qquad (x, y) \in \Omega \\ u_y(t=0, x, y) = 0 \end{cases}$$

- Inward unitary flux on the western boundary.

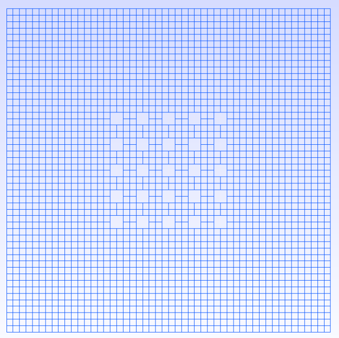


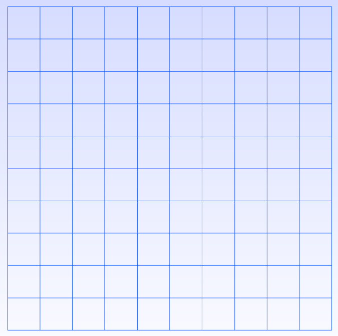
Second test case: a fictional flood simulation

- Propagators:

$\mathcal{F}_{\delta t}$	SWE	$\delta t = 0.001$	$\delta x = 2$	$\delta y = 2$
$\mathcal{G}_{\Delta t}$	por-SWE	$\Delta t = 1$	$\Delta x = 10$	$\Delta y = 10$

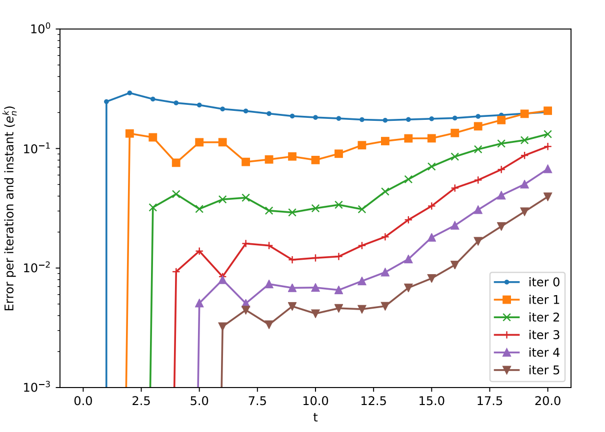
- $T = 20,\ N_{\Delta t} = 20,\ P = 20$ processors
- ROM-based parareal with 4 extra snapshots per Δt ($\alpha = 1/5$);



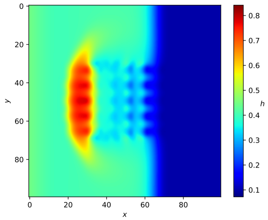


Second test case: a fictional flood simulation

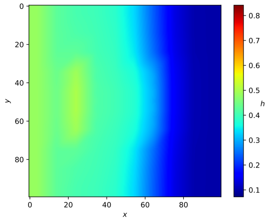
Error per instant and iteration



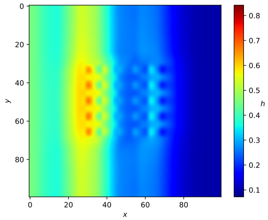
Test case: water depth at $t = 20$



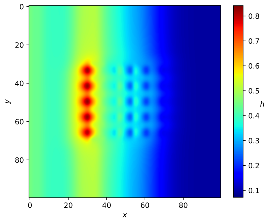
(a) Ref. (SWE)



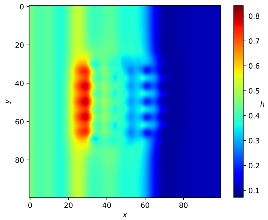
(b) Iter. 0 (por-SWE)



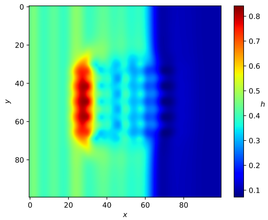
(c) Iteration 1



(d) Iteration 2

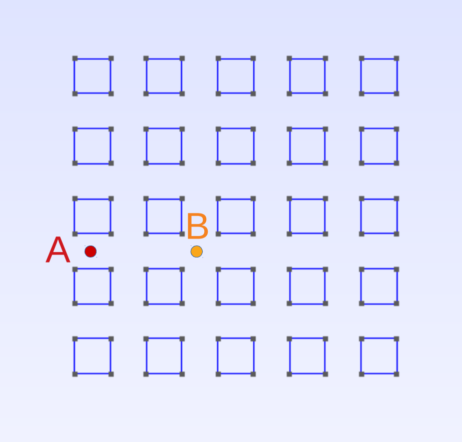


(e) Iteration 3

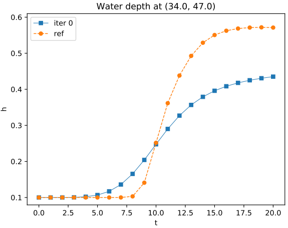


(f) Iteration 4

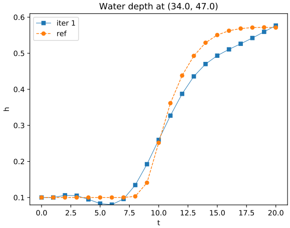
Test case: probes locations



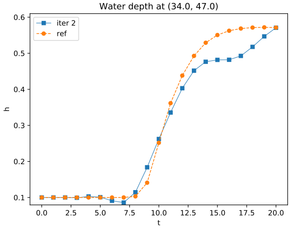
Second test case: water depth at $A(34, 47)$



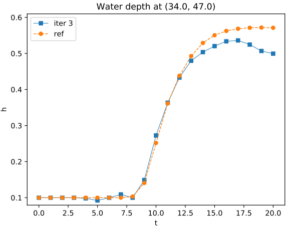
(a) Iteration 0



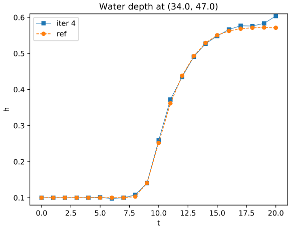
(b) Iteration 1



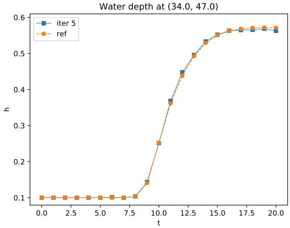
(c) Iteration 2



(d) Iteration 3

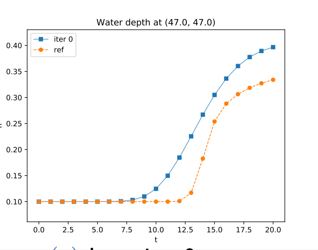


(e) Iteration 4

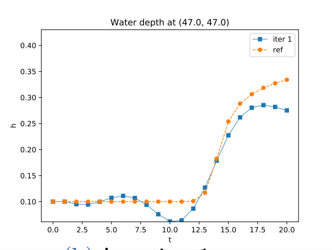


(f) Iteration 5

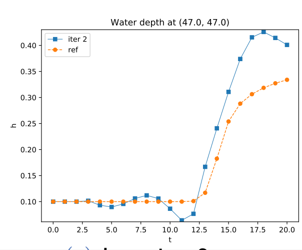
Test case: water depth at $B(47, 47)$



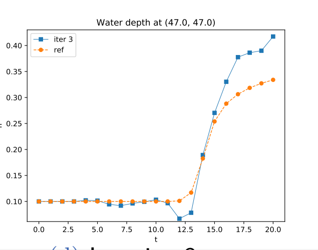
(a) Iteration 0



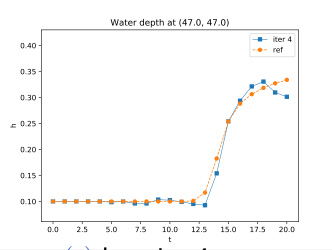
(b) Iteration 1



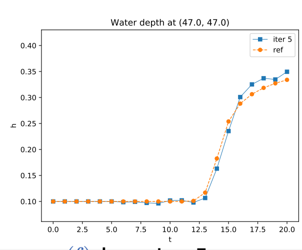
(c) Iteration 2



(d) Iteration 3



(e) Iteration 4

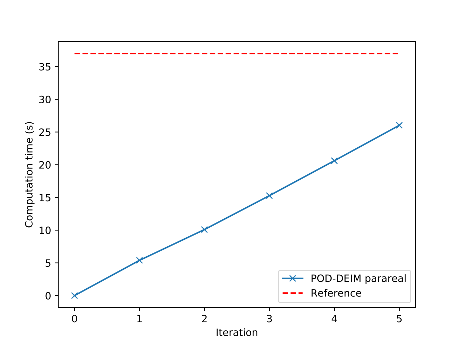


(f) Iteration 5

Test case: speedup

$$\text{speedup} = \frac{\tau_{\text{ref}}}{\tau_{\text{parareal}}}$$

- Coarse model (iteration 0) $\approx$ 0.11 s (speedup ≈ 3500)



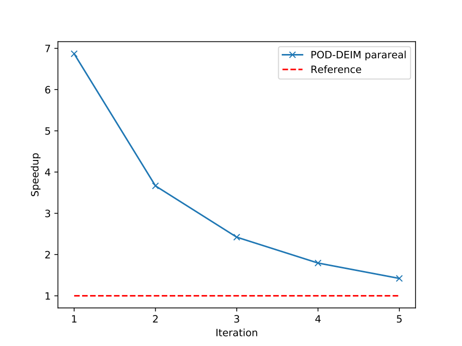


Table of Contents

Adaptative parareal

[Maday and Mula (2020)]

- Modification of the classical parareal method

Adaptative parareal

[Maday and Mula (2020)]

- Modification of the classical parareal method
  - The parallel efficiency is limited by the fine propagator;

Adaptative parareal

[Maday and Mula (2020)]

- Modification of the classical parareal method
  - The parallel efficiency is limited by the fine propagator;
  - Idea: use fine propagators with an increasing accuracy ζ along iterations;

Adaptative parareal

[Maday and Mula (2020)]

- Modification of the classical parareal method
  - The parallel efficiency is limited by the fine propagator;
  - Idea: use fine propagators with an increasing accuracy ζ along iterations;

$$\boldsymbol{y}_{n+1}^{k+1} = \mathcal{G}_{\Delta t}(\boldsymbol{y}_n^{k+1}) + \mathcal{F}_{\delta t}(\boldsymbol{y}_n^k) - \mathcal{G}_{\Delta t}(\boldsymbol{y}_n^k)$$

Adaptative parareal

[Maday and Mula (2020)]

- Modification of the classical parareal method
  - The parallel efficiency is limited by the fine propagator;
  - Idea: use fine propagators with an increasing accuracy ζ along iterations;

$$\boldsymbol{y}_{n+1}^{k+1} = \mathcal{G}_{\Delta t}(\boldsymbol{y}_n^{k+1}) + [\mathcal{E}(\boldsymbol{y}_n^k), \zeta^k] - \mathcal{G}_{\Delta t}(\boldsymbol{y}_n^k)$$

Adaptative parareal

[Maday and Mula (2020)]

- Modification of the classical parareal method
  - The parallel efficiency is limited by the fine propagator;
  - Idea: use fine propagators with an increasing accuracy ζ along iterations;

$$\boldsymbol{y}_{n+1}^{k+1} = \mathcal{G}_{\Delta t}(\boldsymbol{y}_n^{k+1}) + [\mathcal{E}(\boldsymbol{y}_n^k), \zeta^k] - \mathcal{G}_{\Delta t}(\boldsymbol{y}_n^k)$$

  - Ideally: same precision is attained.

Adaptative ROM-based parareal

- Define $\widehat{\mathcal{F}}^1_{\delta t}, \widehat{\mathcal{F}}^2_{\delta t}, \ldots, \widehat{\mathcal{F}}^{\overline{n}}_{\delta t} = \mathcal{F}_{\delta t}$ with increasing accuracy.

Adaptative ROM-based parareal

- Define $\widehat{\mathcal{F}}^1_{\delta t}, \widehat{\mathcal{F}}^2_{\delta t}, \ldots, \widehat{\mathcal{F}}^{\overline{n}}_{\delta t} = \mathcal{F}_{\delta t}$ with increasing accuracy.
- ROM-based parareal iteration:

$$\boldsymbol{y}^{k+1}_{n+1} = \mathcal{F}^{k}_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^{k+1}_n) + \mathcal{F}_{\delta t}(\boldsymbol{y}^k_n) - \mathcal{F}^{k}_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^k_n)$$

Adaptative ROM-based parareal

- Define $\widehat{\mathcal{F}}^{1}_{\delta t}, \widehat{\mathcal{F}}^{2}_{\delta t}, \dots, \widehat{\mathcal{F}}^{\overline{n}}_{\delta t} = \mathcal{F}_{\delta t}$ with increasing accuracy.
- ROM-based parareal iteration:

$$\boldsymbol{y}^{k+1}_{n+1} = \widehat{\mathcal{F}}^{k}_{\delta t,r}(\boldsymbol{P}^{k}\boldsymbol{y}^{k+1}_{n}) + \widehat{\mathcal{F}}^{k}_{\delta t}(\boldsymbol{y}^{k}_{n}) - \widehat{\mathcal{F}}^{k}_{\delta t,r}(\boldsymbol{P}^{k}\boldsymbol{y}^{k}_{n})$$

Adaptative ROM-based parareal

- Define $\widehat{\mathcal{F}}^1_{\delta t}, \widehat{\mathcal{F}}^2_{\delta t}, \dots, \widehat{\mathcal{F}}^{\overline{n}}_{\delta t} = \mathcal{F}_{\delta t}$ with increasing accuracy.
- ROM-based parareal iteration:

$$\boldsymbol{y}^{k+1}_{n+1} = \widehat{\mathcal{F}}^k_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^{k+1}_n) + \widehat{\mathcal{F}}^k_{\delta t}(\boldsymbol{y}^k_n) - \widehat{\mathcal{F}}^k_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^k_n)$$

- Reduced cost of the fine propagator;

Adaptative ROM-based parareal

- Define $\widehat{\mathcal{F}}_{\delta t}^{1}, \widehat{\mathcal{F}}_{\delta t}^{2}, \ldots, \widehat{\mathcal{F}}_{\delta t}^{\overline{n}} = \mathcal{F}_{\delta t}$ with increasing accuracy.
- ROM-based parareal iteration:

$$\boldsymbol{y}_{n+1}^{k+1} = \widehat{\mathcal{F}}_{\delta t,r}^{k}(\boldsymbol{P}^{k}\boldsymbol{y}_{n}^{k+1}) + \widehat{\mathcal{F}}_{\delta t}^{k}(\boldsymbol{y}_{n}^{k}) - \widehat{\mathcal{F}}_{\delta t,r}^{k}(\boldsymbol{P}^{k}\boldsymbol{y}_{n}^{k})$$

- Reduced cost of the fine propagator;
- Reduced cost for formulating the ROMs;

Adaptative ROM-based parareal

- Define $\widehat{\mathcal{F}}_{\delta t}^{1}, \widehat{\mathcal{F}}_{\delta t}^{2}, \ldots, \widehat{\mathcal{F}}_{\delta t}^{\overline{n}} = \mathcal{F}_{\delta t}$ with increasing accuracy.
- ROM-based parareal iteration:

$$\boldsymbol{y}_{n+1}^{k+1} = \widehat{\mathcal{F}}_{\delta t,r}^{k}(\boldsymbol{P}^{k}\boldsymbol{y}_{n}^{k+1}) + \widehat{\mathcal{F}}_{\delta t}^{k}(\boldsymbol{y}_{n}^{k}) - \widehat{\mathcal{F}}_{\delta t,r}^{k}(\boldsymbol{P}^{k}\boldsymbol{y}_{n}^{k})$$

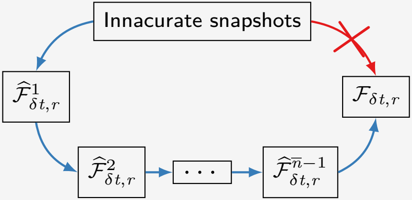
- Reduced cost of the fine propagator;
- Reduced cost for formulating the ROMs;
- Reduced cost for solving the ROMs;

Adaptative ROM-based parareal

- Define $\widehat{\mathcal{F}}^1_{\delta t}, \widehat{\mathcal{F}}^2_{\delta t}, \dots, \widehat{\mathcal{F}}^{\overline{n}}_{\delta t} = \mathcal{F}_{\delta t}$ with increasing accuracy.
- ROM-based parareal iteration:

$$\boldsymbol{y}^{k+1}_{n+1} = \widehat{\mathcal{F}}^k_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^{k+1}_n) + \widehat{\mathcal{F}}^k_{\delta t}(\boldsymbol{y}^k_n) - \widehat{\mathcal{F}}^k_{\delta t,r}(\boldsymbol{P}^k \boldsymbol{y}^k_n)$$

- Reduced cost of the fine propagator;
- Reduced cost for formulating the ROMs;
- Reduced cost for solving the ROMs;
- More stable ROMs.



Second test case with the adaptative ROM-parareal

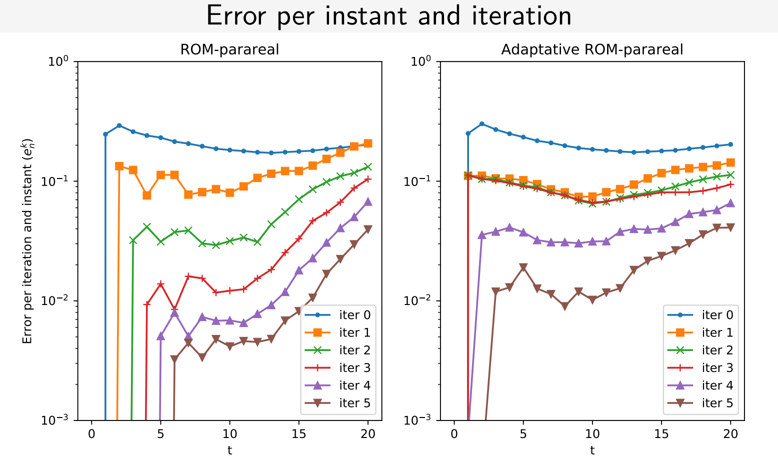
- **Coarse model (iteration 0):** $\Delta t = 1$; $\Delta x = \Delta y = 10$

Second test case with the adaptative ROM-parareal

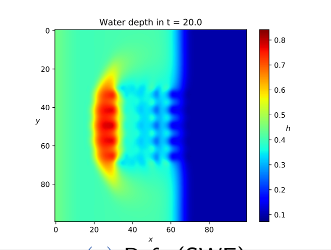
- **Coarse model (iteration 0):** $\Delta t = 1$; $\Delta x = \Delta y = 10$
- **Fine models:**

Iteration	**Fine model**	δt	$\delta x = \delta y$
1	$\widehat{\mathcal{F}}^1_{\delta t}$	0.05	4
2	$\widehat{\mathcal{F}}^2_{\delta t}$	0.01	4
3	$\widehat{\mathcal{F}}^3_{\delta t}$	0.005	4
4	$\widehat{\mathcal{F}}^4_{\delta t}$	0.005	2
5	$\widehat{\mathcal{F}}^5_{\delta t} = \mathcal{F}_{\delta t}$	0.001	2

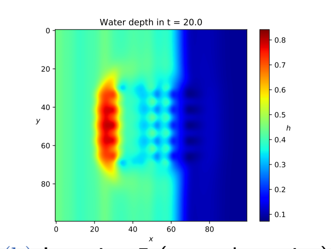
Adaptative test case: errors



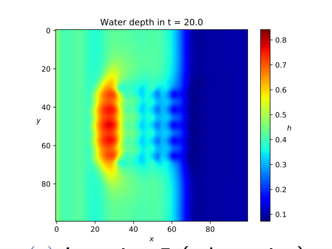
Adaptative test case: water depth at $t = 20$



(a) Ref. (SWE)

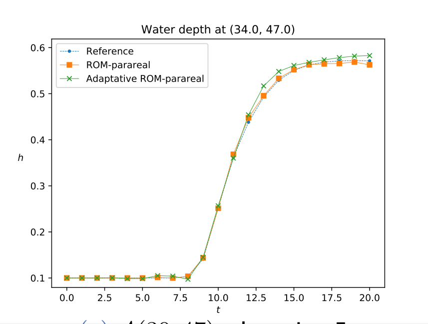


(b) Iteration 5 (non-adaptative)

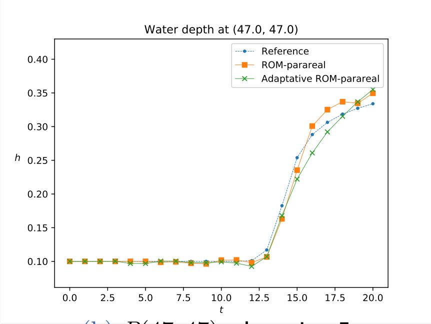


(c) Iteration 5 (adaptative)

Adaptative test case: water depth at probes

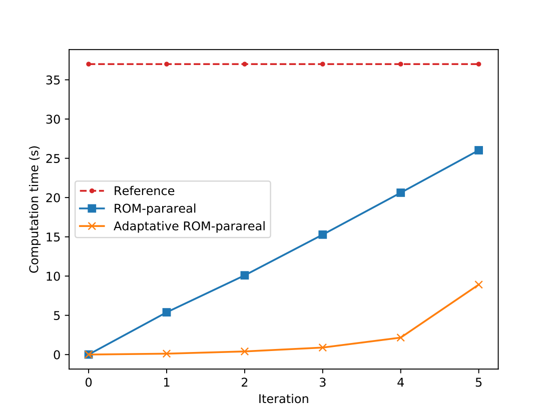


(a) $A(30, 47)$ - Iteration 5



(b) $B(47, 47)$ - Iteration 5

Test case: speedup



- **Speedup (iteration 5):**
  - Non-adaptative: 1.4
  - Adaptative: 4.2

Table of Contents

Conclusions

- ROM-based parareal methods are efficient for speeding up nonlinear hyperbolic problems;

Conclusions

- ROM-based parareal methods are efficient for speeding up nonlinear hyperbolic problems;
- Crucial aspect: formulation of the ROM;

Conclusions

- ROM-based parareal methods are efficient for speeding up nonlinear hyperbolic problems;
- Crucial aspect: formulation of the ROM;
- Enrichment of the snapshots sets:
  - No extra cost for computing the extra snapshots;
  - More stability and faster convergence;
  - Quadratic dependency of the POD on the number of snapshots

Conclusions

- ROM-based parareal methods are efficient for speeding up nonlinear hyperbolic problems;
- Crucial aspect: formulation of the ROM;
- Enrichment of the snapshots sets:
  - No extra cost for computing the extra snapshots;
  - More stability and faster convergence;
  - Quadratic dependency of the POD on the number of snapshots
- Adaptative approach
  - Progressive refinement of $\mathcal{F}_{\delta t}$;
  - Improved speedup and stability;

Perspectives

- Adaptative approach: study configurations, propose objective criteria for defining the fine models;
- Application to more realistic test cases:
  - Long simulations;
  - Solutions with strong variation;
  - Interpolation between spatial meshes

Thank you for your attention

Barrault, M., Maday, Y., Nguyen, N. C., and Patera, A. T. (2004). An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667 – 672.

Chaturantabut, S. and Sorensen, D. C. (2010). Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764.

Chen, F., Hesthaven, J. S., and Zhu, X. (2014). *On the Use of Reduced Basis Methods to Accelerate and Stabilize the Parareal Method*, pages 187–214. Springer International Publishing, Cham.

Defina, A. (2000). Two-dimensional shallow flow equations for partially dry areas. *Water Resources Research*, 36(11):3251–3264.

Guinot, V., Sanders, B. F., and Schubert, J. E. (2017). Dual integral porosity shallow water model for urban flood modelling. *Advances in Water Resources*, 103:16 – 31.

Guinot, V. and Soares-Frazão, S. (2006). Flux and source term discretization in two-dimensional shallow water models with porosity on unstructured grids. *International Journal for Numerical Methods in Fluids*, 50(3):309–345.

Lions, J.-L., Maday, Y., and Turinici, G. (2001). Résolution d’edp par un schéma en temps 'pararéel'. *Comptes Rendus de l’Académie des Sciences - Series I - Mathematics*, 332(7):661 – 668.

Maday, Y. and Mula, O. (2020). An Adaptive Parareal Algorithm. *Journal of Computational and Applied Mathematics*.

Ruprecht, D. (2018). Wave propagation characteristics of Parareal. *Computing and Visualization in Science*, 19:1–17.