# Towards real-time physics simulation

G. Gbikpi-Benissan, F. Magoulès
CentraleSupélec – Laboratoire MICS

## Para-real-time numerical integration

- **Parallel time integration** schemes for solving evolution partial differential equations (climate, financial options, population growth, plasma dynamics, ...)
- **Predictions-corrections** on time windows $[T_{i-1}, T_i]$, combining coarse approximations $\tilde{u}_{i,\Delta t}$ and fine or exact solutions $u_i$

- **Goals**

→ Assess the ability to (almost) **simultaneously solve all time steps** of PDEs using computer clusters

→ Develop a new framework to take into account **frequent perturbations** in large computer networks

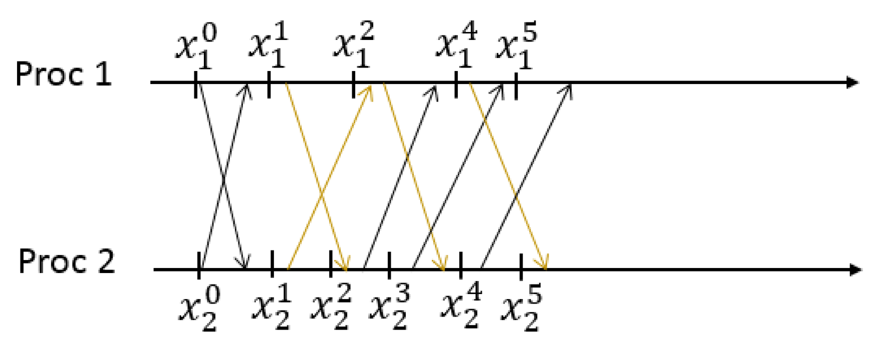
## Classical scheme [5, 4]

- **Time decomposition** ($i \in \{1, \dots, n\}$)
$\partial u_i/\partial t = f(u_i), \quad t \in [T_{i-1}, T_i], \quad T_i = i\Delta t$
$u_i(T_{i-1}) = \lambda_{i-1}, \quad \lambda_0 = u(0)$
- **Iterations** (initialization: $\lambda_i^0 = \tilde{u}_{\Delta t}(T_i)$)
$\lambda_i^{k+1} = \tilde{u}_{i,\Delta t}^{k+1}(T_i) + u_i^k(T_i) - \tilde{u}_{i,\Delta t}^k(T_i)$

→ **Transmission conditions:** $\lambda_{i-1}^{k+1}, \ \lambda_{i-1}^k$

## Asynchronous pattern [2]

General example ($x = \mathcal{T}(x)$)




$$
\begin{aligned}
x_1^1 &:= \mathcal{T}_1(x_1^0, x_2^0) & x_2^1 &:= \mathcal{T}_2(x_1^0, x_2^0)\\
x_1^2 &:= \mathcal{T}_1(x_1^1, x_2^0) & x_2^2 &:= \mathcal{T}_2(x_1^0, x_2^1)\\
x_1^3 &:= x_1^2 & x_2^3 &:= \mathcal{T}_2(x_1^1, x_2^2)\\
x_1^4 &:= \mathcal{T}_1(x_1^3, x_2^2) & x_2^4 &:= \mathcal{T}_2(x_1^2, x_2^3)\\
x_1^5 &:= \mathcal{T}_1(x_1^4, x_2^3) & x_2^5 &:= \mathcal{T}_2(x_1^2, x_2^4)
\end{aligned}
$$

$$
x_i^{k+1} = \begin{cases} \mathcal{T}_i(x_1^{\psi_1^i(k)}, \dots, x_n^{\psi_n^i(k)}) & \forall i \in P_k \\ x_i^k & \forall i \in \{1, \dots, n\} \setminus P_k \end{cases}
$$

→ $P_k$: arbitrary set of components updated at instant $k$ (**Ostrowski's free steering**)

→ $\psi_i^j(k) \le k$: version of component $j$ used for updating component $i$ at instant $k$

## Computational efficiency

- Aubanel's implementation pattern [1]

proc 1 $G\mathring{F}\mathring{F}\mathring{F}\mathring{F}$

proc 2 $GG\mathring{F}\mathring{F}\mathring{F}$ $\overset{C}{\mathring{F}G}$ $\mathring{F}$ $\mathring{F}$ $\mathring{F}$ $\mathring{F}$

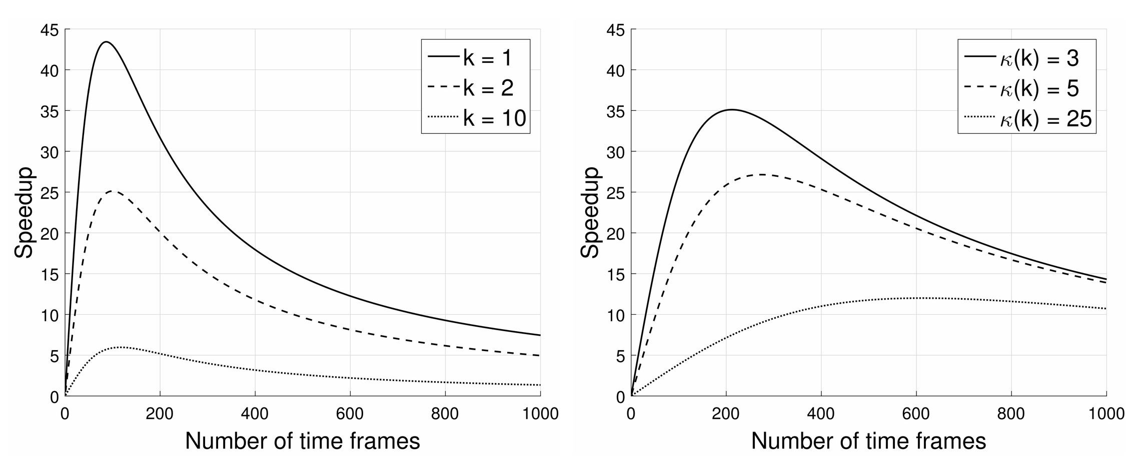
proc 3 $GGG\mathring{F}\mathring{F}$ $\mathring{F}\mathring{F}$ $\overset{C}{-}$ $G$ $\mathring{F}$ $\mathring{F}$ $\overset{C}{\mathring{F}\mathring{F}G}$ $\mathring{F}\mathring{F}\mathring{F}\mathring{F}$

proc 4 $GGGG\mathring{F}$ $\mathring{F}\mathring{F}$ $\mathring{F}$ $-$ $\overset{C}{-}$ $G$ $\mathring{F}\mathring{F}\mathring{F}$ $\overset{C}{\mathring{F}G\mathring{F}\mathring{F}}$

$G$: one-step coarse integration
$\mathring{F}$: one-step fine integration ($F = \mathring{F} \circ \mathring{F} \circ \mathring{F} \circ \mathring{F}$)
$C$: communication

→ **Delayed communications might not be overlapped despite the initial shift**

- Speedup bounded by $n/2$ (cf. [3])




## Asynchronous time integration scheme [6, 7]

- **Explicit iterations** ($i \in P_k$)
$\lambda_i^{k+1} = \tilde{u}_{i,\Delta t}^{\psi_{1,i}(k)}(T_i) + u_i^{\psi_{2,i}(k)}(T_i) - \tilde{u}_{i,\Delta t}^{\psi_{2,i}(k)}(T_i)$
- **Implicit iterations** ($i \in \{1, \dots, n\} \setminus P_k$)
$\lambda_i^{k+1} = \lambda_i^k$
- **Fixed-point formulation**
$\lambda_i^{k+1} = \widetilde{\mathcal{T}}_i\left(\lambda^{\psi_{1,i}(k)}, \lambda^{\psi_{2,i}(k)}\right)$

→ **Transmissions perturbation:** $\psi_1, \ \psi_2$

- Unconditionally generates nested sets $S_{l+1} \subset S_{l \in \mathbb{N}}$ with at most $S_{l \ge n} = \{\lambda^\infty\}$

→ Termination in finite number of steps (as for the classical scheme)

- **Error at each iteration:** $\|\lambda^\infty - \lambda^k\|_\infty \le \left(\|\tilde{u}_{\Delta t}\|_{\Delta t}^{\sup} + \|u - \tilde{u}_{\Delta t}\|_{\Delta t}^{\sup}\right)^{\sigma(k)} \|\lambda^\infty - \lambda^0\|_\infty$
with $\lim\limits_{k \to +\infty} \sigma(k) = +\infty$

→ The asynchronous scheme is convergent if $\|\tilde{u}_{\Delta t}\|_{\Delta t}^{\sup} + \|u - \tilde{u}_{\Delta t}\|_{\Delta t}^{\sup} < 1$
**which is the same condition for the classical scheme on unbounded time intervals**

- The computational cost **fits the ideal Aubanel's model [1]**
$\mathcal{C}(\kappa(k_{\text{async}}), n) = n\,\mathcal{C}_{\tilde{u}_{\Delta t}}^{\Delta t} + \kappa(k_{\text{async}})\left(\mathcal{C}_u^{\Delta t} + \mathcal{C}_{\tilde{u}_{\Delta t}}^{\Delta t}\right), \qquad \kappa(k) := \max\limits_{i \in \{1,\dots,n\}} |\{l \le k \, : \, i \in P_l\}|$

→ Speedup over the classical scheme
$$\frac{\mathcal{C}(k, n, \bar{\mathcal{C}}_{\text{com}})}{\mathcal{C}(\kappa(k_{\text{async}}), n)} \le 1 + \frac{(n-2)\,\bar{\mathcal{C}}_{\text{com}}}{\mathcal{C}_u^{\Delta t} + \mathcal{C}_{\tilde{u}_{\Delta t}}^{\Delta t}}$$

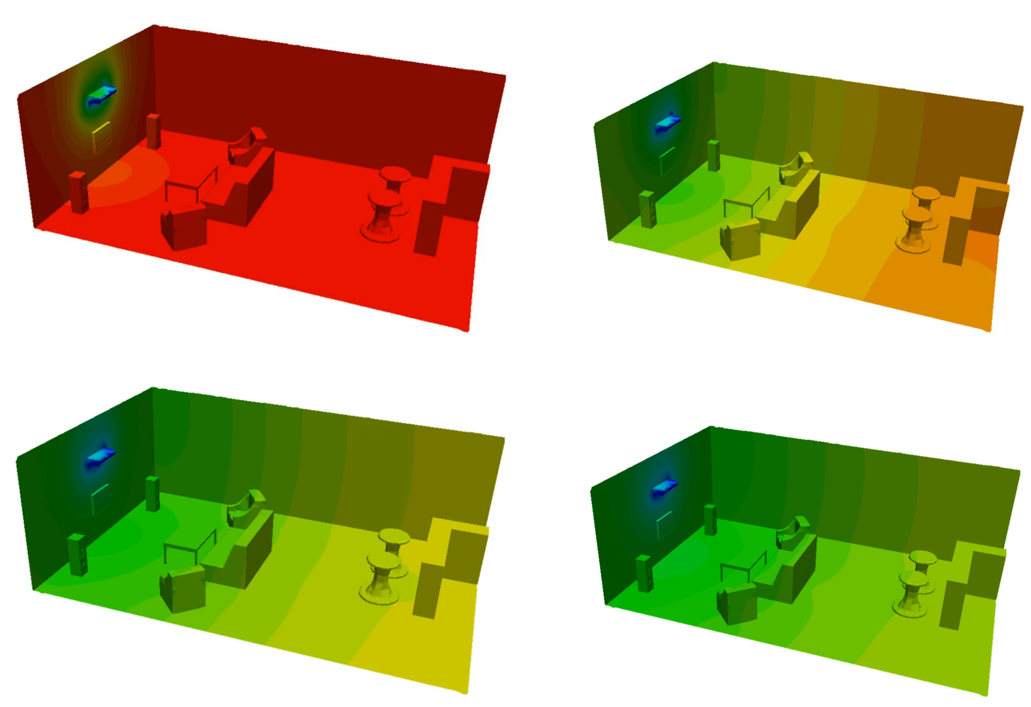
→ **Speedup on unbounded time intervals**
$$\lim_{n \to +\infty} \frac{\mathcal{C}(k, n, \bar{\mathcal{C}}_{\text{com}})}{\mathcal{C}(\kappa(k_{\text{async}}), n)} = \mathbf{1} + \mathbf{k}\frac{\bar{\mathcal{C}}_{\text{com}}}{\mathcal{C}_{\tilde{\mathbf{u}}_{\Delta \mathbf{t}}}^{\Delta \mathbf{t}}} \ge \mathbf{1}$$

with a **generalization of the Aubanel's model** by considering network perturbations
$$\mathcal{C}(k, n, \bar{\mathcal{C}}_{\text{com}}) = n\,\mathcal{C}_{\tilde{u}_{\Delta t}}^{\Delta t} + k\left(\mathcal{C}_u^{\Delta t} + \mathcal{C}_{\tilde{u}_{\Delta t}}^{\Delta t} + \left(n - 1 - \frac{k+1}{2}\right)\bar{\mathcal{C}}_{\text{com}}\right)$$

**Take-home message**
The asynchronous scheme is unconditionally preferable on unbounded time intervals

## Weak scaling experiment

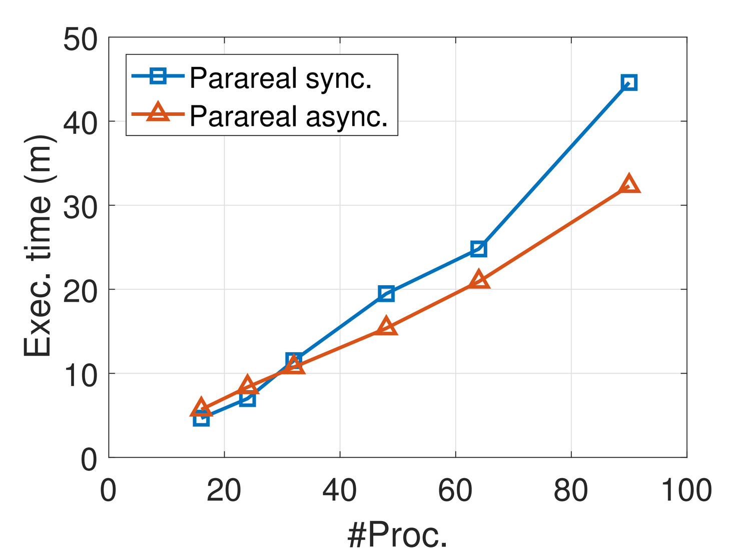
- Heat evolution: $\partial u/\partial t = \nabla^2 u$
- Fine time step: $\delta t = 0.002$
- Coarse time step: $\Delta t = 0.2$



- Sequential spatial integration
- $[0, T]$ increases proportionally to #proc

→ Expectation: constant execution time

- Fine integration: trapezoidal rule
- Coarse integration: backward Euler
- Convergence: $\|\lambda^k - \lambda^{k-1}\|_\infty < 10^{-6}$




## References

[1] E. Aubanel. Scheduling of tasks in the parareal algorithm. *Parallel Comput.*, 37(3):172–182, 2011.

[2] D. Chazan and W. Miranker. Chaotic relaxation. *Linear Algebra Appl.*, 2(2):199–222, 1969.

[3] C. Farhat and M. Chandesris. Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid–structure applications. *Int. J. Numer. Methods Eng.*, 58(9):1397–1434, 2003.

[4] M. J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):556–578, 2007.

[5] Jacques-Louis Lions, Yvon Maday, and Gabriel Turinici. Résolution d'EDP par un schéma en temps "pararéel". *C. R. Acad. Sci. Paris, Ser. I*, 332(7):661–668, 2001.

[6] F. Magoulès and G. Gbikpi-Benissan. Asynchronous parareal time discretization for partial differential equations. *SIAM J. Sci. Comput.*, 40(6):C704–C725, 2018.

[7] F. Magoulès, G. Gbikpi-Benissan, and Q. Zou. Asynchronous iterations of Parareal algorithm for option pricing models. *Mathematics*, 6(4), 2018. Paper 45.