GAIN OF EFFICIENCY
FROM NEMO 4 TO NEMO 5

Sibylle TECHENE, Gurvan MADEC, Nicolas DUCOUSSO, Florian LEMARIE, Jérôme CHANUT, Andrew COWARD, Dave STORKEY, Mike BELL, Daley CALVERT, Clément ROUSSET, Christian ETHE, Sébastien MASSON

Commodore - 09/10/2024 I NCAR

OUTLINE

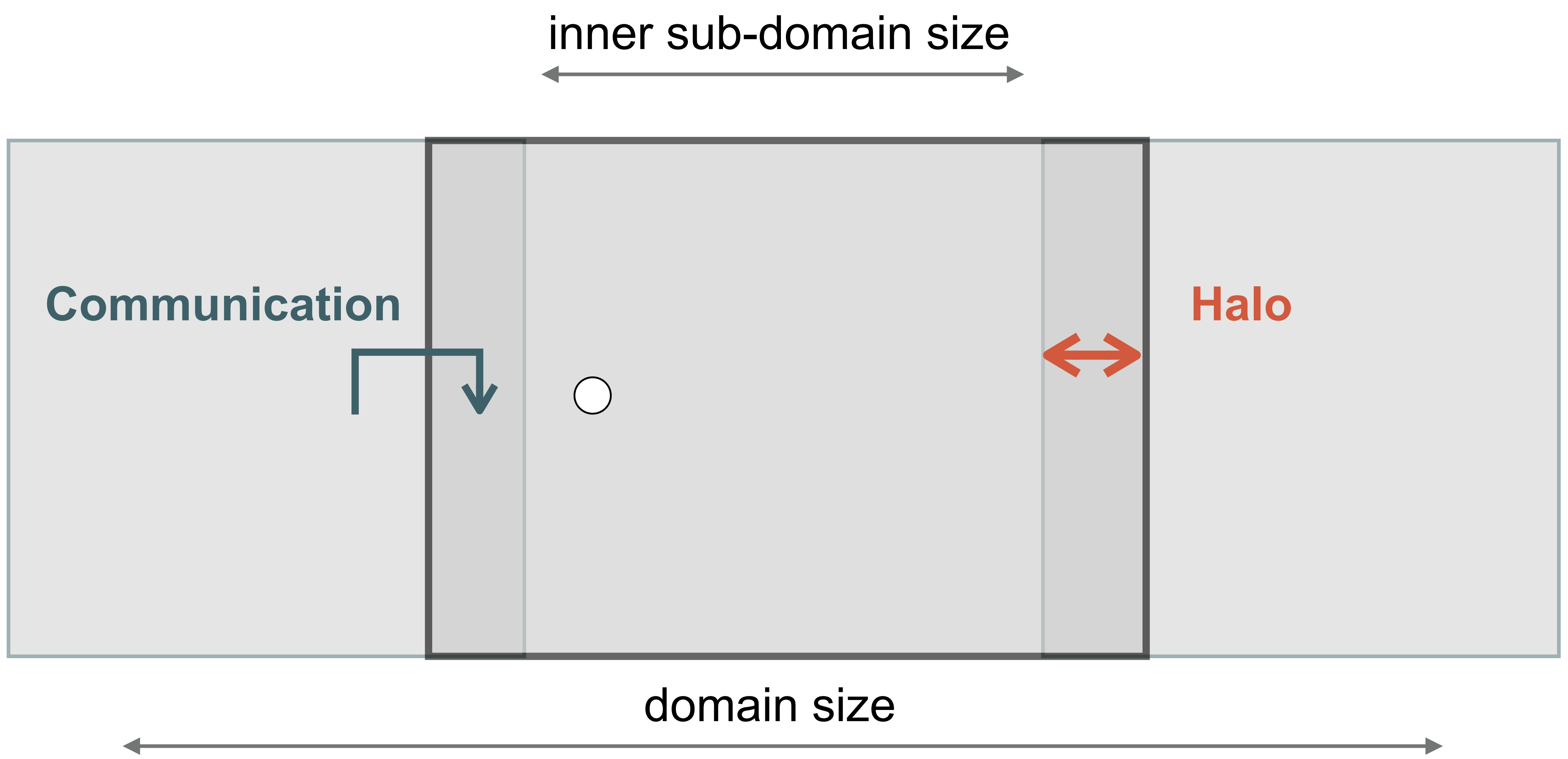
- Part 1: Optimizing the kernel accelerates NEMO run time by 20%
- Part 2: Introducing RK3 time-stepping scheme considerably accelerates NEMO
- Part 3: Changing version of I/O server of NEMO adds an extra 30%
- Part 4: Science Results

OUTLINE

- **Part 1: Kernel refactoring**
  - NEMO: basics reminder
  - Limit MPI communications
  - Reduce memory footprint
  - Performance study
- Part 2: LF to RK3 speed up
- Part 3: I/O optimization
- Part 4: Science results

NEMO: BASICS REMINDER

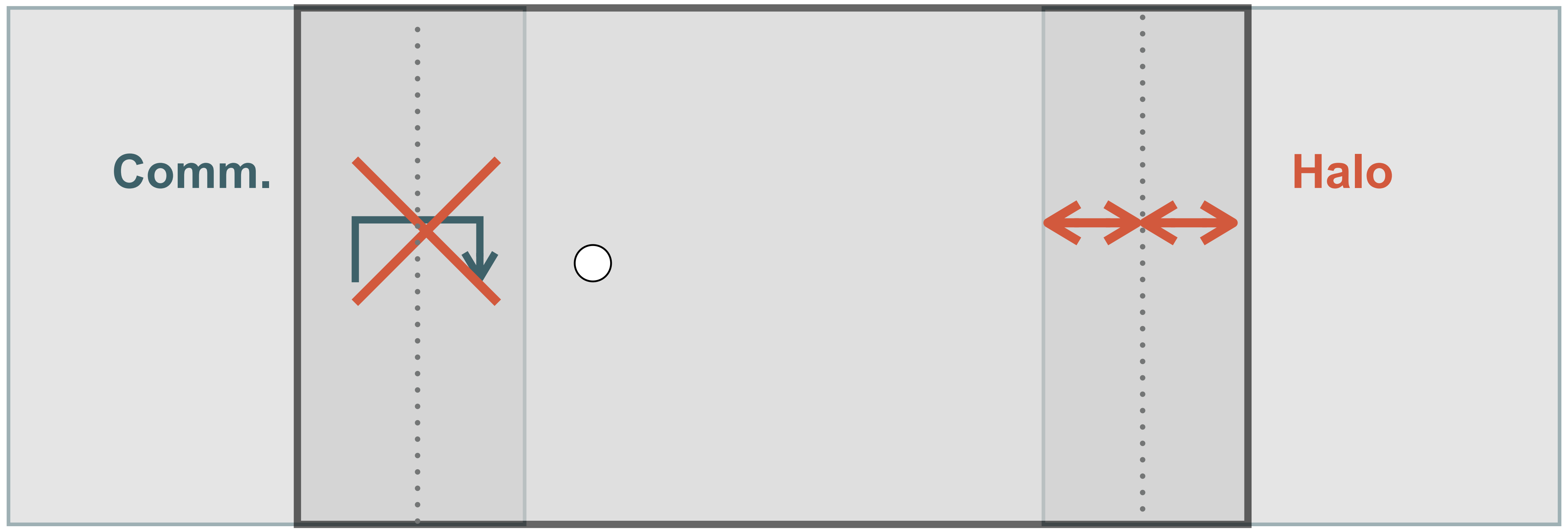
- FORTRAN 90 parallel code
  - optimized for vector machines
  - generalized vertical coordinate
  - split/explicit free surface
  - grid refinement, ice, bio
- Domain decomposition
  - Require MPI **communication** to exchange data
  - Overlapping sub-domains : **halo**



Domain is decomposed into overlapping sub-domains distributed on computing units

LIMIT MPI COMMUNICATIONS

- Too many small communications
  - move from 1 to **2 halo** size



  - **gather** communications
- Limit synchronization barrier
  - non **blocking** communications

Reduce communication cost for all components

REDUCE MEMORY FOOTPRINT

NEMO is memory bound

- ✦ Reduce global memory footprint
  - ✓ Quasi-eulerian z-coordinate

$$e_3(k) = e_3^0(k) \times (1 + \frac{\eta}{h^0} * \delta_k)$$

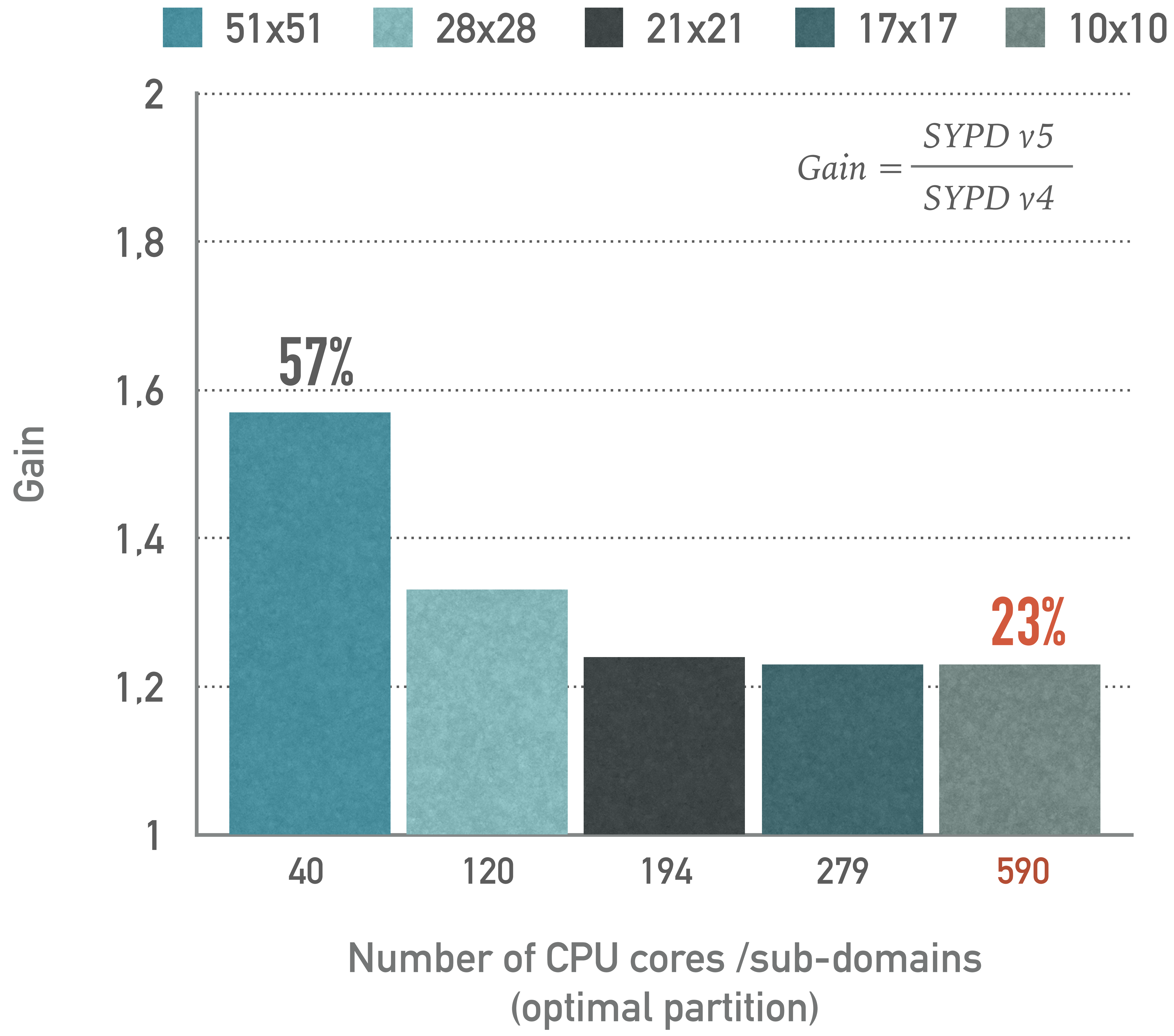
- ✦ Reduce local memory footprint
  - ✓ 3D loop with 2D slices
  - ✓ Right array size

Reduce memory footprint cost for all components

PERFORMANCES: NEMO 4 VS 5

eORCA1° OCE + ICE + I/O

- 1 year simulation daily In/Out
- Domain size : 360x331
- 75 vertical levels
- Run time dominated by comm
- Time-to-solution (10 x 10)
  - NEMO v4: 142 SYPD
  - NEMO v5: 174 SYPD



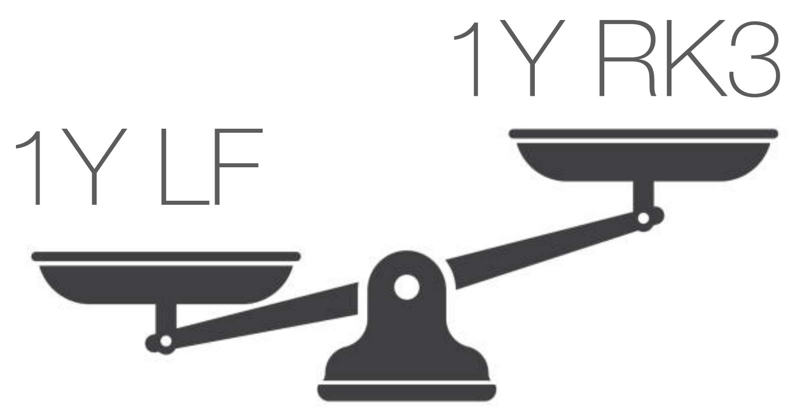
KERNEL REFACTORING

Refactoring the kernel accelerates NEMO run time by 20% at least

OUTLINE

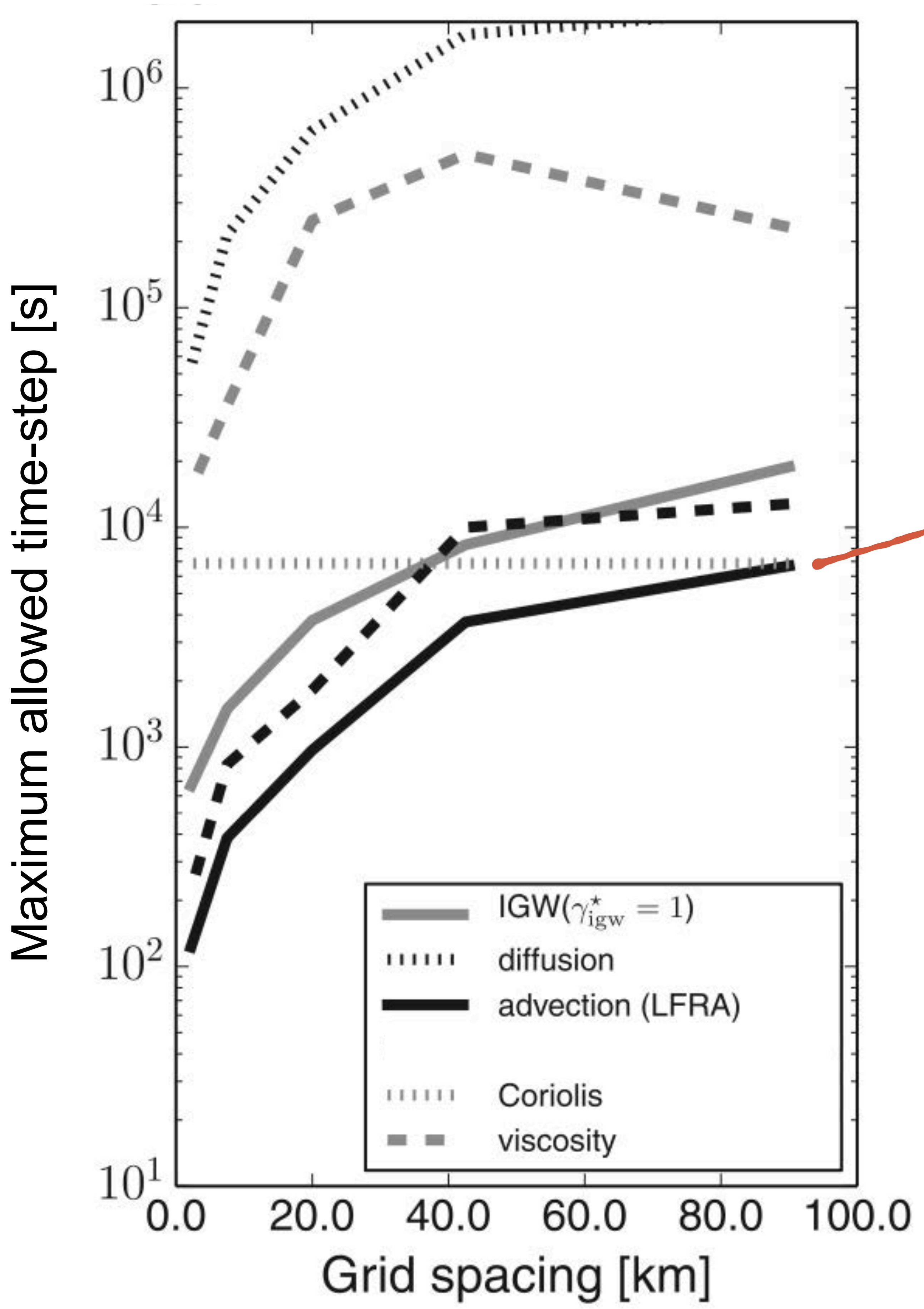
- Part 1: Kernel refactoring
- **Part 2: LF to RK3 speed up**
  - Theoretical stability constraints or RK3 strength
  - RK3 weakness
  - RK3 in NEMO
  - Performances
- Part 3: I/O optimization
- Part 4: Science results

THEORETICAL STABILITY CONSTRAINTS



3D advection is a hurdle* **

Eulerian coordinate



…that can be overtaken*

F. Lemarié et al. / Ocean Modelling 92 (2015) 124–148

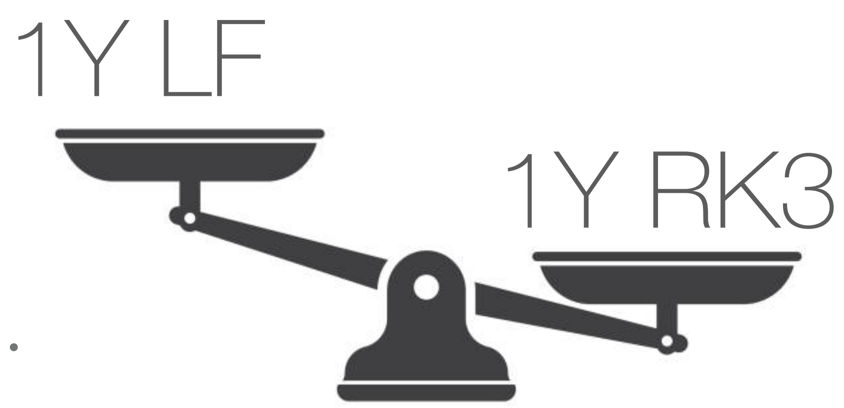
Time-scheme for advection		n_{rhs}	Stability constraints		
			$\alpha^\star_{c2}$	$\alpha^\star_{up3}$	$\alpha^\star_{Co4}$
LFRA ($\nu = 0.1$)	order 1	1	0.904	0.472	0.522
RK3	order 2,3	3	1.73	1.626	1

$$\Delta t_{RK3} \approx 2 \times \Delta t_{LF}$$

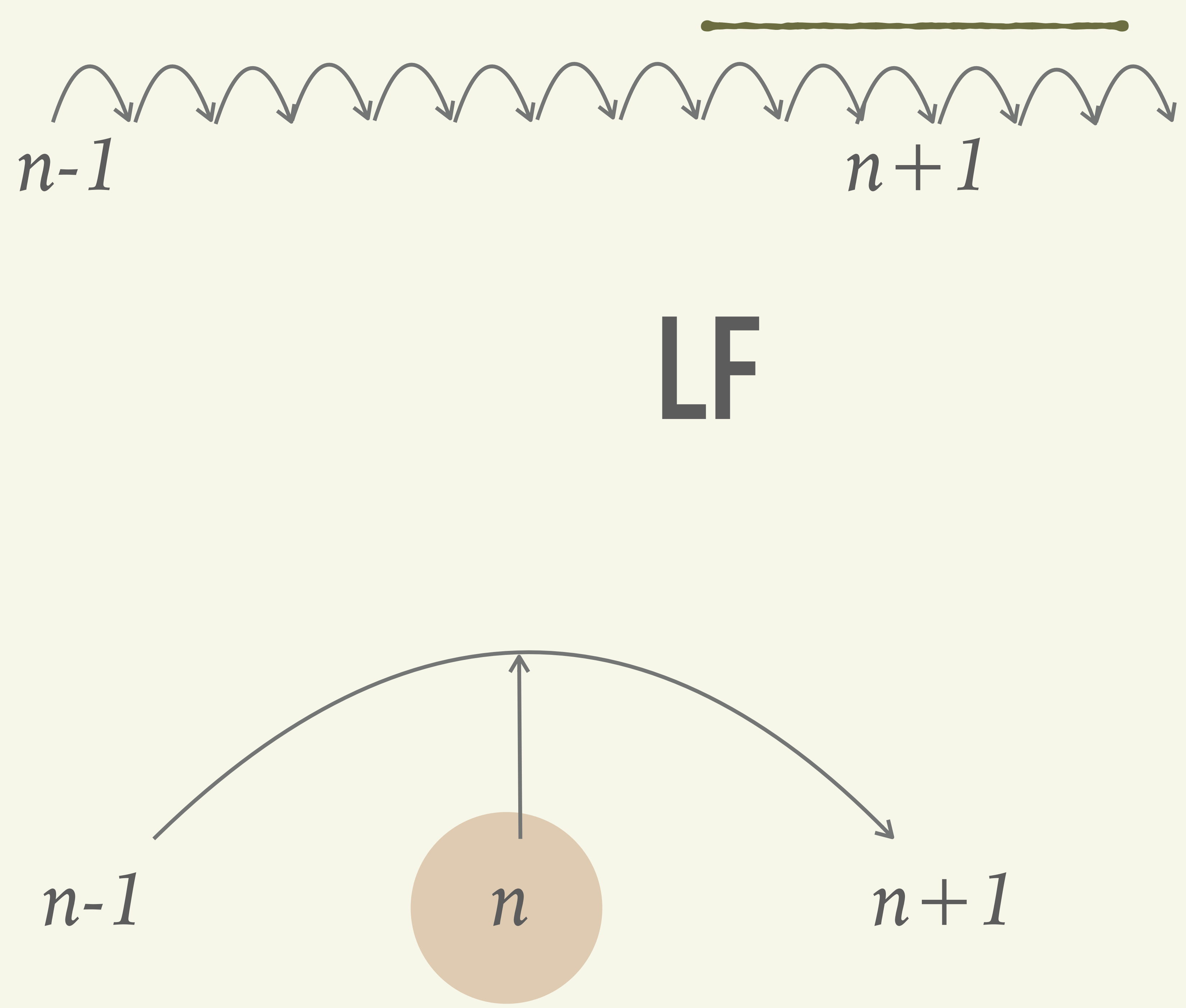
**Lemarié et al., 2015: Stability constraints for oceanic numerical models: implications for the formulation of time and space discretizations.*

***Schepetkin, 2015: An adaptive, Courant-number-dependent implicit scheme for vertical advection in oceanic modeling.*

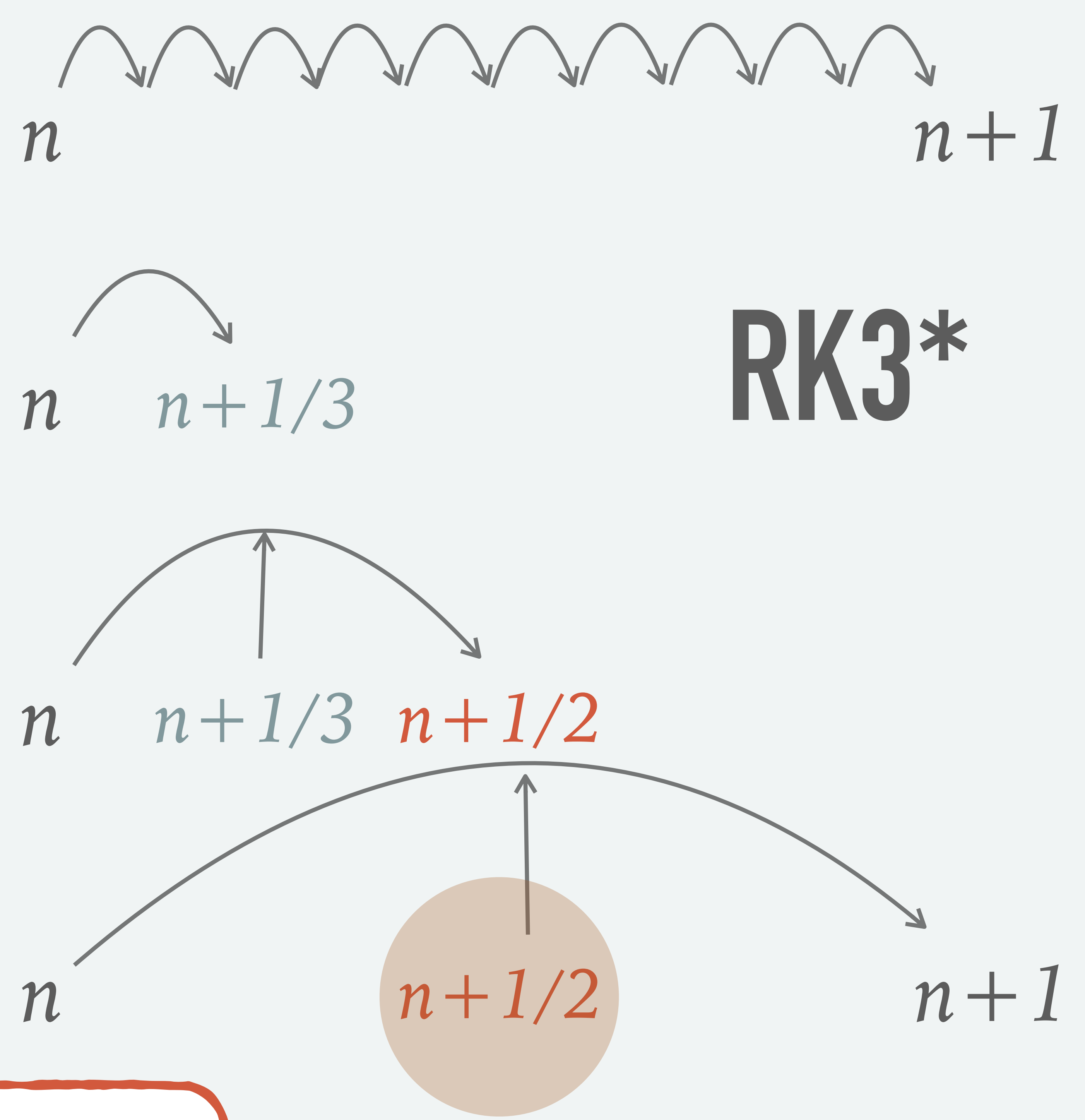
RK3 WEAKNESS: 3 TIMES MORE EXPENSIVE



LF



RK3*

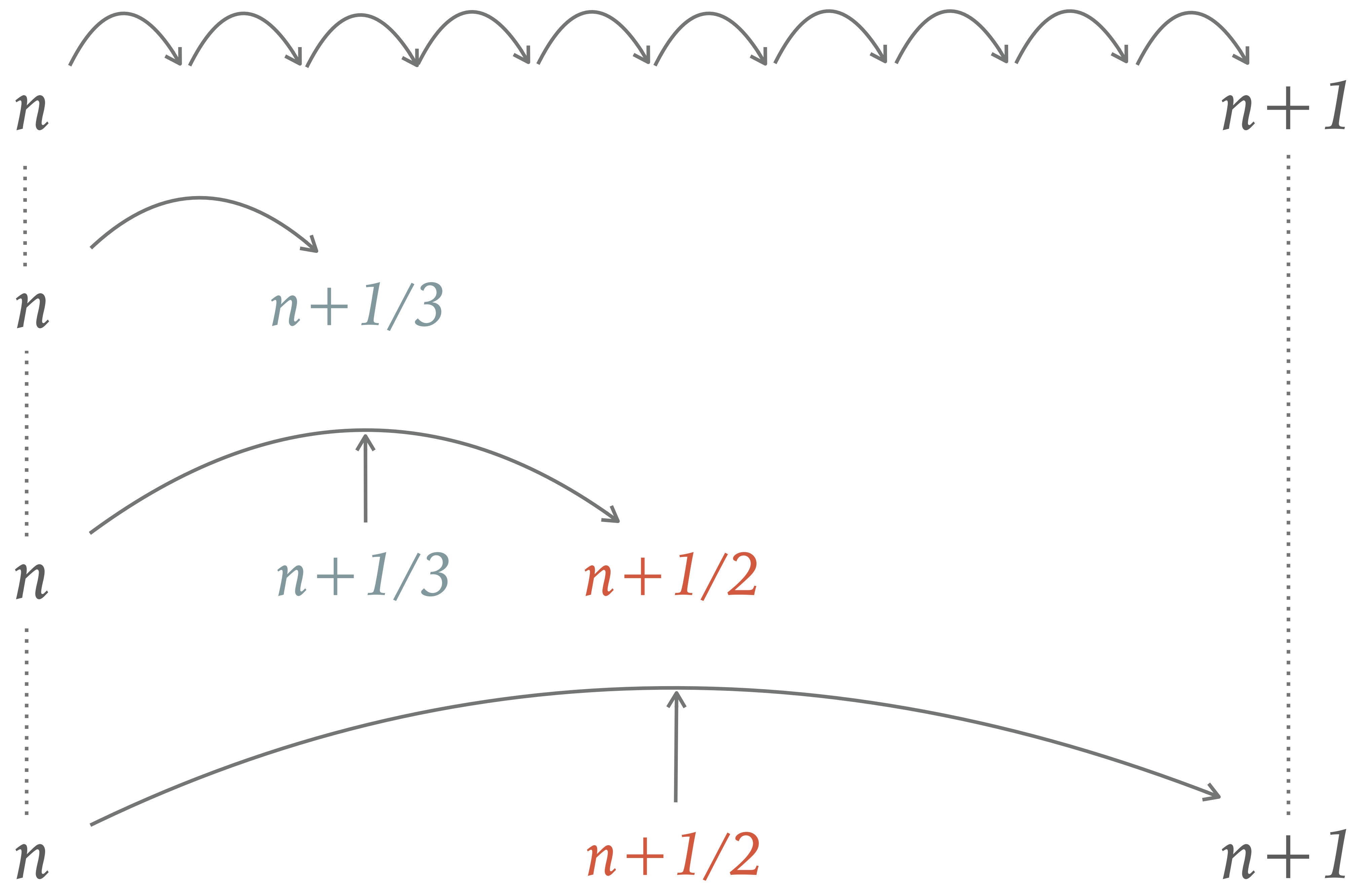


$$n_{RK3} \approx 3 \times n_{LF}$$

*Wicker, L. J., & Skamarock, W. C. (2002). Time-splitting methods for elastic models using forward time schemes. Mon. Weather Rev., 130(8), 2088–2097.

RK3 IN NEMO

Active tracers



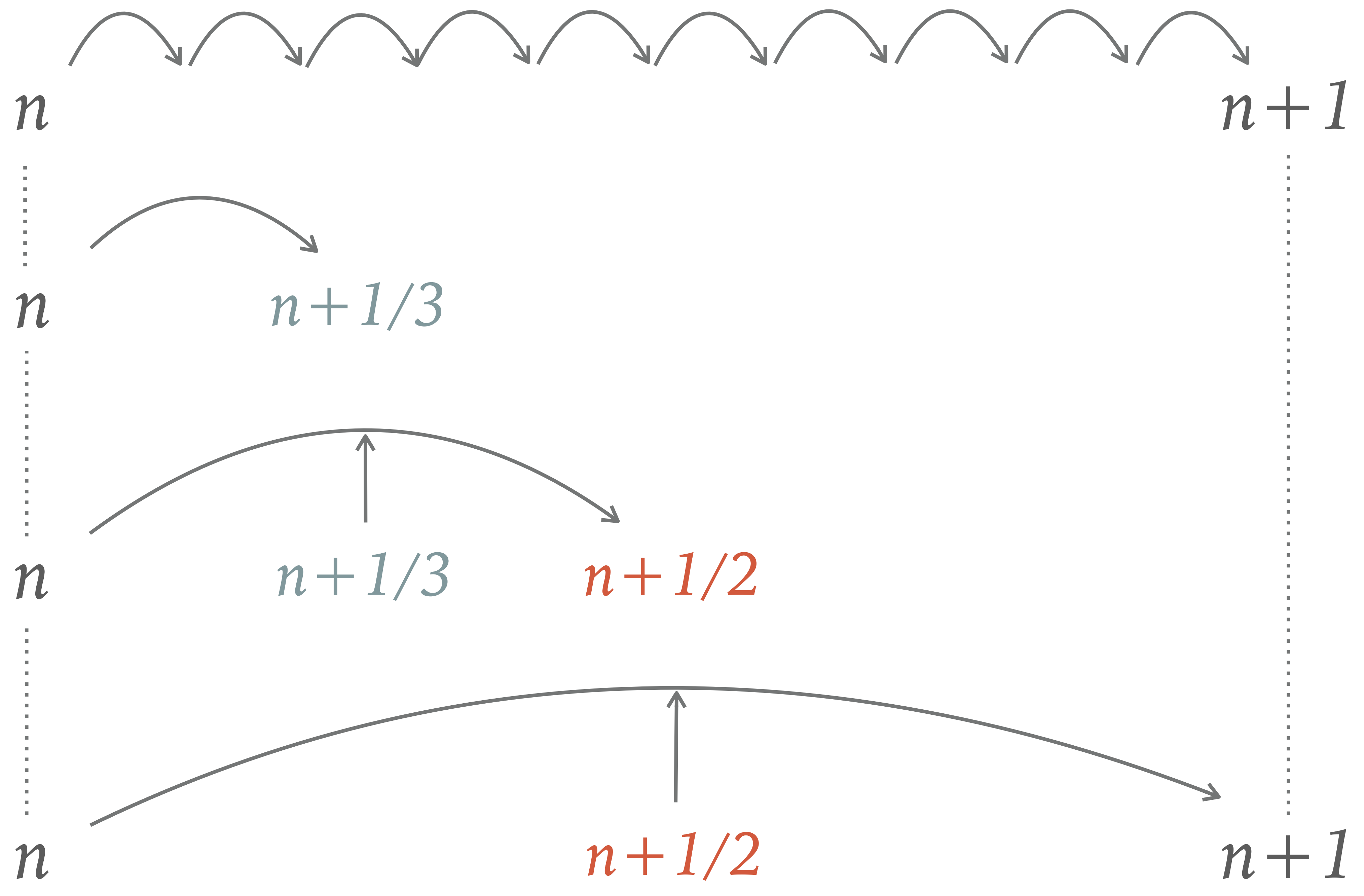
$$\mathbf{T}^{n+1/3} = \mathbf{T}^n + \frac{\Delta t}{3}\mathscr{F}_1\left(\mathbf{T}^n\right) \qquad \mathscr{F}_1\left(\mathbf{T}^n\right) = Adv* + \cancel{Ldf + Zdf} + Frc*$$

$$\mathbf{T}^{n+1/2} = \mathbf{T}^n + \frac{\Delta t}{2}\mathscr{F}_2\left(\mathbf{T}^{n+1/3}\right) \qquad \mathscr{F}_2\left(\mathbf{T}^{n+1/3}\right) = Adv* + \cancel{Ldf + Zdf} + Frc*$$

$$\mathbf{T}^{n+1} = \mathbf{T}^n + \Delta t\mathscr{F}_3\left(\mathbf{T}^{n+1/2}\right) \qquad \mathscr{F}_3\left(\mathbf{U}^{n+1/2}\right) = Adv + Ldf + Zdf + Frc$$

RK3 IN NEMO

Momentum



*Single first strategy : Ducousso et al., 2024**

$$\mathbf{U}^{n+1/3} = \mathbf{U}^{n} + \frac{\Delta t}{3}\mathscr{F}_1\left(\mathbf{U}^{n}\right)$$

$$\mathscr{F}_1\left(\mathbf{U}^{n}\right) = Adv + Cor + Hpg$$

$+$ ~~$Ldf + Zdf + Frc$~~

$$\mathbf{U}^{n+1/2} = \mathbf{U}^{n} + \frac{\Delta t}{2}\mathscr{F}_2\left(\mathbf{U}^{n+1/3}\right)$$

$$\mathscr{F}_2\left(\mathbf{U}^{n+1/3}\right) = Adv + Cor + Hpg$$

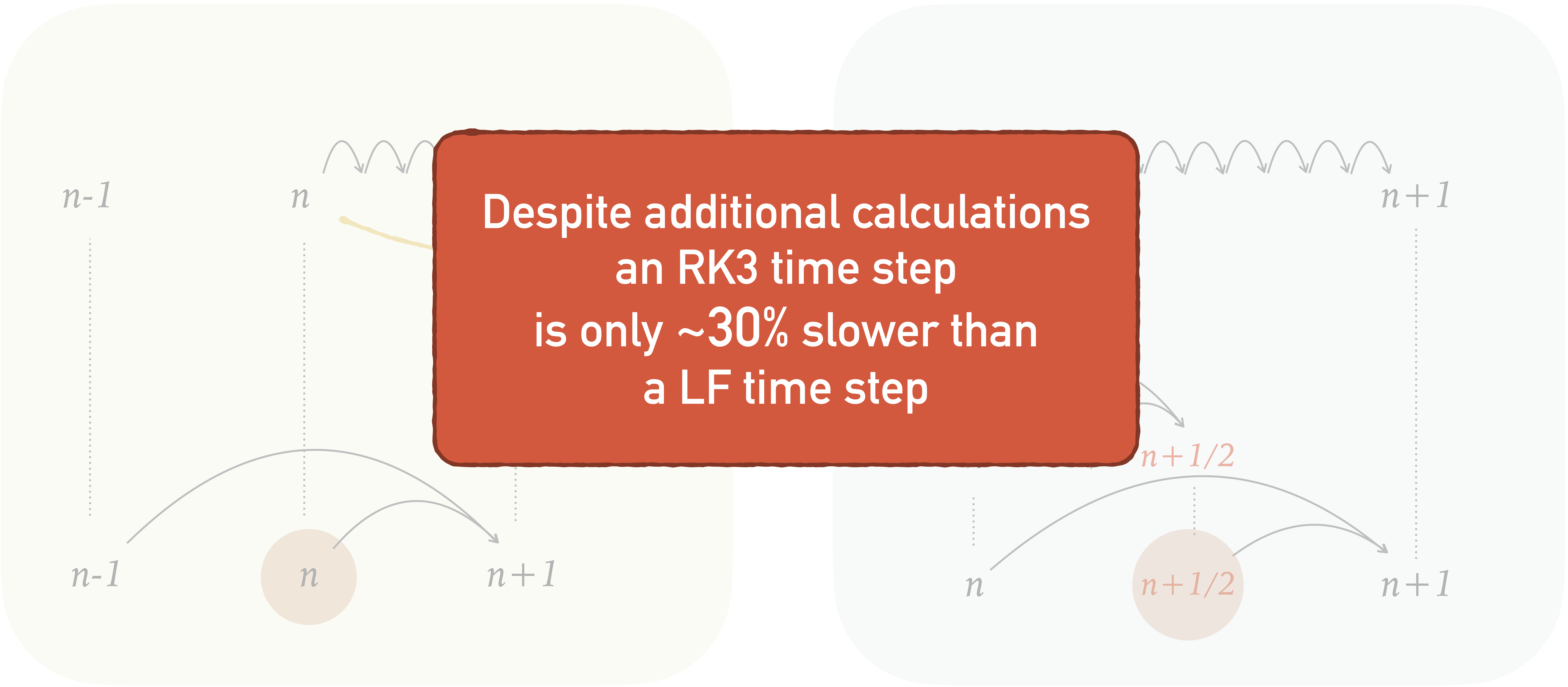
$+$ ~~$Ldf + Zdf + Frc$~~

$$\mathbf{U}^{n+1} = \mathbf{U}^{n} + \Delta t\mathscr{F}_3\left(\mathbf{U}^{n+1/2}\right)$$

$$\mathscr{F}_3\left(\mathbf{U}^{n+1/2}\right) = Adv + Cor + Hpg + Ldf + Zdf + Frc$$

Ducousso et al., 2024: Stability and accuracy of Runge–Kutta-based split-explicit time-stepping algorithms for free-surface ocean models, submitted to JAMES.

RK3 IN NEMO

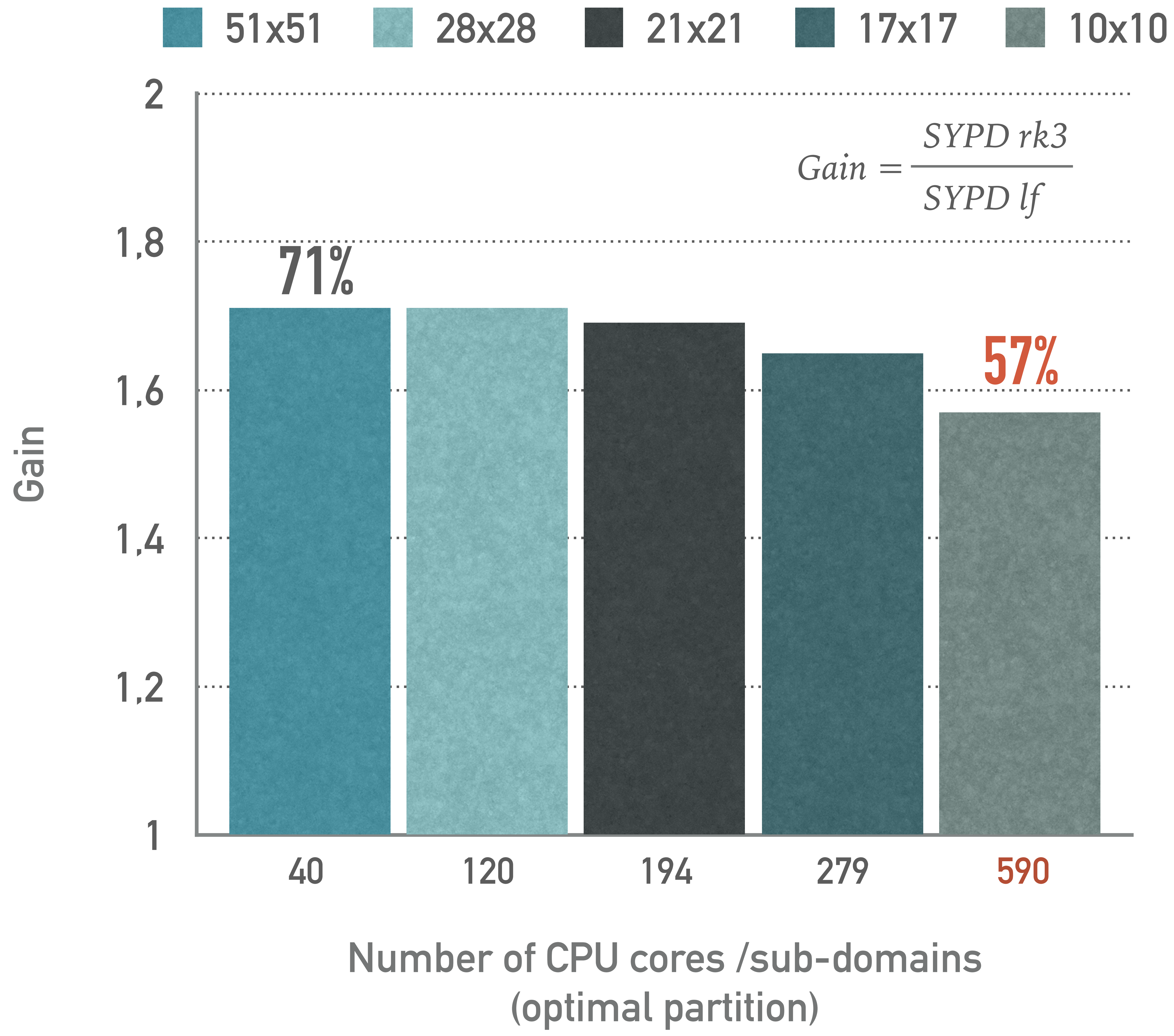


Despite additional calculations
an RK3 time step
is only ~30% slower than
a LF time step

PERFORMANCES: LF VS. RK3

eORCA1° OCE + ICE + I/O

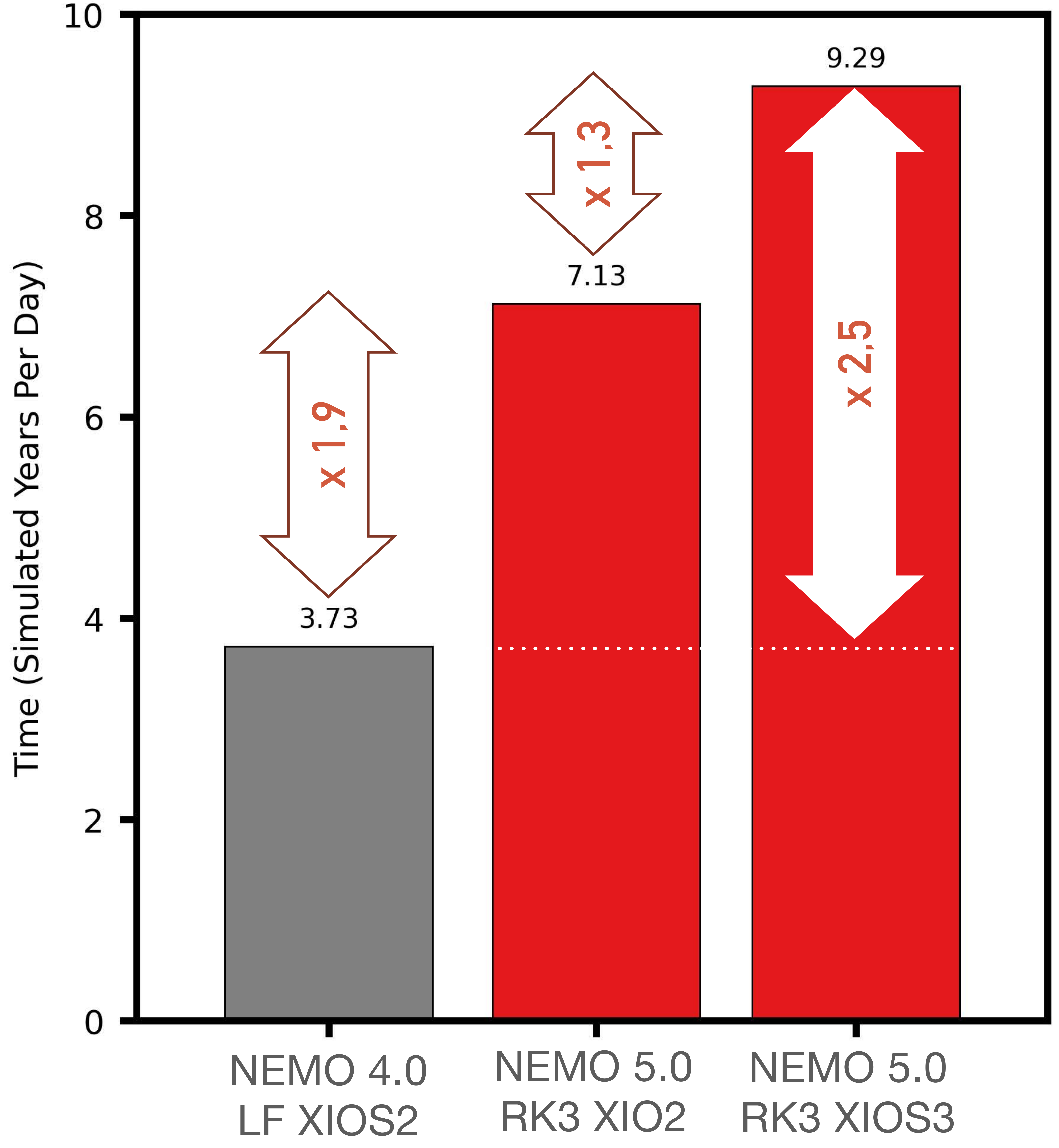
- 1 year simulation daily I/O
- Domain size: 360x331
- 75 vertical levels
- Time-to-solution to achieve a given accuracy (10 x 10)
  - Leap Frog v5: 119 SYPD
  - RK3 v5: 187 SYPD



OUTLINE

- Part 1: Kernel refactoring
- Part 2: LF to RK3 speed up
  - Theoretical stability constraints or RK3 strength
  - RK3 weakness
  - RK3 in NEMO
  - Performances
- **Part 3: I/O optimization**
- Part 4: Science results

ICING ON THE CAKE XIOS: V2 → V3



ORCA 1/4°

OCE + ICE + I/O

347 CPU cores

- IO server running on dedicated cpu cores
- Developed at IPSL

OUTLINE

- Part 1: Kernel refactoring
- Part 2: LF to RK3 speed up
  - Theoretical stability constraints or RK3 strength
  - RK3 weakness
  - RK3 in NEMO
  - Performances
- Part 3: I/O optimization
- **Part 4: Science results**

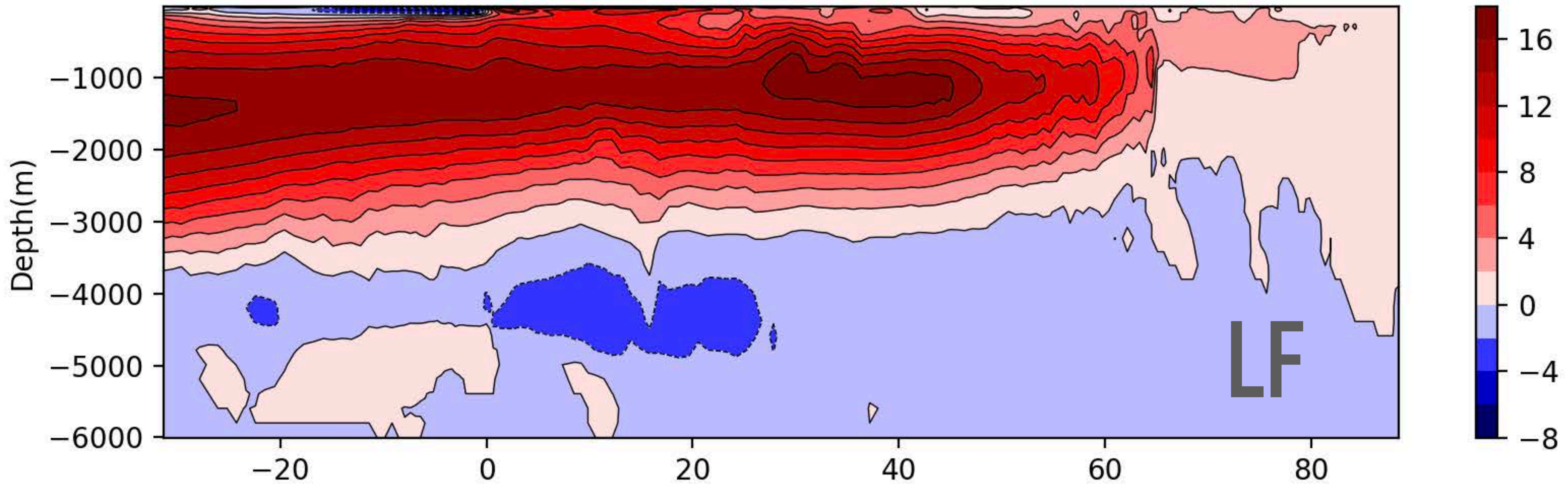
WHAT IS THE PHYSICAL RESPONSE OF THE OCEAN WITH RK3?

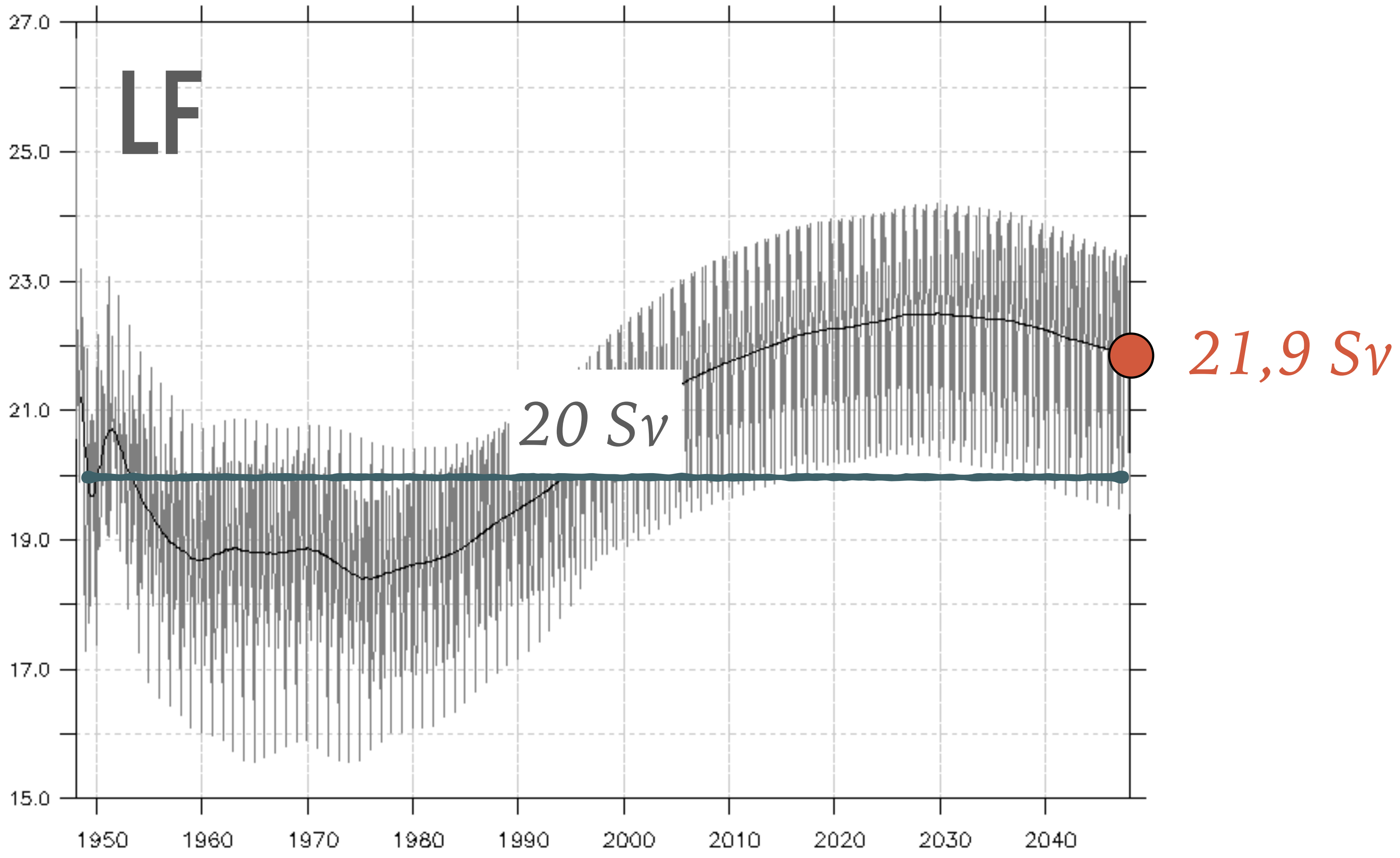
eORCA1° OCE + ICE + I/O

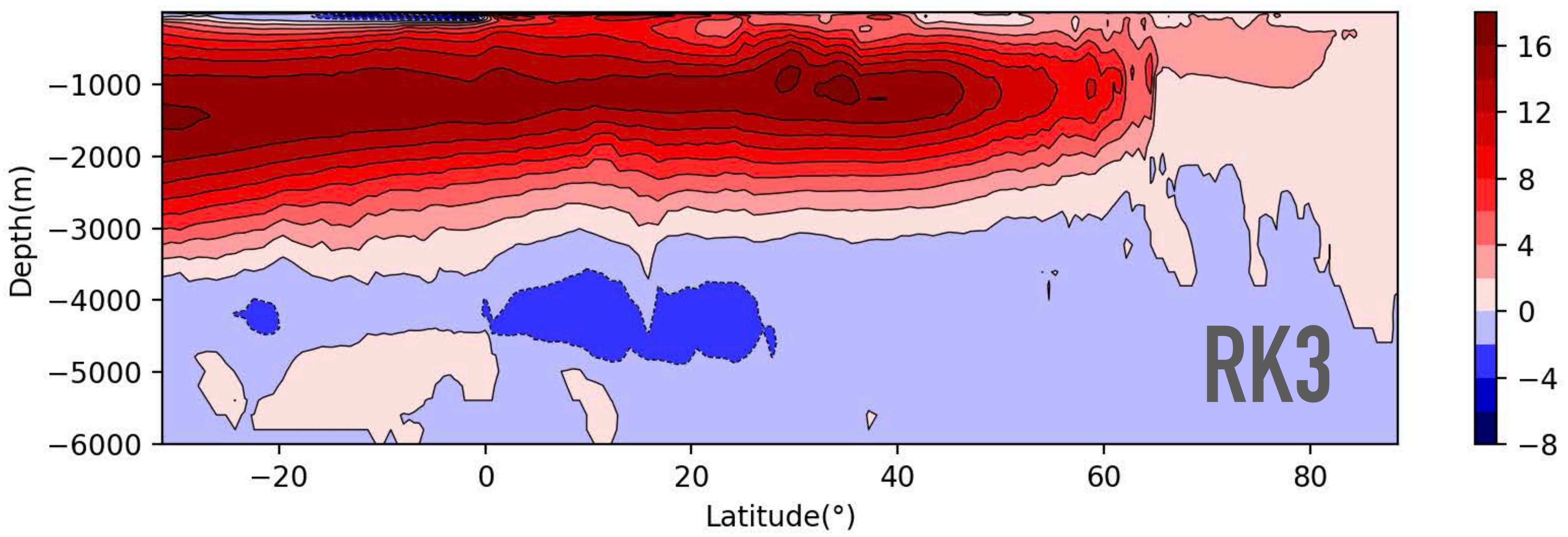
- on going validation
- 100 years
- forced climate

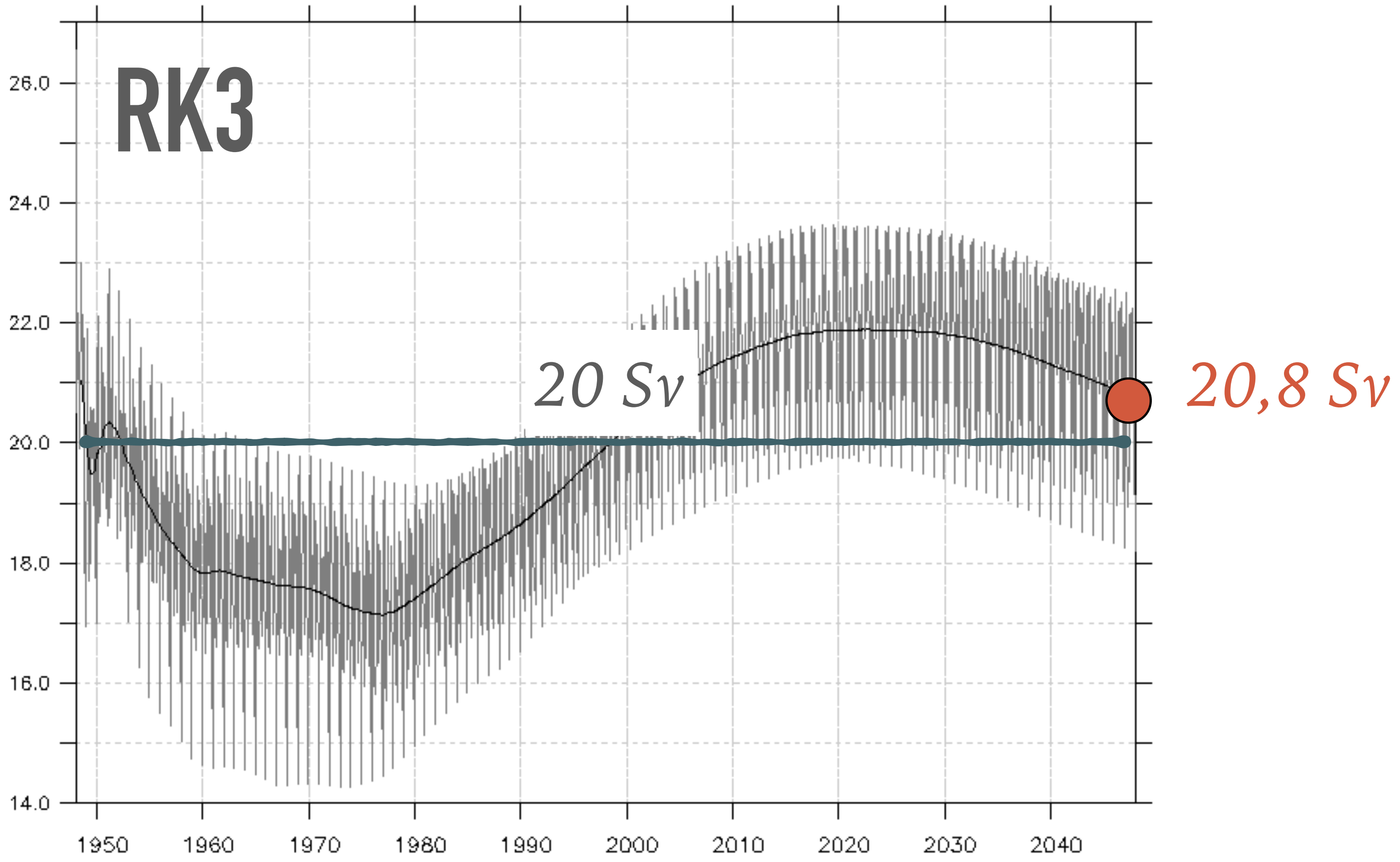
AMOC

(last 10 years mean)









CONCLUSION: V4 → V5 MORE THAN 2X FASTER

- Optimization +20%
  - ✓ Refactoring
- RK3 +40%
  - ✓ See Madec et.al 2024*
- I/Os +30%

- Perpective
  - Will these conclusions translate to the GPU case?

* Madec G., Lemarié F., Chanut J., Téchené S. et al., 2024. Implementation of a Runge-Kutta-based time-stepping algorithm in the NEMO ocean model : formulation, robustness and efficiency, in preparation for JAMES

THANKS FOR YOUR ATTENTION !

The end