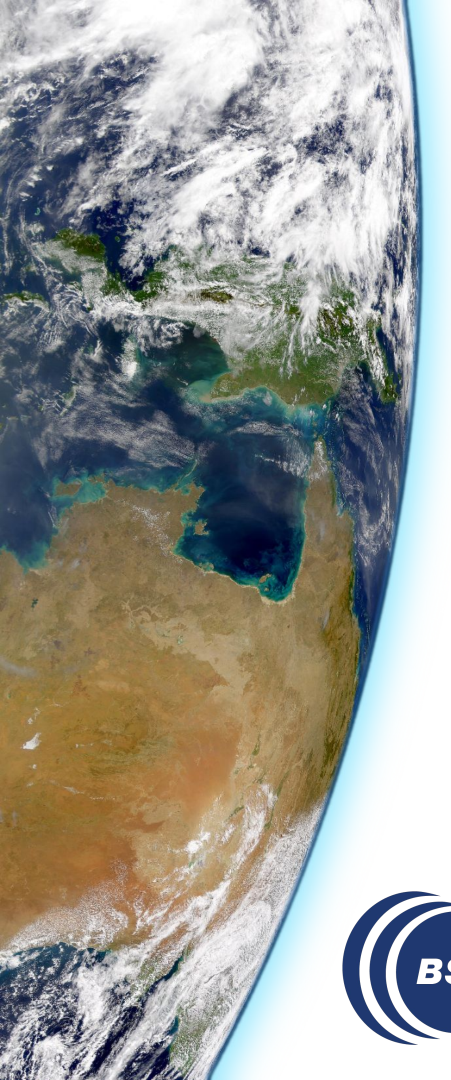


An evaluation of the mixed precision version of NEMO 4.0.1

Oriol Tintó, Stella Valentina Paronuzzi Ticco, Mario C. Acosta,

Miguel Castrillo, Kim Serradell and Francisco J. Doblas-Reyes

May 7th 2020

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

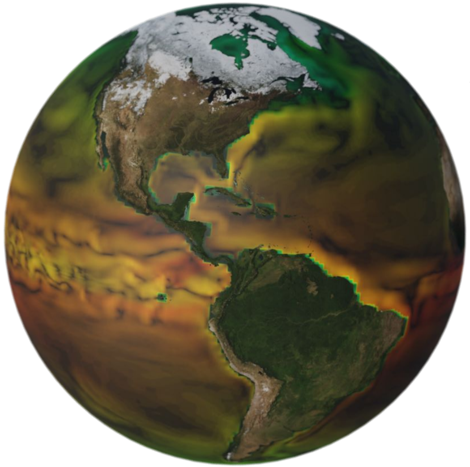
EGU General Assembly 2020

Why performance?

Making new experiments possible:

- Increasing model's resolution and complexity requires more computational resources, without proper development in the HPC capabilities of the model, it won't be possible to face future challenges like running simulations with ORCA at **1km** resolution (which will cost more than a million times the cost of ORCA1).

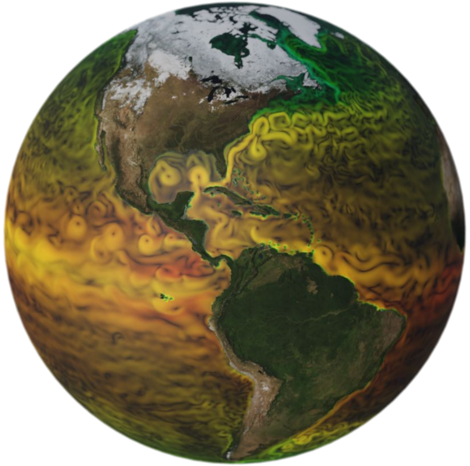
ORCA 1º **ORCA 1/4º** **ORCA 1/12º**



Cost

~ x 64

→



Cost

~ x 27

→

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EGU General Assembly 2020

Why performance?

Making current experiments cheaper:

- Given the amount of resources invested in simulations involving NEMO, improvements in the model performance always pay back in terms of resources, time and reduced carbon footprint:

Some numbers:

Nine modelling groups around Europe reported that their overall investment in CMIP simulations was around **1749.4 million core hours***.

*Data from **IS-ENES3**. While it is true that the simulations accounted for in these data included also other components (i.e. atmosphere), these are not taking into account simulations done for tuning, preparation nor OMIP experiments, so the actual value of resources invested in ocean simulation might be even higher. Source: Mario Acosta (BSC)

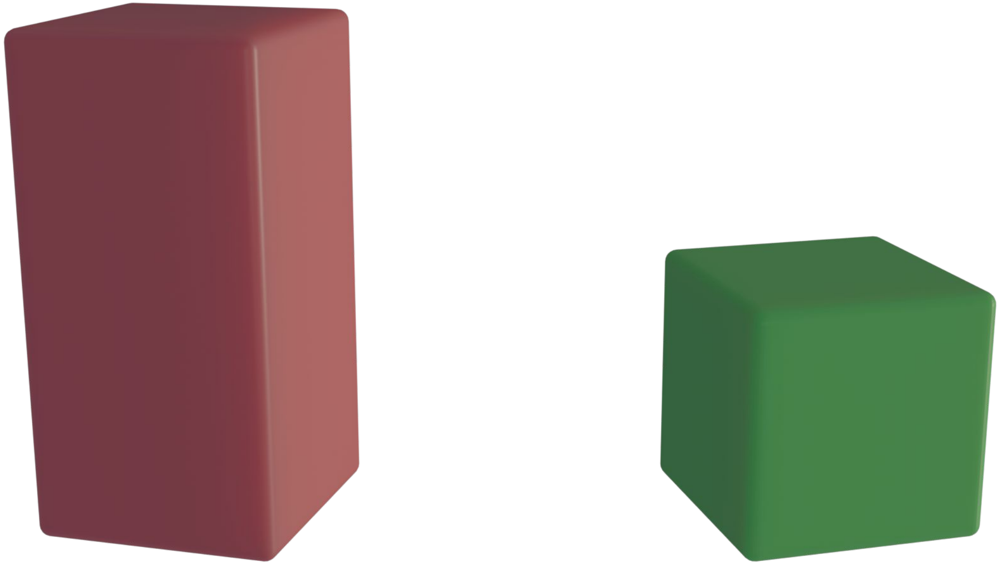
Why performance?

Investing in computational performance for our models is a **necessary step** to enable better science in the future and represents at the same time an opportunity to **save resources and reduce the environmental impact** of our experiments.

Working on computational performance is a **WIN-WIN** investment.

Why numerical precision?

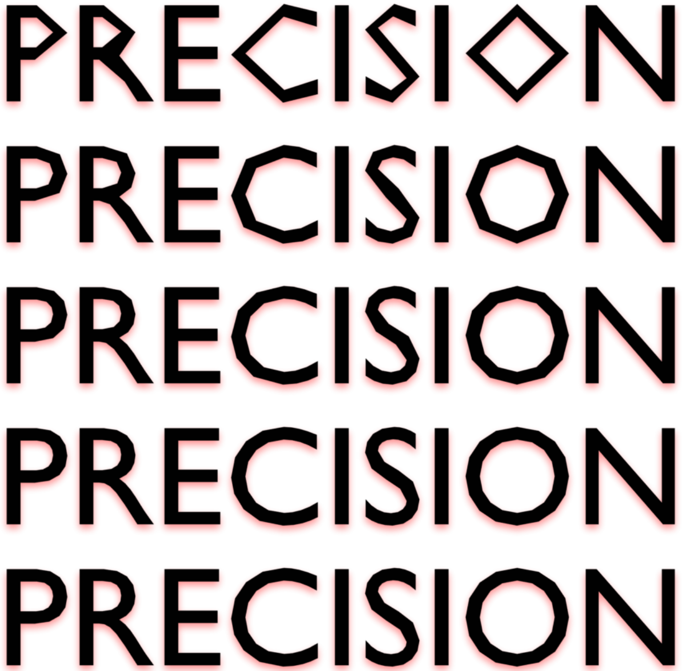
Since **NEMO** is a **memory bound** code, its performance would benefit from a reduction in memory footprint and data movements.



Why mixed precision?

Different variables/algorithms might require **different level of precision**.

If we use the same numerical precision throughout all the code, we will be forced to use the precision required at the most sensitive parts. As a consequence the precision in all the other parts will be **over engineered**.



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EGU General Assembly 2020

How can we identify which precision is required in the different regions of a code as big as NEMO?

Given a code, our method can identify **which variables require higher precision**.

The method is based on:

- The use of a **precision emulator**.
- A **search algorithm** to find sensitive variables.

It is **not exclusive** for NEMO and we plan to use it with **other models** in the future.

For more details : **Oriol Tintó Prims**, *Mario Acosta, Andrew M. Moore, Miguel Castrillo, Kim Serradell, Ana Cortés, Francisco J. Doblas-Reyes,*
How to use mixed precision in ocean models:
exploring a potential reduction of numerical precision in NEMO 4.0 and ROMS 3.6
Geoscientific Model Development, 2019, Volume 7, Pages 3135-3148, DOI:10.5194/gmd-12-3135-2019.

Testing on NEMO 4.0.1

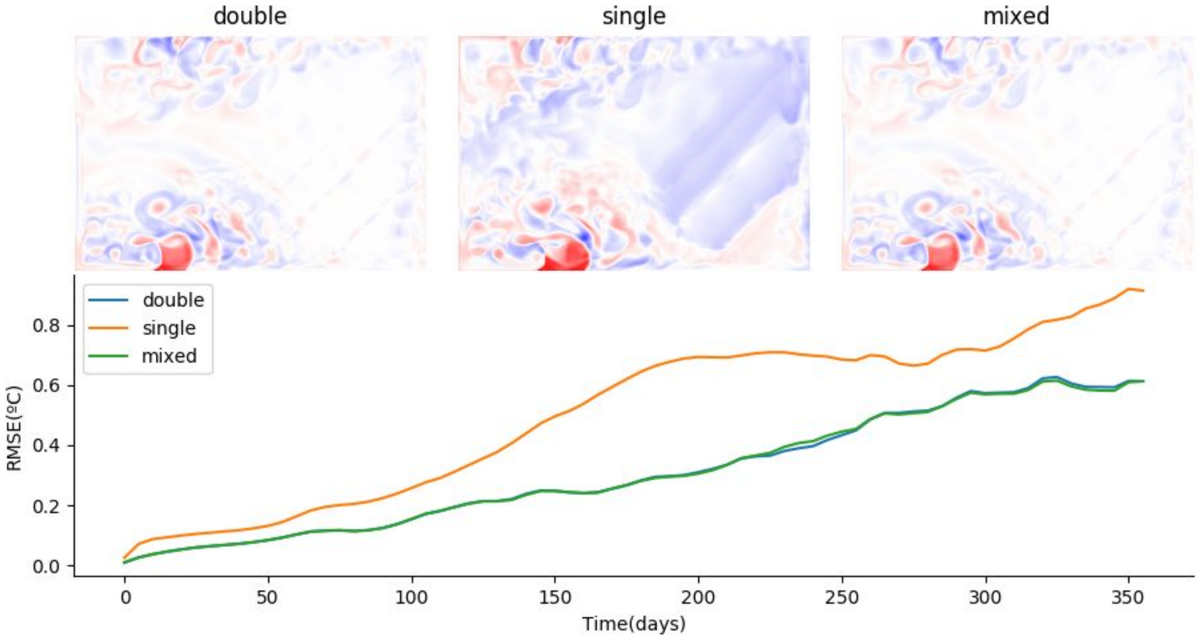
Experiment design:

- Run a reference simulation with **GYRE R27** (1/27º).
- Run three different simulations with **GYRE R9** (1/9º), one using double precision, another one using single precision and the last one using mixed-precision.
- Compare the differences between these three **R9** simulations and the **R27** reference.

Testing on NEMO 4.0.1

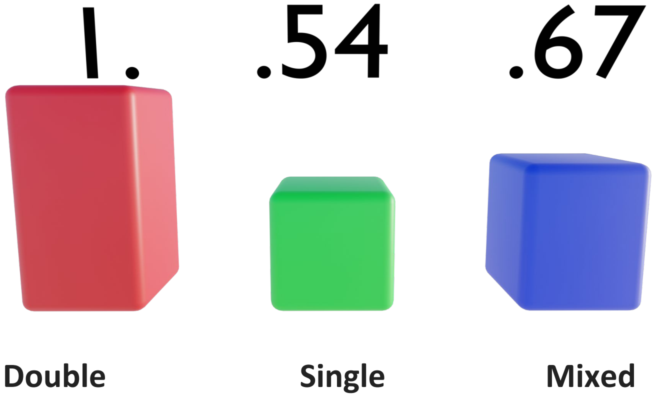
Results:

Impact of precision on sea-surface temperature in NEMO4:
comparison of GYRE1/9º simulations using different precisions



Testing on NEMO 4.0.1

Relative cost of the simulations:



Configuration:

GYRE R9 (272x182)

System:

A single node in Marenostrum-4 (48 cores)

Important points to have in mind

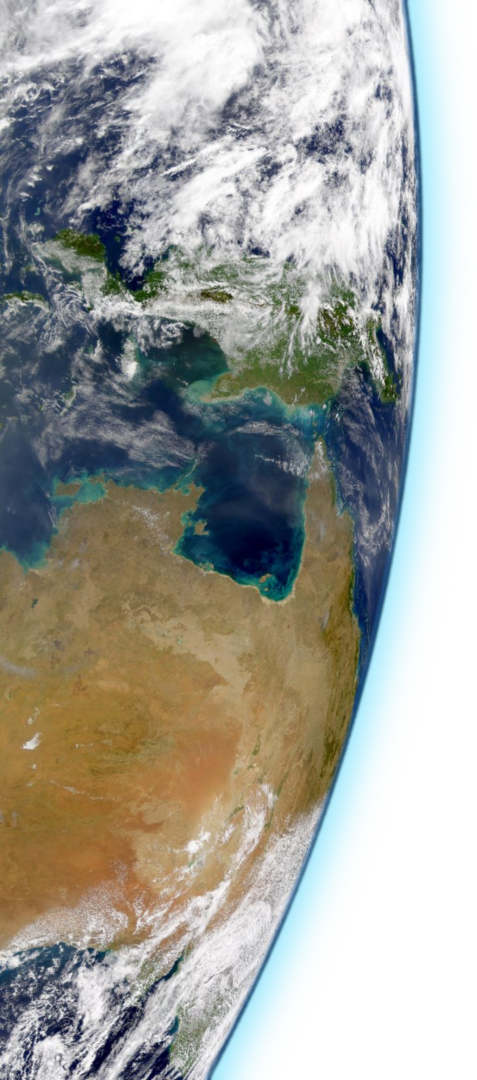
- This implementation …:
  - … was developed on top of **NEMO 4.0.1** to prove that:
    - Keeping the identified variables in double precision allows us to **keep model accuracy**.
    - Reducing the precision of all the other variables **increases the performance**.
  - … has **not been tuned yet**. Looking with more detail the parts that are kept in double precision it might be possible to find solutions to get closer to single-precision in terms of performance.
  - … is not the end of the story:
    - Future architectures will have a **16-bit floating point** type that will allow, in some parts, to reduce even further the precision.
    - The actual implementation will not go on top of 4.0.1 but the on top of the outcome of the last merge party.

Conclusions

- Applying mixed-precision to NEMO can provide **important savings while keeping the accuracy**.

Future Steps

- Implement the mixed-precision version in the NEMO repository for **version 4.2** .



Thank you!

For any question or comment please don't hesitate to contact me at:

oriol.tinto@bsc.es