Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 108C (2017) 776–785

Procedia Computer Science

International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland

Optimizing domain decomposition in an ocean model: the case of NEMO

Oriol Tintó[1,2], Mario Acosta[1], Miguel Castrillo[1], Ana Cortés[2], Alícia Sanchez[1], Kim Serradell[2], and Francisco J. Doblas-Reyes[1,3]

[1] Barcelona Supercomputing Center, Barcelona, Spain
[2] Universitat Autònoma de Barcelona, Barcelona, Spain
[3] ICREA, Pg. Llus Companys 23, 08010 Barcelona, Spain
oriol.tinto@bsc.es

Abstract

Earth System Models are critical tools for the study of our climate and its future trends. These models are in constant evolution and their growing complexity entails an incrementing demand of the resources they require. Since the cost of using these state-of-the-art models is huge, looking closely at the factors that are able to impact their computational performance is mandatory. In the case of the state-of-the-art ocean model NEMO (Nucleus for European Modelling of the Ocean), used in many projects around the world, not enough attention has been given to the domain decomposition. In this work we show the impact that the selection of a particular domain decomposition can have on computational performance and how the proposed methodology substantially improves it.

Peer-review under responsibility of the scientific committee of the International Conference on Computational Science

Keywords: Earth System Models, HPC, domain decomposition, NEMO

1 Introduction

Climate change and the threat that it poses to the humankind and to the entire planet has become one of the main challenges for the society. For that reason, it is not coincidence that the understanding of the climate system is a subject of growing interest and a hot topic that has even moved to the center of political actuality. There are different tools available to scientists to study climate change, from which Earth System Models (ESMs) are the most powerful ones. ESMs are complex systems made up by one or more mathematical models describing different parts of the Earth System. These models simulate the atmosphere, ocean, sea-ice and the other components, usually involving the resolution of differential equations that describe the behaviour of these systems.

These models have been in constant evolution over the last decades and have considerably grown in complexity, including new phenomena or solving better old features. One of the factors that helped to represent better simulated processes is the increment of grid resolution, that is,

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the International Conference on Computational Science
10.1016/j.procs.2017.05.257

the vertical and horizontal space where the equations are solved. For example, several works have proved that enhanced horizontal resolution can lead to significant changes in large-scale aspects of circulation as well as improvements in small-scale processes and extremes [6] [1]. This fact is consistent with other studies showing that an increment of resolution is useful to reduce biases [12]. In the ocean, some of the processes that are better represented at higher resolution are boundary currents, water exchange through narrow straits, coastal currents, upwelling, oceanic eddies [15] [3] [7], ENSO [10] [14] [13], and sea-ice drift and deformation [16] [5]. Nevertheless, the additional complexity has demanded an increase of the computational resources required to run these models, reducing the range of machines in which they can work to the domain of supercomputers. The consequence is that, although in theory the usage of the highest possible resolution for ESMs grids would benefit scientific performance of simulations, at the time of defining an experiment there is usually a tradeoff between scientific interest and computational cost. In other words, the computational requirements of an ESM determine both the amount of resources that are needed to perform an experiment and also the kind of experiments that can realistically be done. Having a computationally efficient model is not just a way to use the resources wisely, but a manner to allow better experiments. And even more, the possibility to keep increasing the complexity of these models will exist only if the way they can use the new architectures and technologies is improved.

Like any other software, in order to execute ESMs in supercomputers, they must first be parallelized. ESMs that have been for more than two decades in development had already completed this process in late eighties [11] [9], to start taking profit of multi-core systems when they were first appearing. They usually exploited domain decomposition strategies, in which the different processes execute the same code, performing the same work on different sub-domains. Which is the domain to divide in the case of ocean models? Since the Earth is not a regular sphere made of water, when discretizing the ocean, models have to deal with the fact that the planet Earth has emerged land areas and the ocean has different depths. If we try to describe the domain as a regular grid, we will have grid-points where there is no water, because they are located over the land or because they are located under the ocean bottom. A way to deal with this, consists in using a mask to determine which points of a regular grid belong to ocean or land.

Using this approach, when it is time to compute over the grid there are two main possibilities: the first one is to perform calculations over the whole domain and, afterwards, use the mask to omit the results over land-points and, the second one, consist in, before doing any computation, check if the point about to be computed corresponds to a sea-point or not. Between these two options, the former presents one main disadvantage and many advantages with respect to the latter. The disadvantage is that the model does some useless computation, and the advantages are that the regular pattern of the memory accesses and the lack of conditionals benefits the computational performance by allowing the exploitation of many modern processor features like pipelining and vectorization. Contrarily, the use of conditionals inside computation loops prevents vectorization and can cause pipeline bubbles that can seriously hurt performance. On the other hand, a complete different approach consists in using a non-regular grid in which only the ocean-points are defined. This approach could be a better option if taking into account aspects like memory usage (useless land points are not allocated in memory) and better load balance among subdomains. However, these advantages have a direct impact in the model discretization, adding complexity in the implementation of the equations solving. For this reason, this approach has not been widely explored.

Even though the compute everything choice has many advantages, it can lead to a ridiculous situation. Applying a regular domain decomposition can produce some subdomains where all

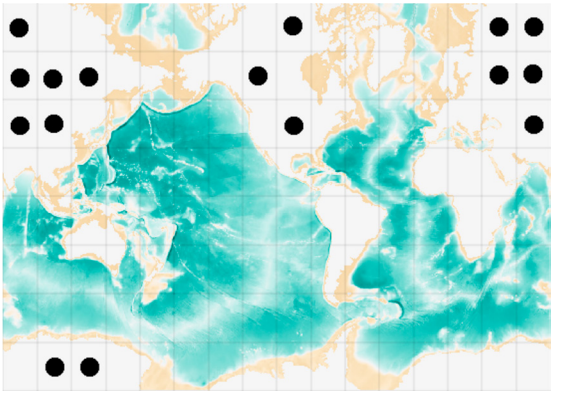

Figure 1: Domain decomposition of a tripolar grid of the ORCA family with a resolution of a of degree into 128 subdomains (16 x 8). Subdomains marked with a black dot do not contain any ocean grid point.

the points correspond to land, and consequently, all the computation done in that specific subdomain is useless. Figure 1 shows this situation, with a particular case in which 12 % of the depicted subdomains do not contain any sea-point and, consequently, the work done at these subdomains is completely useless. Anyhow, a way to exploit the advantage of having a regular domain decomposition without suffering the penalty of considering only-land subdomains as computational elements, consists in removing these subdomains from the set of subdomains to compute. This simple strategy can save a significant part of the computational resources but it is not the only thing to take into consideration.

In this work, we propose a methodology to determine the best domain decomposition taking into account the portion of the domain that can be omitted and other aspects as the grid overlapping required for satisfying boundary conditions and the resources available.As a case study we have used the state-of-the-art ocean model NEMO[8] (Nucleus for European Modelling of the Ocean), that was the ocean component of many of the ESMs taking part in the last Coupled Model Intercomparison Project (CMIP) experiment for the Intergovernmental Panel on Climate Change and will be in 18 of the 37 ESMs participating in the next CMIP6. It has thousands of users and it is being developed by a broad number of scientists from many institutions, mainly in Europe. Since so many resources are invested in simulations involving the NEMO model, to squeeze at maximum its domain decomposition is a relevant point. In this state-of-the-art ocean model, it is possible to switch off the land-only processes with the aim of not wasting resources on it. However, the model itself does not take advantage of this capability and its activation requires a fine-tuning from the user.

The next section describes how the domain decomposition is currently done in NEMO and the proposed methodology to determine the best domain decomposition for the ocean model. The experimental results of the improvements in the NEMO performance and in ESMs in general are shown in section 3. Finally, section 4 states the main conclusions of this work

2 Methodology

The problem of finding an optimal domain decomposition can be defined as a multi-objective optimization problem since there are multiple factors to optimize that are conflicting and, for that reason, there is no unequivocal optimal solution. Even more, the resulting layout may have to fit into limited resources. To understand the problem that we want to solve, the section 2.1 explains how the domain decomposition is done in the NEMO model, in section 2.2 a method to find decompositions that adjust the user interests is presented.

2.1 Domain decomposition in NEMO

For global simulations, NEMO uses ORCA grids. They are a grid family for global models whose common characteristic is that they are tripolar, with the aim of avoiding numerical singularities at the north pole. The geographic south pole is conserved and from 80°S to 20°N the grids are like a regular Mercator-grid. By contrast, in the north hemisphere two poles are used, placing them over land, one in Canada and the other in Asia. From 20°N to the north-poles, the circles of the Mercator-grid are progressively distorted into ellipses. For the third-dimension of the grid, the earth is assumed to be spherical and the vertical dimension is considered to be perpendicular to the surface.

The resolution of these grids is not homogeneous, and the denomination of the different ORCA grids refer to their resolution at the latitude at its coarsest part, that is the equator i.e. ORCA2, ORCA1, ORCA025, ORCA12, etc. The number of points of the grid is $jpiglo \times jpjglo \times jpk$ where $jpiglo$ is the number of grid points in the longitudes direction, $jpjglo$ is the number of grid points in the latitudes direction and jpk is the number of vertical levels. The specific values of these variables depend on the resolution of the grid (See Table 1).

Name	jpiglo	jpjglo	jpk	Horizontal resolution (km)	TotalSize (Million points)
ORCA2	182	149	32	219.8	0.86
ORCA1	362	292	75	110.5	7.92
ORCA025	1442	1021	75	39.1	110.42
ORCA12	4322	3059	75	9.3	991.57

Table 1: Parameters of some ORCA grids with different resolutions.

The model was parallelized and is able to run on both shared and distributed memory environments, using a 2D-Pencil domain decomposition method (Figure 2) that splits the global three-dimensional domain in two dimensions, keeping all the vertical levels of the global domain in the same subdomain. Each one of the subdomains is executed in a different processor core and it uses the Message Passing Interface message passing system to manage the necessary communications between subdomains. These include not only the corresponding part of the global domain but also the halos, that are used to satisfy the boundary conditions, containing information from the border points of the neighbour subdomains. We refer to these points that are represented twice as overlapping. Due to the overlapping, the sum of the size of the subdomains is bigger than the original domain.

The global domain of size $jpiglo \times jpjglo \times jpk$ is decomposed into $jpni \times jpnj$ subdomains of size $jpi \times jpj \times jpk$ where $jpi = \frac{jpiglo}{jpni} + 2 \cdot haloSize$ and $jpj = \frac{jpjglo}{jpnj} + 2 \cdot haloSize$. The user can specify the jpi and jpj himself or can leave these values empty and let the model find a particular decomposition.

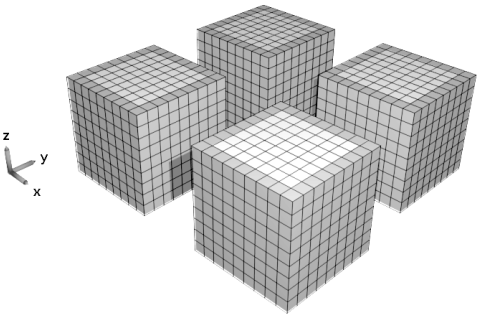

Figure 2: 2D pencil decomposition of a 3D grid. Light gray cubes represent the inner domain and dark gray cubes correspond to halo regions.

As it was explained before, some of the subdomains can end-up containing only land points (See figure 1), so it is possible to eventually exclude these subdomains from the calculus. However, despite the model being able to exclude land-only subdomains, the way this option can be activated is not straightforward. To enable it, the user has to specify the values for $jpni$ and $jpnj$ and also the number of subdomains that contain at least one sea-point, such that $seaSubdomains < jpni \times jpnj$. Finally, at the time of execution $seaSubdomains$ must coincide with the number of launched processes.

2.2 About the proposed method

Since the domain decomposition is done to distribute the work among different processes, one of the main targets is to distribute the work as much as possible with the objective of reducing the computing resources required by each subdomain. Further dividing the domain will have two additional consequences, on one hand the proportion of land-only subdomains that can be omitted in computation will increase, but on the other hand the part of the domain that corresponds to overlapped areas will have more significance. These two factors determine the size of the computed domain, which is the sum of the sizes of all the subdomains.The objective is then to minimize at the same time the **subdomain size** and the **computed domain**, which depends on both the proportion of land-only subdomains and the overlapping.

To assess how good a specific decomposition is, regarding the resulting computed domain, we can use the ratio between the computed domain and the original domain size . This ratio, that we will call **gridFactor**, can be expressed as the product of the overlapping factor by the proportion of sea-subdomains. The **overlapping** factor is the proportion of the sum of all the subdomains to the size of the global domain, and the **proportion of sea-subdomains** is the ratio of the subdomains that contain at least one sea-point to the total number of subdomains.

$$overlappingFactor = \frac{subdomainSize \cdot totalSubdomains}{globalDomainSize} \tag{1}$$

$$seaSubdomainProportion = \frac{seaSubdomains}{totalSubdomains} = 1 - \frac{landSubdomains}{totalSubdomains} \tag{2}$$

The subdomain size depends not only on the total number of subdomains but on the specific decomposition, i.e. the number of divisions in each dimension (latitudes and longitudes). It will be then the number of points in the longitudes direction ($jpiglo$) divided by the number of divisions in that direction ($jpni$) plus the halo, multiplied by the number of points in the latitude direction ($jpjglo$) divided by the number of divisions in that direction ($jpnj$) plus the halo. For implementation specifics, the formula used in the model is the following.

$$seaSubdomainSize = \left(\frac{jpiglo - 2 \cdot halo + (jpni - 1)}{jpni} + 2 \cdot halo\right) \cdot \left(\frac{jpjglo - 2 \cdot halo + (jpnj - 1)}{jpnj} + 2 \cdot halo\right) \tag{3}$$

Then the **gridFactor** is:

$$gridFactor = seaSubdomainProportion \cdot overlappingFactor$$

If our intention was only to minimize the overall work, the **gridFactor** index would be enough but since we are also interested in minimizing the **subdomainSize** we can define a fitness function that takes into account both elements:

$$fitness = gridFactor^{\alpha} \cdot subdomainSize^{\beta}$$

where α and β are the exponents that we give to each factor. With $\alpha = 0$ and $\beta > 0$ we are considering better the decompositions that minimize the subdomain-size, with $\alpha > 0$ and $\beta = 0$ we are considering better the decompositions that minimize the computed domain.

To compute the fitness function of a specific decomposition, the only parameter that we can not know a priori is the number of land subdomains, that must be computed using the bathymetry information. To find how many of the subdomains are land-only, the process consists in building the subdomains and looking if there are any sea-points inside.

With this information we can compute the fitness of any decomposition and estimate which one is better for our interests. There is yet another problem that needs to be solved; while it is straightforward to find the fitness of a specific domain decomposition, answering the question of which decomposition has the best fitness with a limited number of resources is not trivial. Since we can omit the land subdomains, the number of subdomains will be reduced, and therefore decompositions that would not fit into a specific number of resources without removing these processes may fit if these are discarded. To find the decomposition that fits in the resources and has the best fitness value we can perform a search of all the possible decompositions, sort the resulting list by fitness and filter the ones that have more subdomains than available processes.

The process of finding the best decomposition will be:

- For each possible number of subdomains between 1 and the maximum number of available cores plus 45%, find all the possible factorizations of two elements using integers (for instance, 100 = 10 x 10 = 20 x 5 = 25 x 4 = 50 x 2 = 100 x 1). The value of 45% comes from the fact that approximately 30% of the surface points correspond to land, so combinations that have a number of subdomains bigger than 145% of the available resources will not fit even if it is possible to remove all the land points.
- For each case compute its fitness and the number of sea-subdomains.
- Sort the list by fitness.

- When it is time to run the model, remove from the list the combinations where the number of sea-subdomains is bigger than the number of available processes.
- The first combination of the list is the best option.

The whole process can be done once for each specific bathymetry and a ranking with the fitness of the different decompositions can be saved. In that manner, if the number of available resources changes, it is not necessary to repeat all the analysis.

3 Results

Results commented in this section come from simulations performed on the Marenostrum 3 supercomputer, hosted by the Barcelona Supercomputing Center. It has a peak performance of 1,1 Petaflops, with 48,896 Intel Sandy Bridge processors in 3,056 nodes. A The metric used to quantify the model throughput is Simulated Years per Day ($SYpD$) that is the number of model years that can be simulated in a period of one day of wall-clock time [2]. This metric is widely used in climate science because it is an important measure of the ability to perform research in a timely manner. A general consensus is that 5 years per day is the minimum acceptable useful simulation rate [4].

Case	jpni	jpnj	Computed Domains	Throughput
With land processes	64	32	2048	5.47 SYpD
Without land processes	64	32	1533	7.15 SYpD

Table 2: Impact of land-process removal.

In table 2 we compared the throughput of two ORCA025LIM3 simulations using 2048 cores and the same domain decomposition (jpni=64,jpnj=32). In one case the land subdomains are included and the other avoids its computation. In the first case the model achieves a throughput of 5.47 $SYpD$ and in the second, where the 515 subdomains that only contained land-points are removed, the achieved throughput is 7.15 $SYpD$. Summarizing, using 25% less resources the model goes 30% faster, resulting in an efficiency improvement of 74%.

The Figure 3 shows the ratio between the computed and global domain sizes. This figure shows that the NEMO algorithm to find a domain decomposition in most of the cases ends up using a really bad decomposition as a solution that increments hugely the size of the computed domain, and it is merely random if one specific number of subdomains leads to either a good or a bad decomposition. On the other hand, the method proposed in section 2.2 always maximizes the fitness function for any given number of subdomains. Since the method proposed evaluates all the possible decompositions that fit with a specific number of resources, it will always outperform the default NEMO decomposition.

The figure 4 shows the model throughput of real simulations using an ORCA025 grid with the NEMO's default decomposition and with the proposed method, It can be seen how the proposed methodology impacts the performance of real simulations achieving a speed-up thanks to the reinvestment of the resources saved by land process removal. In the case of having up to 2048 processor cores available, the default decomposition chosen by NEMO is 64 x 32 domains, achieving a throughput of 5.47 SYpD, while with the proposed method the decomposition of 64 x 47 achieves 7.73 SYpD, 41% more throughput. With the double of resources (4096 processor cores) and being 64 x 64 the default decomposition, the measured throughput was 5.25 SYpD, which is even slower than the 2048 case. Nevertheless, using the proposed method

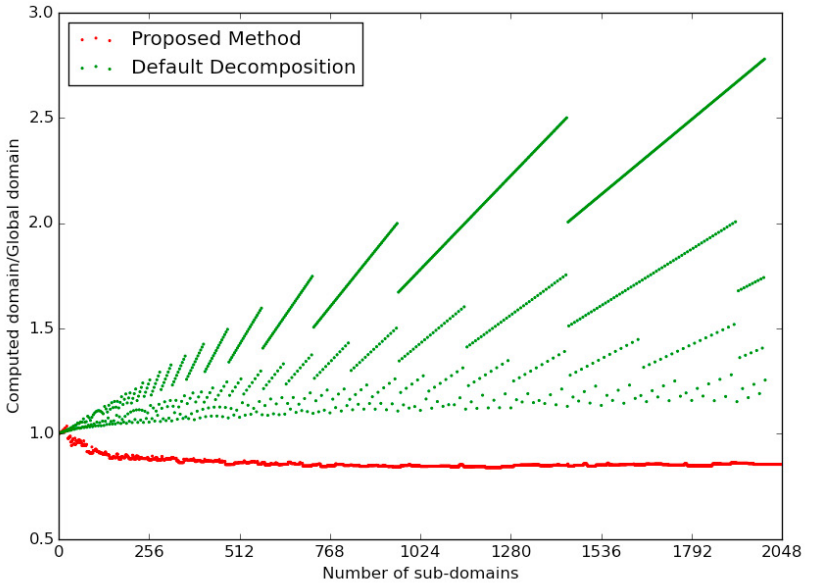

Figure 3: Ratio between the computed domain size and the global domain size for the domain decompositions used for each number of subdomains in a ORCA025 grid. In green the cases using the NEMO's default algorithm, that do not remove land-only subdomains, and in red the decompositions found using the method proposed, removing the land-only subdomains. The red data samples corresponds to domain decompositions found using the fitness function with $\alpha = 1$ and $\beta = 1$.

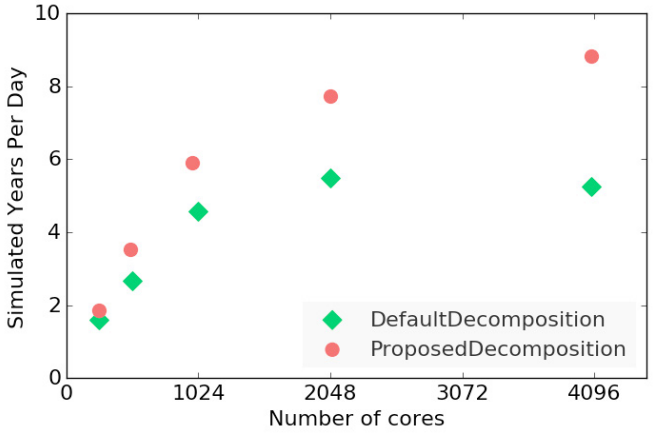

Figure 4: Model throughput using the model's default decomposition without land process removal and using the proposed domain decomposition with land process removal.

the throughput was 8.81 SYpD using an 85 x 68 decomposition, which is faster than the 2048 case.

4 Conclusions

Regular domain decomposition is sometimes the best option for Earth System computational models. However, results shown in this work prove that the impact of the domain decomposition and the removal of the land-processes have been alarmingly underestimated.

In the case of NEMO, the model evaluated and which is used and supported by a large community around the world, the default domain-decomposition given by the model's algorithm can lead to decompositions that are quite far from the optimal because they do not minimize the overlap between the subdomains and do not maximize the possibility to remove land-only processes. From the measures performed in real simulations, shown at section 3, we can get two important conclusions. In first place, the fact that removing the land-only subdomains the model has better throughput (30% faster using 25% less resources) indicates that these processes were not only doing useless computation but also harming the overall performance, since they were increasing communication overhead due to their participation in collective communications. In second place, the proposed method outperforms the model default domain decomposition and it is useful to find the best domain decomposition for a specific number of available resources. As an example, we achieve 41% gain in throughput for ORCA025LIM3 simulations with 2048 available processor cores. We have shown that taking into consideration both the overlapping and the possibility to remove land-only sub-domains it is not only possible to reduce the resources used but also to achieve performance gains in throughput. Using this methodology we are able to exploit at the maximum the available resources, by speeding-up the model or by reducing the necessary resources to achieve the target throughput. Over all these facts, the most important conclusion is that the proper selection of a domain decomposition can not be ignored any longer.

Acknowledgements

The research leading to these results has received funding from the EU H2020 Framework Programme under grant agreement n^o 675191 and by the Ministerio de Economa y Competitividad (MINECO) under grant TIN2014-53234-C2-1-R

References

[1] Mario Acosta, Mancia Anguita, F. Javier Fernández-Baldomero, Cintia L. Ramón, S. Geoffrey Schladow, and Francisco J. Rueda. Evaluation of a nested-grid implementation for 3D finite-difference semi-implicit hydrodynamic models. *Environmental Modelling and Software*, 64:241–262, 2015.

[2] Venkatramani Balaji, Eric Maisonnave, Niki Zadeh, Bryan N. Lawrence, Joachim Biercamp, Uwe Fladrich, Giovanni Aloisio, Rusty Benson, Arnaud Caubel, Jeffrey Durachta, Marie-Alice Foujols, Grenville Lister, Silvia Mocavero, Seth Underwood, and Garrett Wright. CPMIP: measurements of real computational performance of Earth system models in CMIP6. *Geoscientific Model Development*, 10(1):19–34, 2017.

[3] Thomas L. Delworth, Anthony Rosati, Whit Anderson, Alistair J. Adcroft, V. Balaji, Rusty Benson, Keith Dixon, Stephen M. Griffies, Hyun Chul Lee, Ronald C. Pacanowski, Gabriel A. Vecchi, Andrew T. Wittenberg, Fanrong Zeng, and Rong Zhang. Simulated climate and climate change

in the GFDL CM2.5 high-resolution coupled climate model. *Journal of Climate*, 25(8):2755–2781, 2012.

[4] John M Dennis, Mariana Vertenstein, and Anthony P Craig. Performance Evaluation of Ultra-High-Resolution Climate Simulations. In *10th LCI International Conference on High-Performance Clustered Computing*, 2009.

[5] Peter R. Gent, Stephen G. Yeager, Richard B. Neale, Samuel Levis, and David A. Bailey. Improvements in a half degree atmosphere/land version of the CCSM. *Climate Dynamics*, 34(6):819–833, 2010.

[6] Reindert J Haarsma, Malcolm J Roberts, Pier Luigi Vidale, A. Catherine, Alessio Bellucci, Qing Bao, Ping Chang, Susanna Corti, Neven S. Fu??kar, Virginie Guemas, Jost Von Hardenberg, Wilco Hazeleger, Chihiro Kodama, Torben Koenigk, L. Ruby Leung, Jian Lu, Jing Jia Luo, Jiafu Mao, Matthew S. Mizielinski, Ryo Mizuta, Paulo Nobre, Masaki Satoh, Enrico Scoccimarro, Tido Semmler, Justin Small, and Jin Song Von Storch. High Resolution Model Intercomparison Project (HighResMIP v1.0) for CMIP6. *Geoscientific Model Development*, 9(11):4185–4208, 2016.

[7] R. Justin Small, Enrique Curchitser, Katherine Hedstrom, Brian Kauffman, and William G. Large. The Benguela upwelling system: Quantifying the sensitivity to resolution and coastal wind representation in a global climate model. *Journal of Climate*, 28(23):9409–9432, 2015.

[8] Gurvan Madec. NEMO Ocean Engine. *Note du Pole de modelisation*, (27):1288–1619, 2006.

[9] Gurvan Madec, Pascale Delecluse, Maurice Imbard, and Claire Levy. OPA 8 . 1 Ocean General Circulation Model reference manual. *Notes du Pôle de Modélisation, Institut Pierre Simon Laplace*, (11):97, 1998.

[10] Yukio Masumoto, H Sasaki, T Kagimoto, N Komori, a Ishida, Y Sasai, T Miyama, T Motoi, Humio Mitsudera, and K Takahashi. A fifty-year eddy-resolving simulation of the world ocean: Preliminary outcomes of OFES (OGCM for the Earth Simulator). *Journal of the Earth Simulator*, 1(April):35–56, 2004.

[11] Carlos R. Mechoso, Chung-Chun Ma, John D. Farrara, Joseph A. Spahr, and Reagan Moore. Parallelization and Distribution of a Coupled AtmosphereOcean General Circulation Model, 1993.

[12] Chloe Prodhomme, Lauriane Batté, Franois Massonnet, Paolo Davini, Omar Bellprat, Virginie Guemas, and Francisco J. Doblas-Reyes. Benefits of Increasing the Model Resolution for the Seasonal Forecast Quality in EC-Earth. *Journal of Climate*, 29(24):9141–9162, 12 2016.

[13] T. Rackow, H. F. Goessling, T. Jung, D. Sidorenko, T. Semmler, D. Barbi, and D. Handorf. Towards multi-resolution global climate modeling with ECHAM6-FESOM. Part II: climate variability. *Climate Dynamics*, 2016.

[14] Richard D Smith, Mathew E Maltrud, Frank O Bryan, and Matthew W Hecht. Numerical Simulation of the North Atlantic Ocean at 1 / 10 °. *Journal of Physical Oceanography*, 30:1532–1561, 2000.

[15] Takashi T. Sakamoto, Yoshiki Komuro, Teruyuki Nishimura, Masayoshi Ishii, Hiroaki Tatebe, Hideo Shiogama, Akira Hasegawa, Takahiro Toyoda, Masato Mori, Tatsuo Suzuki, Yukiko Imada, Toru Nozawa, Kumiko Takata, Takashi Mochizuki, Koji Ogochi, Seita Emori, Hiroyasu Hasumi, and Masahide Kimoto. MIROC4hˆ—ˆmdash;A New High-Resolution Atmosphere-Ocean Coupled General Circulation Model. *Journal of the Meteorological Society of Japan*, 90(3):325–359, 2012.

[16] Yuxia Zhang, Wieslaw Maslowski, and Albert J. Semtner. Impact of mesoscale ocean currents on sea ice in high-resolution Arctic ice and ocean simulations. *Journal of Geophysical Research*, 104(C8):18409, 1999.