Computational Performance Aspects of CROCO-BFM Coupling

Filip Strniša[a], Martin Vodopivec[b], Gregor Kosec[a]
[a] Jožef Stefan Institute, Parallel and Distributed Systems Laboratory, Ljubljana, Slovenia
[b] National Institute of Biology, Marine Biology Station, Piran, Slovenia
filip.strnisa@ijs.si, martin.vodopivec@nib.si, gregor.kosec@ijs.si

***Abstract*—Coastal and Regional Ocean COmmunity model (CROCO) is a modeling system used in oceanographic simulations. CROCO solves primitive hydrodynamic equations for momentum, heat and mass transport on a three dimensional, terrain following grid. Several biogeochemical models are already implemented in CROCO, including the PISCES model, which it borrows from NEMO, a different oceanography modeling system. While PISCES is a highly complex model, used in many oceanographic applications, it lacks variable stoichiometry and some plankton functional types needed in certain applications. On the other hand, Biogeochemical Flux Model (BFM) is a dedicated biogeochemistry modeling system, customizable and of user-defined complexity. It can be run as a standalone model or in conjunction with an oceanographic simulation. Previously we have reported on our work regarding the incorporation of BFM into CROCO. In this work we address the performance aspects of this coupling. Namely, compared to e.g. PISCES, BFM is a much more complex model with more tracers, which makes it more computationally expensive. It is therefore of the essence to investigate, and to highlight possible areas of improvement in this coupling's computational performance.**

***Keywords*—CROCO; BFM; coupling; oceanogrphy; biogeochemistry; modelling; computational performance**

I. Introduction

Coupling specialist models with each other is important, as it merges different branches of ocean science. Often circulation models are coupled with marine biogeochemical models [1]–[4]. The circulation models bring the spatial aspect into biogeochemical calculations, while on the other hand, the biogecohemical models can provide spatialy and temporally variable light absorption which influences the temperature profile in the circulation models [5]. Although some circulation models come prepackaged with some of the biogeochemical models [6], [7], comprehensive studies may sometimes require the use of a specialist model such as the Biogeochemical Flux Model (BFM) [8]. The biogeochemical models that come coupled with the oceanographic models, can serve as a template on how to implement a different biogeochemical model in the existing code.

CROCO is an oceanographic model, which incorporates momentum, mass (tracer), and heat transport [6]. CROCO code includes coupling with several biogeochemical models and the most sophisticated among these is PISCES biogeochemical model. The latter was originally coupled with NEMO oceanographic model [7]. Although PISCES is widely used in biogeochemical simulations, we have decided that our needs would be better served with BFM. The latter was chosen because it allows for variable stoichiometry (nutrient ratios in living functional groups — LFG), because the addition of new LFGs is relatively straightforward, and because the BFM is the model of choice in Copernicus Mediterranean biogeochemical forecasts and reanalyses, which will provide the boundary conditions in simulations.

In this work we present some aspects of our coupling of CROCO and BFM. First we briefly introduce both modeling environments, and proceed to highlight some key aspects of the coupling. The main focus of the paper is the assessment of the computational performance of this coupling, as compared to the original CROCO-PISCES model. We then additionally briefly explore the accuracy of a performance enhancing solution, and discuss possible areas of improvement for the current state the coupled codes.

II. CROCO

CROCO is a branch of the widely used Regional Ocean Modeling System (ROMS) [9], with an additional non-hydrostatic core and several other features including an extensive MATLAB toolbox for model configuration and results analysis [10]. Although written in Fortran, CROCO's design enables the user to toggle between different model complexities through C-preprocessor by defining or undefining its components at compilation [11]. In its essence CROCO solves primitive transport equations. Besides the oceanographic physics, it also enables the user to employ the included biogeochemical models. This is done by introducing tracer variables which represent the concentrations of biochemical components in the ocean. By enabling ocean biology, the user may further make their pick of biology-dynamics model of varying complexity (5-24 tracer components). The most sophisticated of the pre-included models is PISCES with 24 tracer variables. PISCES is borrowed from a different oceanographic model — NEMO [7].

The authors would like to acknowledge the financial support of Slovenian Research Agency (ARRS) in the framework of the research core funding No. P2-0095 and projects J7-2599, and Z7-1884.

III. BFM

BFM is an independently developed biogeochemical model [8]. It allows the user to specify its complexity, first at build time, and also at run time. The complexity involves the choice to compute pelagic, and benthic variables, to include sea ice effects, etc. By varying the complexity, the user not only chooses the amount of tracer variables that are used in computations, but also how and to what degree these variables interact with each other. When building BFM the user can pick one of the preset configurations or they can create their own configuration. In this study the "PELAGOS2" preset was used, as it is appropriate for use in coupled models.

IV. Coupling

For the purpose of this research CROCO 1.1, and BFM 5.2.0 were used. BFM was previously successfully coupled with NEMO [7]. Therefore PISCES is a common link between CROCO and BFM. PISCES' implementation in CROCO, as well as BFM's implementation in NEMO were both used as blueprints for coupling of BFM and CROCO. BFM's coupling with NEMO is achieved by almost completely rewriting NEMO's Tracers in Ocean Paradigm (TOP) component. All of this was not necessary in CROCO, as CROCO has its own subroutines for tracer transport. However, BFM's coupling with NEMO was instrumental to understanding the initiation processes, as well as main BFM tracer dynamics calls.

BFM's "PELAGOS2" preset introduces 58 tracer variables. These tracers are then initialized in CROCO's `init_arrays` subroutine. Tracers' initial values can either be obtained from input NetCDF files or from analytical initialization through `ana_initial` subroutine, both of which are taken care of by CROCO. Analytical boundary condition settings need to be manipulated in the coupled source code. The `main` program then calls a BFM subroutine `bfm_ini_tile` which initializes the appropriate BFM arrays and parameters, including reading the BFM namelist files, and initializing the parameters found therein. Also in this step, masks for 3D-to-1D and *vice versa* array transformations are built. An additional call is made to the `trc_sbc_bfm` subroutine, which reads the input files and initializes the surface boundary conditions, such as irradiation, and dust deposition. The latter subroutine has been mostly rewritten by following the preexisting PISCES' subroutine `trc_sbc_pisces`'s example. During timestepping in the `main` program, the 3D tracer arrays are updated via the `step3d_t` subroutine, which also includes a biology update (by calling `biology_tile`). Therein the main BFM call is made. CROCO's 3D tracer arrays (`t`) are transformed to BFM's 1D tracer arrays (`D3STATE`). These are then passed to BFM's main subroutines (`trc_bfm`) to calculate their respective mass fluxes. BFM's tracer arrays are then updated with these mass fluxes through Euler's method. Finally the values from BFM's diagnostic arrays, as well as tracer arrays, are copied back to CROCO's arrays. The output is done entirely through CROCO's subroutines, as BFM's output was disabled, with the exception of a BFM's separate log file. Some of the key communications between different subroutines of the two codes, are laid out in Fig. 1.

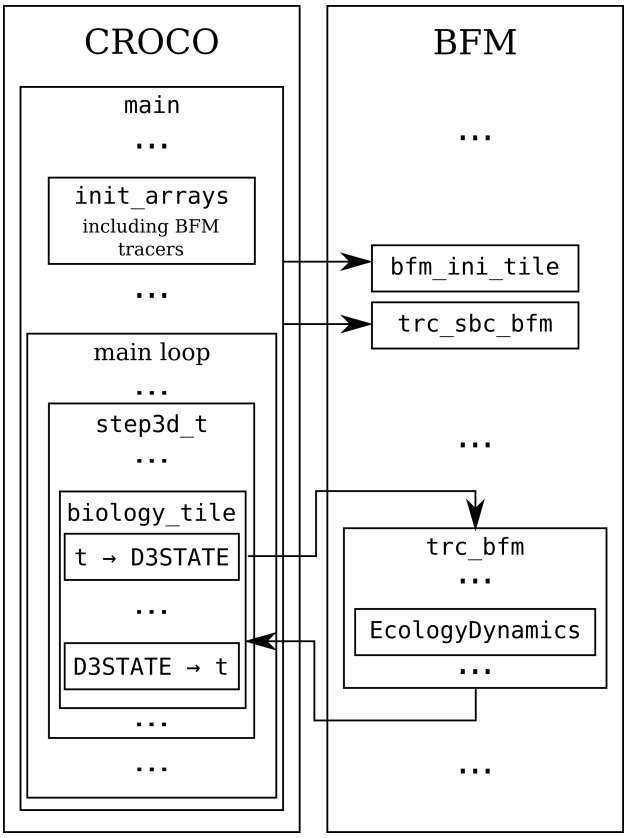

Figure 1. A schematic illustrating some of the key communications made between the two codes.

V. Test case

The CROCO-BFM coupled model was verified by comparing it to the coupling of MITgcm and BFM [2]. A test case was designed where a wind-induced gyre is formed within a closed square oceanic pool: 41.0°N – 43.0°N, -20.5°E – -17.5°E, with even 280 m depth. The grid dimensions were $95 \times 86 \times 32$. Model verification is not the purpose of this study, and these results were presented previously [12]. While conducting these tests, however, very slow performance of the coupled model was observed.

Consequently, to mitigate the slowdown by BFM, the biology call can be made only every n iterations, an approach also explored in [2]. The biological processes are expected to be slower than tracer physics which justifies the longer biology time step. Thus the test case is set up to simulate roughly 14 days of oceanic circulation (2000 time steps of 600 s) and biology in separate runs with PISCES, and BFM. In the PISCES run the biology call is made on every update, whereas in BFM runs it is made every n updates, $n \in [1, 8] \cap \mathbb{N}$. The PISCES run serves merely as an example of what "optimal" computation times could be, and BFM runs are compared to the $n = 1$ run to check for possible accuracy

concerns when decreasing the biology update frequency. To measure the accuracy of $n \in [2, 8] \cap \mathbb{N}$ solutions the normalized ℓ^2-norm is calculated as follows:

$$\ell^2\text{−norm}(c_n) = \frac{1}{\max(c_1)} \sqrt{\frac{\sum_i^N (c_{1,i} - c_{n,i})^2}{N}}, \quad n \in [2, 8] \cap \mathbb{N}, \tag{1}$$

where $c_{n,i}$ represents the tracer concentration for solution n at i-th node, and N is the total number of discrete grid nodes.

VI. Results and Discussion

The results presented in this section were obtained by running the test case on a laptop PC equipped with AMD Ryzen 7 6800HS CPU, and 40 GB DDR5 RAM. Although the computational performance of the code can be significantly improved by running it in parallel mode (MPI), these tests were performed in single-core mode to better highlight the execution time ratios of the biology vs tracer physics step. Some aspects of the speed-up achieved through parallel execution were presented in [12].

First we present a profile break-down of the `step3d_t` subroutine, which lays out the total time spent at a certain part of the aforementioned subroutine. Results are given in Fig. 2, and Table I.

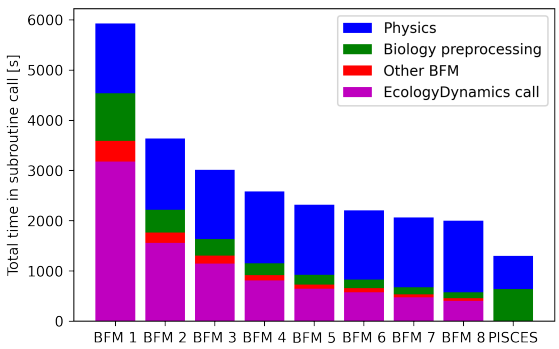

Figure 2. Total time spent on different subroutines. "Physics" refers to the `step3d_t` subroutine, excluding the biology update; "Biology preprocessing" refers to the transformation of `t` to `D3STATE`, and vice versa, and in PISCES's case the full biology update; "Other BFM" are diverse BFM calls; "EcologyDynamics call" is the main BFM call to `EcologyDynamics`. The number following "BFM" is indicating the period of biology update in time steps, n.

The slight deviation in average times between the cases, presented in Table I, are due to the PC not having locked frequencies, and the computations were being carried out, while the PC was used for other tasks as well. However, for the purpose of this study, the quality of these data should be sufficient, as expected trends are shown. Some performance hits of BFM compared to PISCES can be expected as the number of tracers used in the simulation is more than doubled (58 as opposed to 24). Consequently the average execution time of the tracer physics part of `step3d_t` subroutine increases from 0.33 s to 0.70 s. Therefore, equal execution time for BFM and PISCES cannot be expected. However, the average `biology_tile` call goes from 0.32 s for PISCES to 2.34 s for BFM, which is about 7-times slower. With BFM being a more complex model, some degree of slowdown is expected here as well, but this result seems excessive. The transformation between BFM's 1D and CROCO's 3D arrays appears to be the weak point of our coupling's current implementation, as it takes about 20% of `biology_tile` call time on average, and more work should be dedicated to alleviate this issue. Fortran supports vector remapping, which could be a solution here. In this case `t` (of CROCO) should be declared as target, and `D3STATE` (of BFM) as pointer. However, the issue is that `t` is structured as `(positional id, species)`, and `D3STATE` as `(species, positional id)`, which prevents the use of the proposed solution. As Fortran arrays are stored in memory in column-major order, this means that values of `t` for the same tracer species at different coordinates are stored close to each other. This is optimal for computing mass transport — the main goal of CROCO. In `D3STATE` different species at the same coordinate are stored close, which is optimal for computing interplay of these species — the main goal of BFM. Thus to solve the issue of restructuring the basic tracers' data structures on every biology call, the BFM code would need to be rewritten. All instances of `D3STATE`, and other arrays that it interacts with, would need to be changed to `(positional id, species)`, and this may bring upon a performance hit in the `EcologyDynamics` call. By increasing n (the period of biology update in time steps) the total time spent computing biology is brought down significantly, which is shown in Fig. 2. If one's aim was to make the time computing biology about equal to the time computing tracer physics (as is the case in PISCES), n should be between 3, and 4. At $n = 8$ the total time spent computing biology is even lower for BFM than for PISCES, which is expected, as, as it was mentioned above, BFM is about 7-times slower than PISCES. How the value of n affects the results accuracy is presented in Fig. 3.

Some data in Fig. 3 appear distorted, because the output was done every 10 time steps. Consequently, a tracer's concentration may not have been updated at a certain output and the deviation from $n = 1$ solution may be greater than at an output when the concentration was updated. Thus $n = 2$, and $n = 5$ lines appear smooth, as 2 and 5 are divisors of 10. Some tracers are more affected by increasing n than others. Out of 58 those, which suffer from accuracy issues the most, are presented in Fig. 3. These are the concetrations of: N1p - phosphate; N4n - ammonium; P1c - carbon in diatoms; P3c - carbon in picophytoplakton; R1c - carbon in labile dissolved organic matter; R6c - carbon in particulate organic matter; Z4c - carbon in omnivorous mesozooplankton; Z5c - carbon in microzooplankton. As one would expect the deviation from $n = 1$ solution increases with increasing n, which effectively increases the time interval in Euler's

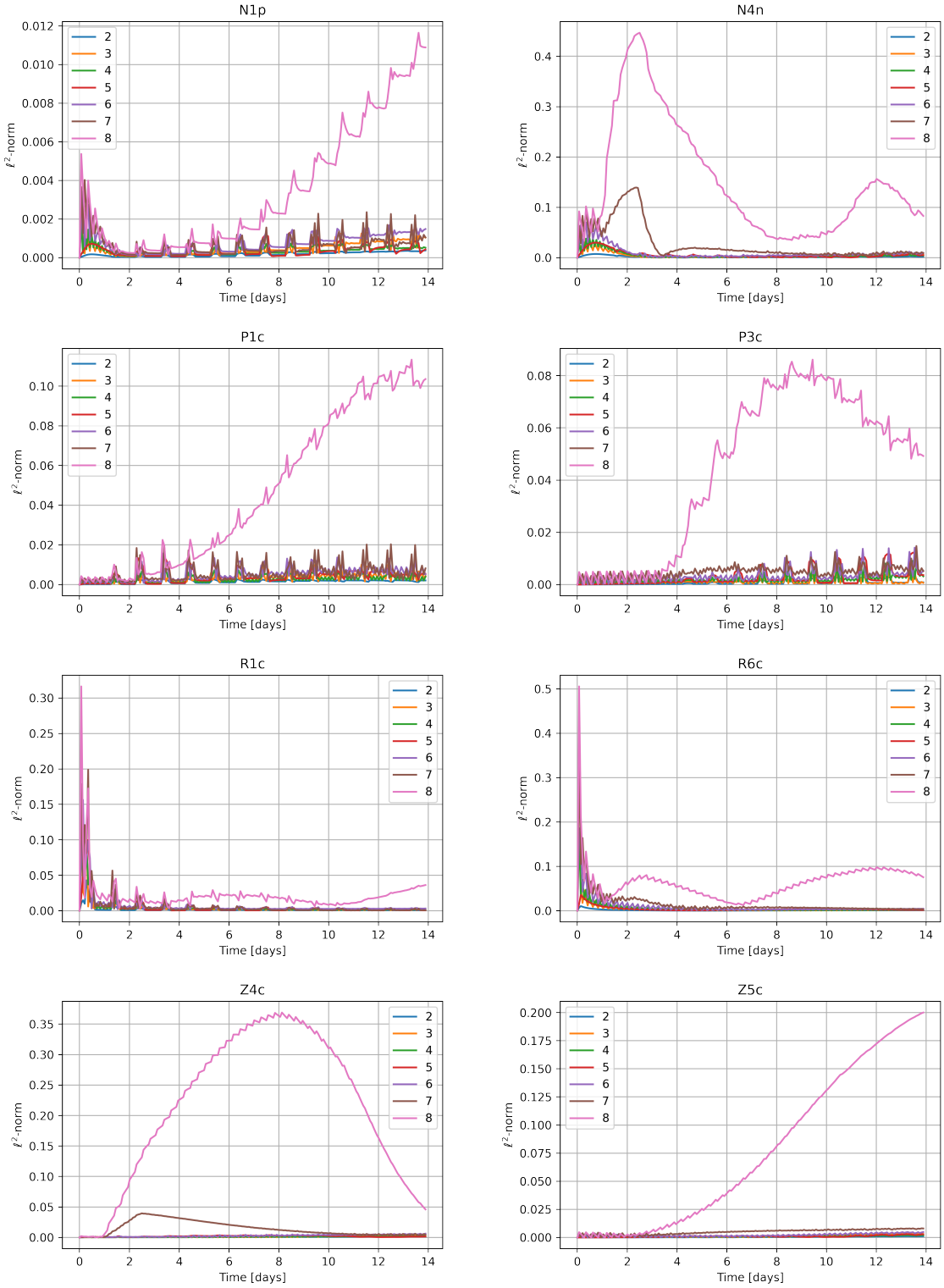

Figure 3. The discrepancy of tracer concentrations for different biology update frequencies. A case where biology is solved on every time step ($n = 1$) is the base result, to which we compare the dimensionless ℓ^2-norm of other cases. Displayed are only the tracers where the ℓ^2-norm exceeds 0.01. The labels represent the concentrations of the following species: N1p - phosphate; N4n - ammonium; P1c - diatoms; P3c - picophytoplakton; R1c - labile dissolved organic matter; R6c - particulate organic matter; Z4c - omnivorous mesozooplankton; Z5c - microzooplankton. The label numbers represent the biology update period in expressed in time steps (n).

TABLE I. AVERAGE AND TOTAL TIME SPENT COMPUTING TRACER PHYSICS AND BIOLOGY IN STEP3D_T.

Case	Average tracer physics time [s]	Average biology time [s]	Total tracer physics time [s]	Total biology time [s]
BFM ($n = 1$)	0.69	2.27	1388	4538
BFM ($n = 2$)	0.71	2.22	1418	2220
BFM ($n = 3$)	0.69	2.46	1373	1639
BFM ($n = 4$)	0.71	2.31	1430	1156
BFM ($n = 5$)	0.70	2.30	1398	923
BFM ($n = 6$)	0.69	2.48	1378	829
BFM ($n = 7$)	0.70	2.36	1392	676
BFM ($n = 8$)	0.71	2.30	1421	578
PISCES	0.33	0.32	661	641

integration. However, these deviations can be over 40% (N4n, R6c, when $n = 8$). Although the computational performance is increased, there can be a significant impact on accuracy, and in going forward a balance between the two should be found. Implementing a different time-integration scheme, such as 2nd order Euler method, or even 4th order Runge-Kutta may improve the integration accuracy, but would inherently increase the amount of `EcologyDynmaics` calls in a given time step, effectively further slowing down the computation. Specifically for the presented case, $n = 3$ or $n = 4$ seem like a good compromise, as at these n the total biology time is about the same, as total tracer physics time, and the deviation from the $n = 1$ solution is not so big. However, one should note that this computations were done in the period of vigorous model dynamics due to the adaptation to the initial conditions. After the initial growth, many discrepancies start to fall as the model reaches a more stable state. It would perhaps be interesting to explore the possibility of an adaptive biology update period, as in oceanographic computations not all of the nodes are equally biologically active.

VII. CONCLUSION

Performance aspects of the CROCO-BFM coupling were presented, i.e. the computational performance, a possible measure to improve the latter, and a brief numerical accuracy analysis of these measures. We have considered a small-scale closed basin as a test case, solving it for approximately 14 days of simulated time. The coupled model performs significantly slower than the reference model, which is CROCO with PISCES biology. One reason behind slower performance is BFM simulating 58 tracers vs PISCES' 24. This results in longer computation times for both tracer physics, and biology. However, while the tracer physics computation time merely doubles, the biology computation time increases sevenfold. Some of these issues come from having to transform between CROCO's, and BFM's data structures in their communication, but there is no simple solution to this problem. The issue of poor performance can be solved by performing the biology update at a lower frequency, i.e every $n-\text{th}$ step. In doing so, one can achieve satisfactory performance results, but there is noticeable penalty to the accuracy. Our tests show that $n = 3$ or $n = 4$ produce good results in test case conditions, while keeping the slowdown at a reasonable amount. The effects of the biology call on reduction of accuracy in large-scale simulations, as well as for longer time periods, have not been explored in this study.

REFERENCES

[1] J. S. Lessels, D. Tetzlaff, S. K. Carey, P. Smith, and C. Soulsby, "A coupled hydrology–biogeochemistry model to simulate dissolved organic carbon exports from a permafrost-influenced catchment," *Hydrological Processes*, vol. 29, no. 26, pp. 5383–5396, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.10566

[2] G. Cossarini, S. Querin, C. Solidoro, G. Sannino, P. Lazzari, V. Di Biagio, and G. Bolzon, "Development of bfmcoupler (v1.0), the coupling scheme that links the mitgcm and bfm models for ocean biogeochemistry simulations," *Geoscientific Model Development*, vol. 10, no. 4, pp. 1423–1445, 2017. [Online]. Available: https://gmd.copernicus.org/articles/10/1423/2017/

[3] R.-H. Zhang, F. Tian, and X. Wang, "A new hybrid coupled model of atmosphere, ocean physics, and ocean biogeochemistry to represent biogeophysical feedback effects in the tropical pacific," *Journal of Advances in Modeling Earth Systems*, vol. 10, no. 8, pp. 1901–1923, 2018. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2017MS001250

[4] H.-C. Jung, B.-K. Moon, H. Lee, J.-H. Choi, H.-K. Kim, J.-Y. Park, Y.-H. Byun, Y.-J. Lim, and J. Lee, "Development and assessment of nemo(v3.6)-topaz(v2), a coupled global ocean biogeochemistry model," *Asia-Pacific Journal of Atmospheric Sciences*, vol. 56, no. 3, pp. 411–428, Aug 2020. [Online]. Available: https://doi.org/10.1007/s13143-019-00147-4

[5] K. Fennel, J. P. Mattern, S. C. Doney, L. Bopp, A. M. Moore, B. Wang, and L. Yu, "Ocean biogeochemical modelling," *Nature Reviews Methods Primers*, vol. 2, no. 1, p. 76, 2022.

[6] "Croco – coastal and regional ocean community model," 10.5281/zenodo.7415055, accessed: 16 Jan 2023.

[7] "Nemo," nemo-ocean.eu, accessed: 17 Feb 2023.

[8] "Biogeochemical flux model — bfm," bfm-community.eu, accessed: 16 Jan 2023.

[9] "Roms," myroms.org, accessed: 17 Feb 2023.

[10] "Croco tools," 10.5281/zenodo.7431960, accessed: 16 Jan 2023.

[11] "Croco – coastal and regional ocean community model documentation," 10.5281/zenodo.7400758, accessed: 16 Jan 2023.

[12] M. Vodopivec, F. Strniša, and G. Kosec, "Coupling the coastal and regional ocean community model (croco) with the biogeochemical flux model (bfm)," in *EGU General Assembly 2022, Vienna, Austria*, 23–27 May 2022.