

# The Technology of Nesting A Regional Ocean Model into A Global One Using A Computational Platform for Massively Parallel Computers CMF

Alexandr Koromyslov<sup>1,2</sup> (✉), Rashit Ibrayev<sup>3,2,4</sup>, Maxim Kaurkin<sup>4,2</sup>

<sup>1</sup>M.V. Lomonosov Moscow State University, Moscow, Russia

<sup>2</sup>Hydrometeorological Centre of Russia, Moscow, Russia

<sup>3</sup>Institute of Numerical Mathematics RAS, Moscow, Russia

<sup>4</sup>P.P. Shirshov Institute of Oceanology RAS, Moscow, Russia

alexandr.koromyslov@gmail.com

ibrayev@mail.ru

kaurkinmn@gmail.com

**Abstract.** When developing regional ocean circulation model, the problem arises of providing the model with boundary conditions. An algorithm for one-way nesting (inclusion with boundary conditions) for a local model of an arbitrary ocean region in the model of the global ocean is proposed. Two problems are solved: 1) generation of a rectangular grid of the local model; 2) receive information. The nesting algorithm is developed within the framework of the CMF3.0 (Compact Modeling Framework) computing platform for massively parallel computers. Local and global models work as components of a coupled system running CMF3.0. Data nesting functions work as a CMF3.0 software service.

**Keywords:** Earth System Modeling · Ocean Modeling · Nesting · Coupler.

## 1 Introduction

When modeling the World Ocean, it sometimes becomes necessary to obtain an accurate prediction only for a particular region, for example, a certain sea. At the same time, numerical integration of the entire high-resolution World Ocean dynamics model ( $0.1^\circ$  and more) is a computationally expensive task that sometimes requires several thousand processor cores [11]. To optimize the computational costs, it is reasonable to model the area under study using a high spatial-temporal resolution model (hereinafter, the local model), and to provide the model with the boundary and initial conditions for modeling the rest of the ocean using the World Ocean model of a more coarse resolution (global model). An important point that has a significant impact on the quality of the results is the way in which the boundary conditions are set in the local model. In real applications, several approaches are currently used. One of these is nesting, a computational approach in which data is exchanged between global and local models. In this case, the global model usually has low resolu-

tion (for example,  $1^\circ$ ), and local - high or ultra-high resolution ( $0.1^\circ$  or more). This formulation of the problem requires considerable computational resources, since the global and local models sometimes use several hundred computing cores. But at the same time, this approach is more optimal from the point of view of computational costs than the modeling of the entire World Ocean with a high resolution, while the quality of the model forecast in the region under study with a correct formulation of the boundary conditions remains high.

The relevance of using the nesting method has previously been examined with examples of climate modeling in winter in South America [8], modeling the northwestern Caribbean and the Scottish shelf in the northwestern Atlantic Ocean [6]. There is also an active research in the field of nesting algorithms [6], [7], [9].

The purpose of this work is to implement the nesting algorithm as a software service NST (abbr.NeSTing) of the CMF 3.0 computing platform for use in high spatial resolution models on massively parallel computers with distributed memory. This work continues a series of studies of authors published earlier in [2], [10] and devoted to the development of tools for modeling the Earth system on computers with distributed memory based on the compact modeling framework CMF 3.0.

## 2 Compact Modeling Framework CMF 3.0 and NST Service.

Despite the fact that the logic of the operation of the coupler interpolation procedures in CMF 3.0 remained the same as in CMF 2.0 [3], the PGAS abstraction made it possible to greatly simplify the code. Now all the data necessary for the process from neighbors is obtained using the Communicator class.

The disadvantage of this approach is the performance degradation associated with the inability to use deferred MPI operations and the availability of GA own costs. However, the results of CMF 2.0 showed that we can sacrifice some of the performance to select a simpler (and possibly less efficient) abstraction to simplify communication algorithms [2]. Because, first, although the pure MPI-approach to communications has a high speed, it requires explicit work with data buffers, which greatly complicates the development and improvement of PCM. Secondly, the development of regional sub-models of the seas embedded in the grid of the global model, became rather complicated when using only MPI-procedures. On the other hand, this solution has made it possible to simplify the connection of new models, as well as to expand the functionality of the joint simulation system by adding new software services.

### 2.1 PGAS-Communicator

CMF3.0 contains the Communicator class, which encapsulates the logic of working with the Global Arrays library (GA) [8] and provides only an interface for put / get operations of parts of global data of model components. The GA library implements the PGAS (Partitioned Global Address Space) paradigm and allows working with distributed memory as a shared memory.

All exchanges between parts of the system are implemented using the Communicator class. It contains a hash table for storing all information about arrays, including their state and metadata. Each array of the component involved in exchanges contains a distributed copy, stored as a virtual global array GA. When a process needs to send data, it fills this copy with its current data. Due to the fact that the distribution of the global array completely repeats the decomposition of the component, this operation takes place locally.

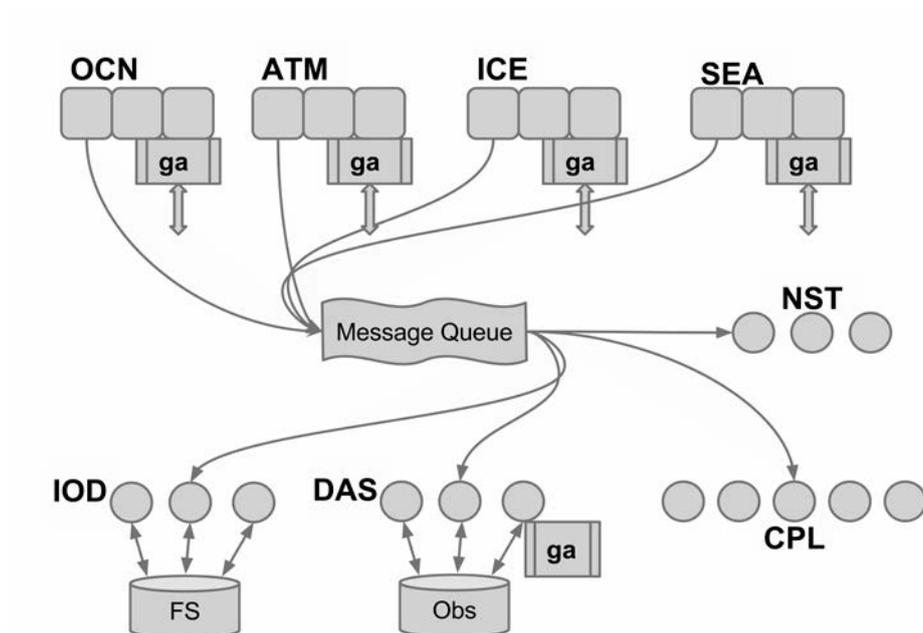
## 2.2 The Architecture of The Coupling Model

Due to the growing complexity of the coupling system, a more convenient way of combining components - SOA (Service-Oriented Architecture) was used.

In CMF 3.0, all models send their general requests to a single message queue (Fig. 1). Service components take from this queue only messages that can process, take data from virtual global arrays, and perform appropriate actions. The architecture allows minimizing the connections between physical and service components and greatly simplifying the development. Moreover, since all services inherit the common base class Service, adding a new service is not difficult. Now CMF 3.0 contains the following independent parallel services: CPL (mapping operations), IOS / IOF (I / O Fast, I / O Slow - fast and slow file devices, DAS (Data Assimilation System) [10].

Service CPL is a CMF 2.0 coupler, which now deals exclusively with operations related to interpolation. It receives data using Communicator, performs interpolation, and places the data in the virtual global array of the recipient.

In CMF 3.0, a separate I / O service responsible for all I / O operations was allocated. It is interesting that one service solves only half of the problem, because, for example, a simultaneous request to write a large amount of information (a system check point) and a small model diagnostics will also take place sequentially. Therefore, the service was divided into a fast and slow recording device (thanks to the abstraction this division is performed by several lines of code). This mechanism provides a flexible and asynchronous mechanism for working with the file system.



**Fig. 1.** The architecture of the compact framework CMF 3.0. There are four components in this example: ocean model (OCN), ice model (ICE), atmosphere model (ATM) and sea model (SEA). They send requests to the common message queue, where they are retrieved by coupler (CPL), data assimilation (DAS), input and output data (IOD), nesting (NST) services. The data itself is transferred through the mechanism of global arrays, which are also used for interprocessor communication in the components and services

### 3 Description of The NST Service

#### 3.1 Mathematical Formulation One-Way Nesting Problem

We are interested in representing as accurately as possible the local ocean model in a domain  $\Omega_{loc}$ . The circulation is supposed to be described on a time period  $[0, T]$  by a model which can be written symbolically

$$L_{loc}u_{loc} = f_{loc} \text{ in } \Omega_{loc} \times [0, T] \quad (1)$$

with convenient initial conditions at  $t = 0$ .  $L_{loc}$  is a partial differential operator,  $u_{loc}$  is the state variable, and  $f_{loc}$  the model forcing. The conditions at the solid boundaries will never be mentioned in this note, since they do not interfere with our subject.

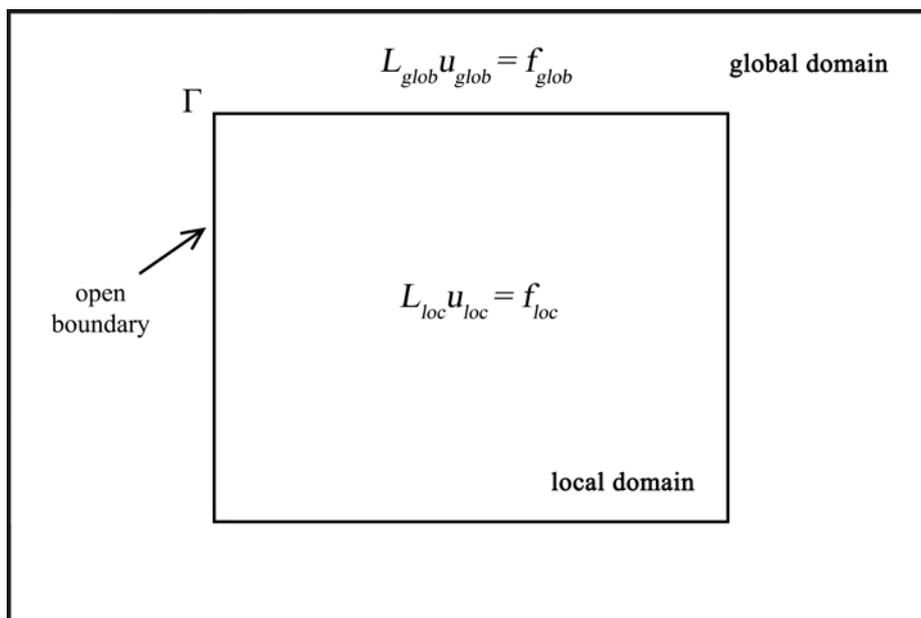
Since  $\Omega_{loc}$  is not closed, a portion of its boundary does not correspond to a solid wall, and has no physical reality. This artificial interface, also called open boundary (OB), is denoted  $\Gamma$ . The local solution  $u_{loc}$  is thus in interaction with the external

ocean through  $\Gamma$ , and the difficulty consists in adequately representing this interaction in order to get a good approximation of  $u_{loc}$  in  $\Omega_{loc} \times [0, T]$ .

We also assume that we have at our disposal a (probably less accurate) representation of the global ocean, either under the form of some data  $u_{glob}$  or of a global model

$$L_{glob}u_{glob} = f_{glob} \quad \text{in } \Omega_{glob} \times [0, T] \quad (2)$$

where  $\Omega_{glob}$  is a global oceanic domain. Note that, in our notations,  $\Omega_{loc}$  and  $\Omega_{glob}$  do not overlap (Fig. 2).



**Fig. 2.** A schematic view of nesting problem

The solution  $U_{glob}$  of an external model cover an area  $\Omega_{glob} \cup \Omega_{loc}$  larger than  $\Omega_{loc}$  is available. Therefore tis larger scale solution can be used to force the local model along  $\Gamma$ . The formulation of the problem which is solved in that approach is (3).

$$L_{glob}u_{glob} = f_{glob} \quad \text{in } \Omega_{glob} \cup \Omega_{loc} \times [0, T]$$

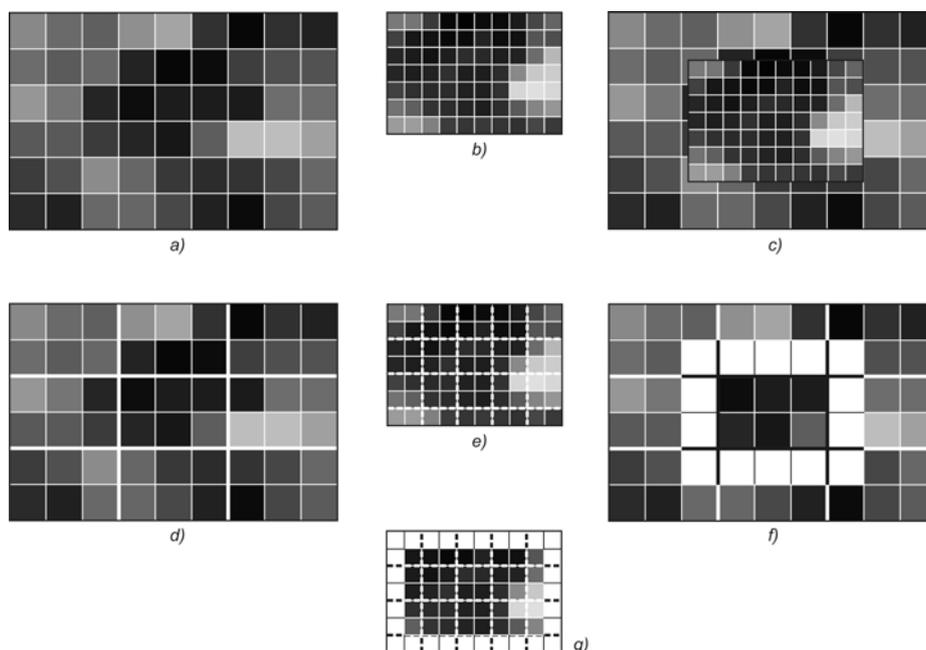
$$\begin{cases} L_{loc}u_{loc} = f_{loc} & \text{in } \Omega_{loc} \times [0, T] \\ Bu_{loc} = Bu_{glob} & \text{on } \Gamma \times [0, T] \end{cases} \quad (3)$$

where B denotes an open boundary operator (in particular cases are Dirichlet and Neumann conditions).

This interaction between the two models can be performed on-line using features CMF 3.0.

### 3.2 Description of The Algorithm

Suppose we have a global model with a rough spatial resolution (Fig. 3a), a local high-resolution model (Fig. 3b). The mutual arrangement of the local and global models is as follows (Fig. 3c). Let the decomposition between the processors local and global models differ as well (Fig. 3d - 3e). The global model needs to send the boundary cells of the area that covers the local model (Fig. 3f) to the nesting service. The nesting service interpolates the data received from the global model on the grid of the local model and then sends it to the boundary cells of the local model (Fig. 3g).



**Fig. 3.** Graphical description of the nesting algorithm: (a) global model grid; (b) local model grid; (c) the combination of the calculated grids of the global and local models; (d) processor decomposition of the global model area (shown in white bars); (e) processor decomposition of the region of the local model (shown in white bars); (f) white cells are highlighted in the global grid required for the nesting; (g) the cells of the local grid are highlighted in white, in which the boundary conditions are set by the nesting

### 3.3 Features of Parallel Implementation of The Service

Like any service of the CMF3.0 software complex, the nesting service is performed on separate computing cores. This approach allows us to obtain a transparent structure of the joint model, where each component is engaged in the solution of its task. At the same time, neither the global nor the local models practically participate in the nesting, only mesh and model masks are used, on the basis of which the nesting service builds the matrix of interpolation scales. Moreover, in the NST service, in addition to

specifying the classical Neumann or Dirichlet conditions, the logic can be encapsulated to generate conditions for the sum of the flows (heat, salt, etc.) to be zero across the boundaries of the region under investigation or their correspondence to certain integral characteristics. The data from the global to the local model go without access to the file system, using the cluster interconnect through the GA library, which positively affects the performance of the system as a whole.

Problems and reasons for making a nesting in a separate service:

1. Global and local models have different resolutions and are calculated on a different number of cores.
2. The NST service can be used for other Earth system models (for example, the atmosphere model), which is also successfully used in conjunction with the CMF complex [12].
3. Interpolation matrices occupy a significant amount of memory, and the logic of calculating them can be very difficult, especially in the case of setting complex boundary conditions. Therefore, it is more advantageous to store them on separate computational cores from model components.

General algorithm of the local and global model using the NST service

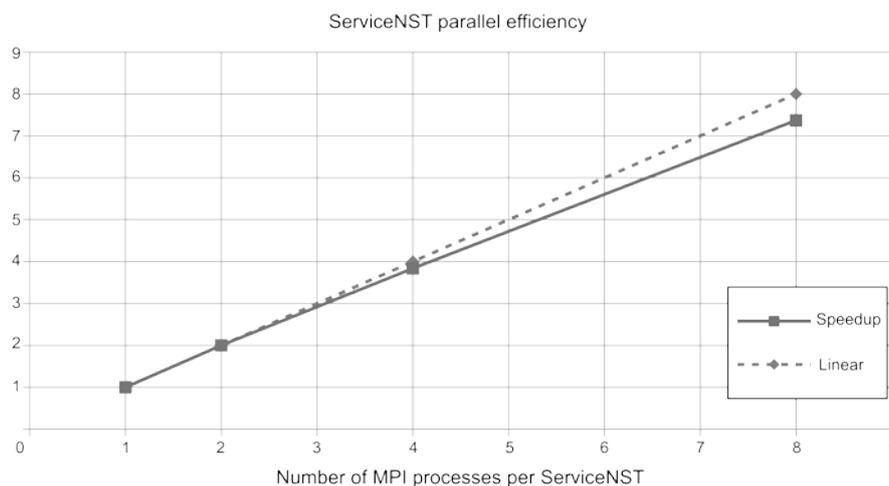
- The initial values in the global and local models are initialized
- The NST service receives from the global and local models the configuration of grids, masks, dimensionality of models, decomposition of processors for each model, steps in time.
- Based on the received data, the NST service calculates the position of the local model relative to the global model.
- Based on the position of the local model, the NST service computes the elements of the arrays that need to be transferred from the global model to the local model.
- Based on the given boundary conditions for the local model and the obtained information about the model masks and grids, the matrices of the interpolation weights are calculated.
- When the transmission time is reached, the global model sends only the necessary parts of the model arrays to the NST service.
- The NST service interpolates the data from the global to the local grid using the previously obtained matrix of weights.
- The NST service sends the resulting boundary conditions to the local model.

### 3.4 Testing The NST Service

To test the computational efficiency of the parallel nesting algorithm using the GA software library, a global test model with a resolution of  $0.25^\circ$  ( $1440 \times 720$  grid) and a local model with a resolution of  $0.1^\circ$  ( $400 \times 400$  grid) were specified, in both 49 vertical z-levels. The global model used two-dimensional decomposition of the computational domain into 64 processor cores, and the local model used 32 cores, and the NST service was allocated from 1 to 8 cores. From the global to the local model,

boundary conditions of the Dirichlet type were transmitted for model fields of temperature, salinity, and velocity [7]. The matrix of the interpolation weights was constructed by the method of bilinear interpolation, taking into account the land mask for the global and local models. The scalability of the parallel Nesting algorithm within the NST service is shown in the calculation of the boundary condition for 4 (the number of model fields) \* 1600 (the extent of the local domain boundary) \* 49 (the number of z-levels) ~ O(10<sup>5</sup>) points in the local model. When constructing the parallel efficiency graph, the sum of the time spent on transferring data from the global model to the NST service, the time of parallel calculation of the boundary conditions by the NST service cores, and the time of data transfer to the local model (4).

$$T_{\text{nesting}} = T_{\text{ocn\_to\_nst}} + T_{\text{nst}} + T_{\text{nst\_to\_sea}} \quad (4)$$

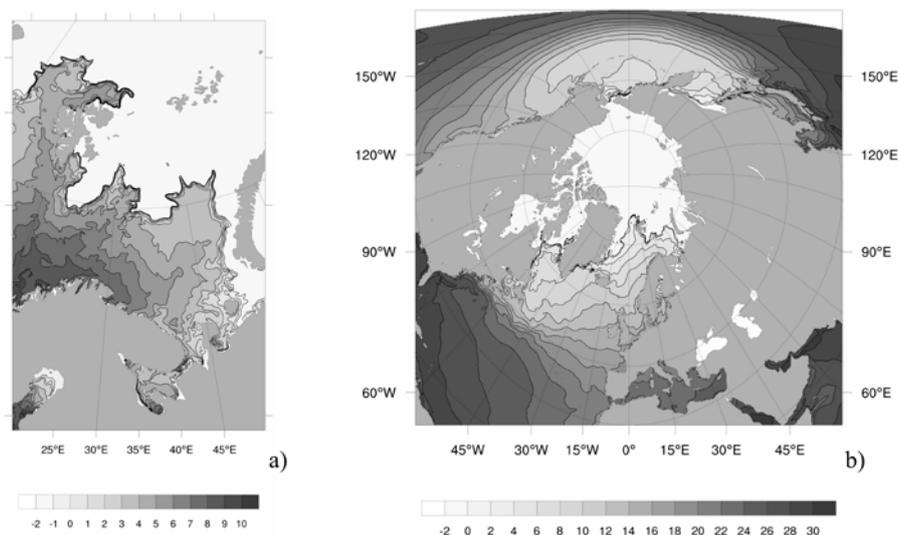


**Fig. 4.** Scalability of the parallel nesting method within the NST service for obtaining boundary conditions for O(10<sup>5</sup>) points on the MVS-10P supercomputer. On the X axis, the number of processor cores per NST service. On the Y-axis, the parallel nesting method is speeded up based on T<sub>nesting</sub> time measurements. The global test model used a two-dimensional decomposition of the computational domain into 64 processor cores, and the local one - to 32 cores

#### 4 Example of Use

(Fig. 5) shows the temperature field in the test region for the local model-the Barents Sea with a resolution of 0.1° (Fig. 5a) and a global model of the World Ocean with a resolution of 0.5° (Fig. 5b). The size of the spatial grids for the World Ocean and the Sea is 720 × 360 and 300 × 220, in each 49 levels vertically. The model of the ocean uses 64 processor cores in calculations, and the Barents Sea model has 12 cores, 2

processor cores are allocated to the NST service for nesting boundary conditions.



**Fig. 5.** Use of nesting technology for modeling the dynamics of the Barents Sea. (a) the surface temperature field in the Barents Sea model with a spatial resolution of 0.1°; (b) the surface temperature field in the World Ocean model with a resolution of 0.5° for specifying the boundary conditions in the sea model. The Barents Sea model uses a two-dimensional processor decomposition of the calculated region into 12 processor cores, and a global one - into 64 cores

## 5 Conclusion

In this paper, a parallel implementation of nesting technology as a software service NST of the compact computing platform CMF3.0 is presented [2]. A parallel nesting algorithm is described in detail, an example of using the developed technology is given.

For the first time in Russia, a parallel technology for transferring boundary conditions (nesting) through a cluster interconnect (without the use of a file system) between ocean and sea dynamics models using CMF 3.0 was implemented. This implementation allowed the ocean and sea models to function as a single coupling model on massively parallel computers with distributed memory. The advantages of this approach include computing efficiency and speed, which will allow further use of this technology for operational forecast of the state of the ocean.

Nesting technology is implemented in other most advanced frameworks for Earth system modeling - CESM [13] and OASIS [14]. The nesting service (NST service) based on CMF 3.0, which is described in this article demonstrates similar results of computing efficiency.

## 6 Acknowledgements

To test the performance of the nesting service and the compact computing platform CMF 3.0, tests were conducted on the supercomputers Lomonosov and MVS-10P, which confirmed the numerical efficiency of the proposed software product.

The study was performed by a Grant #14-37-00053 from the Russian Science Foundation in Hydrometcentre of Russia.

## References

1. Ibrayev, R.A., Khabeev, R.N., Ushakov, K.V.: Eddy-resolving  $1/10^\circ$  Model of the World Ocean. *Izvestiya. Atmospheric and Oceanic Physics* 48(1), 37-46 (2012)
2. Kalmykov V., Ibrayev R. CMF - framework for high-resolution Earth system modeling. *CEUR Workshop Proceedings*. V. 1482. Proceedings of the 1st Russian Conference on Supercomputing (RuSCDays 2015). Moscow, Russia. September 28-29, 2015. Moscow, MSU publishers. 2015. P. 34-40
3. Kalmykov, V.V., Ibrayev, R.A.: A framework for the ocean-ice-atmosphere-land coupled modeling on massively-parallel architectures. *Numerical Methods and Programming* 14(2), 88-95 (2013)
4. Unidata, (2015): Network Common Data Form (netCDF) version 4.3.3.1 (software). Boulder, CO: UCAR/Unidata. (<http://doi.org/10.5065/D6H70CW6>).
5. Craig A, Jacob R, Kauffman B. CPL6: The new extensible, high performance parallel coupler for the Community Climate System Model // *IJHPCA*. 2005. Vol. 19. P. 309-327.
6. Jinyu Sheng, Richard J. Greatbatch, Xiaoming Zhai, Liqun Tang. A new two-way nesting technique for ocean modeling based on the smoothed semi-prognostic method // *Ocean Dynamics*. Springer (2005) 55: P. 162–177. DOI 10.1007/s10236-005-0005-6
7. E. P. Chassignet and J. Verron (eds.). Nesting ocean models // *Ocean Weather Forecasting*. Springer (2006), P.127-146
8. C.G. Menéndez, A.C. Saulo, Z.-X. Li. Simulation of South American windertime climate with a nesting system // *Climate Dynamics*. Springer (2001) 17: P. 219-231
9. Markus Scheinert, Arne Biastoch, Claus W. Böning. The Agulhas system as a prime example for the use of nesting capabilities in ocean modeling // *High Performance Computing on Vector Systems* (2009), P. 191-198. DOI 10.1007/978-3-642-03913-3\_15
10. Kaurkin M., Ibrayev R., Koromyslov A. EnOI-Based Data Assimilation Technology for Satellite Observations and ARGO Float Measurements in a High Resolution Global Ocean Model Using the CMF Platform. *Supercomputing*. Eds. Voevodin V., Sobolev S. Series: Communications in Computer and Information Science. Springer. 2016. V. 687. P. 57-66.
11. Tolstykh A., Ibrayev R. Models of the global atmosphere and the World Ocean: algorithms and supercomputer technologies. Tutorial. Moscow: Publishing House of Moscow University, 2013. 144 p. (in Russian)
12. Fadeev R., Ushakov K., Kalmykov V., et al. Coupled atmosphere–ocean model SLAV–INMIO: implementation and first results // *Russian Journal of Numerical Analysis and Mathematical Modelling*. 2016. V. 31(6). P. 329-337.
13. Craig, A. P., M. Vertenstein, and R. Jacob, 2012: A New Flexible Coupler for Earth System Modeling developed for CCSM4 and CESM1. *Int. J. High Perform. C*, 26-1, 31–42
14. Redler, R., S. Valcke, and H. Ritzdorf, 2010: OASIS4 - a coupling software for next generation earth system modeling. *Geosci. Model Dev.*, 3, 87–104