

# Overview of SCM Coupler and Its Application for Constructing Climate Models

Viacheslav S. Gradov<sup>1</sup> , Gennady A. Platov<sup>1</sup> 

© The Authors 2023. This paper is published with open access at SuperFri.org

Numerical modeling is one of the leading research tools for the climate problems. Among numerical models, researchers use, in particular, coupled models, that are numerical models describing more than one climate component dynamics and their interactions. The simulation results with such models depend on the way how these interactions are configured. Therefore, a proper configuration of the exchanges is crucial. The software called a “coupler” is often used to configure these interactions. The coupler is most helpful if the model components are independent of each other modules, their number exceeds two, and they have their own computational grid and time integration step. Key functions of the coupler are managing data exchange between models, setting up synchronous interaction between them based on the time integration step, and interpolating data from one model’s computational grid to the other model’s grid. Additional functions can also be implemented, e.g., fluxes computation between model components, data assimilation, working with the file system, etc. The coupler has one crucial feature: if there is a set of different models of climate system components, one can construct new coupled models by coupling various subsets of these components with coupler. This paper gives an overview of the SCM (SibCIOM Coupling Module) coupler we first developed for the model SibCIOM (Siberian Coupled Ice and Ocean Model). The description of this coupler has not been published before. The SCM coupler is a separate module to which the main climate system model component, such as atmospheric, oceanic, sea ice and land components, can be attached. Additional functions of this coupler include computation of atmosphere-to-ocean and atmosphere-to-ice fluxes and ocean and sea ice state correction using a tidal model. This paper also gives examples of two models constructed with the SCM coupler.

*Keywords: coupler, climate modeling, numerical modeling, coupled models.*

## Introduction

Numerical modeling is one of the main tools in climate research nowadays. There are many numerical models of different complexity, from simple null or one-dimensional models to ensembles of coupled three-dimensional models. Every model type is used for various aims depending on the investigated problem. For example, simpler models can be used to estimate the influence of a specific physical process on the climate, to analyze particular climate drivers or to estimate some parameters needed for the more complex models to work correctly [9, 19, 25]. More complex models are often used to describe the dynamics and interactions of climate system components in detail [33]. Coupled models also describe some climate system components, such as coupled ocean and sea ice models, e.g., FEMA0 [30], SibCIOM [14, 15], coupled atmosphere and land models [16], etc.

All components of most coupled models are usually independent and interact, or coupled, with each other in a certain way. Various coupling techniques are possible. A program code of one model component can be nested in the code of another as a subroutine, or the model components can run concurrently and communicate with each other through any interface (MPI, file system, UNIX pipe, etc.). An example of the model that uses both these coupling techniques is the INMCM climate model [35]. It includes the General Atmospheric Circulation Model (GACM) component and the General Ocean Circulation Model (GOCM) component, running concurrently and interacting with each other through MPI. At the same time, the land, vegetation and

---

<sup>1</sup>Institute of Computational Mathematics and Mathematical Geophysics, Novosibirsk, Russian Federation

atmospheric aerosol models are implemented in the GACM component as subroutines, and the ice model is implemented in the GOCM component.

Other coupling techniques are associated with a software called “coupler”. The main functions of the coupler are management of data transfer between model components, execution control based on the calendar and time step of model components, and interpolation of coupling data from the computational grid of one model component to the computational grid of the other component. Additional functions can also be delegated to the coupler, such as flux calculation between model components, working with the file system, data assimilation, etc. The set of these functions may depend on the specific problem.

In general, couplers can be referred to one of three types: stand-alone coupler, integrated coupling framework and coupling library. Figure 1 shows the interaction diagrams of some couplers with model components.

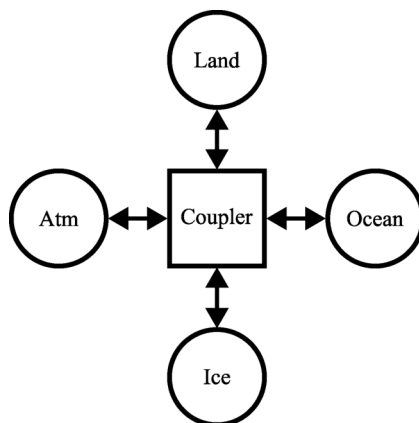
The stand-alone coupler is usually a separate component of the model. In this case, the model components are separate executable modules that run concurrently. Usually, such a coupler is serial and its performance is bottleneck when the array sizes and the number of model components increase, because only the master processes of the model components are involved in interaction with the coupler. The memory usage restrictions arise from this as well. The most advanced examples of such couplers are OASIS3 [34] and older versions of the NCAR CSM Flux coupler (cpl3, cpl4, cpl5), that was used in CCSM model (Fig. 1a).

The integrated coupling framework (e.g. Fig. 1b) is the software that uses component-level interfaces assembling into a single integrated application. This type of system is managed by a high-level driver, and the model component code is usually reorganized as a sequence of method calls, such as initialize, run and finalize, etc. Many systems of this type are usually designed for a predefined set of components, and any changes may require complicated work on the coupler and model component codes. However, the ability of such systems to execute model components sequentially, concurrently or in a mixed mode may give additional opportunities for performance optimization. The most advanced systems of this type are ESMF [6], NCAR CSM Flux Coupler (cpl7) [8], and GFDL FMS [2].

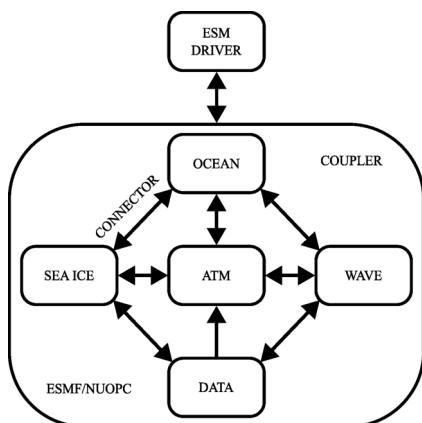
It is worth noting that such systems may have the coupler as a separate module, keeping only a part of the coupler functions, e.g. interpolation or data transfer, and giving other functions, e.g. execution control, to a high-level driver. The NCAR CSM Flux coupler is one of such couplers. Starting with the cpl6 version, this type of coupler works in a parallel mode. There are also couplers which can additionally use abstract interfaces, that enable to couple any number of models with different characteristics. One of them is the CMF (Compact Modeling Framework) coupler [21].

The coupling library (Fig. 1c) is a set of coupling tools that are plugged into each model component as a library. In this case, there is no stand-alone coupler component. Also, a high-level driver is not required in this case. The coupler functions are performed concurrently by a subset of the cores of each of the model components, resulting in improved performance. However, as the number of model components increases, it becomes harder to manage and debug the model code. The OASIS3-MCT [7] and MCT [20, 24] are the most advanced couplers of this type.

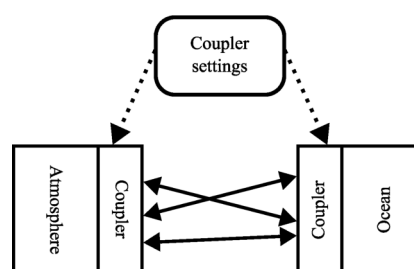
There is no “one-size-fits-all” coupler that is optimal for all coupled models. There is a wide range of couplers that may be suitable for a particular purposes, but not to the user, like such couplers as black box. Nevertheless, it can be convenient to create unique coupler that performs all the additional functions the user needs. In addition, having specific software solution allows



(a) Stand-alone sequential coupler (CCSM model)



(b) Integrated coupling framework (ESMF/NUOPC coupling framework)



(c) Coupling library (OASIS3-MCT)

**Figure 1.** Interaction diagrams of model components with coupler for different coupler types

to fully control its versions and make the necessary changes without any communication with coupler holder. Thus, we decided to develop the SCM (SibCIOM Coupling Module) coupler.

The development of the SCM coupler began in the early 2000s, when it was planned to couple the ocean model developed at ICMMG SB RAS [14, 15] and any sea ice model for Arctic ocean simulations. The CICE model [4, 10, 18, 27], which, at that time, was part of the CCSM (Community Climate System Model), was taken as the sea-ice model. Interaction between the components of this model was performed using the NCAR CSM Flux coupler (cpl4) [23]. Therefore, to avoid modifications in the CICE model, the concept of interaction protocols with the NCAR coupler was taken and the SCM coupler was developed based on it.

Since this coupler accounted for the features of our ocean model and the CICE model, the first version of the coupled sea ice and ocean model SibCIOM (Siberian Coupled Ice-Ocean Model) was developed on its basis. Later, atmosphere and land components were added to the model as pure model components, i.e. they transfer NCEP/NCAR reanalysis data [22] through the coupler to ocean and sea-ice components.

Note that the SCM coupler is designed for coarse-resolution models, in contrast to many modern couplers designed for use on high and ultra-high-resolution models on massively parallel supercomputers. At the time of its development, we were not interested in performance issues. Nevertheless, some changes were made over time, which allowed us to slightly reduce the running

time of the coupler within the SibCIOM model. Note that the coupler code was developed from the ground up.

The main purpose of this article is to describe the design of the resulting SCM coupler and its functionality, since its description has never been published. The new coupled climate models using this coupler are presented. The SCM is written in FORTRAN90 programming language using MPI. The source code is licensed under the GPL-3.0 license and is available on Github [3].

The article consists of three sections. The first and second sections describe SCM coupler design and functionality, respectively. The third section gives examples of constructing two climate models using this coupler.

## 1. SCM Coupler Design

To interact with the coupler, the model components must first be prepared correctly. Then, the work of the coupler and model components can be split into three phases: initialization; exchange of configurations and initial data; and synchronized exchange in time cycle.

### 1.1. Preparation of Model Component

The model being coupled must be either sequential or paralleled using MPI. Note that it is recommended to use the MPI subroutines from [3]: MPI/mpi\_tools.F. The first step is to insert an initialization procedure *setup\_mpi('model\_name')* (see Section 1.2) at the beginning of program and make sure that previous MPI initialization procedures (if they exist) do not override it. Before starting any calculations, the model must receive a one-dimensional integer array of length 100, the first two records of which contain information about the start and end dates of the experiment. This array is then filled one by one with the following data before sending it back to the coupler:

- diagnostic flag (0 – if the model works correctly, 1 – if the model does not work correctly, in which case the whole model will stop);
- the current date of the model component;
- time in seconds elapsed since the beginning of the model's day;
- grid size by longitude;
- grid size by latitude;
- number of model steps per day.

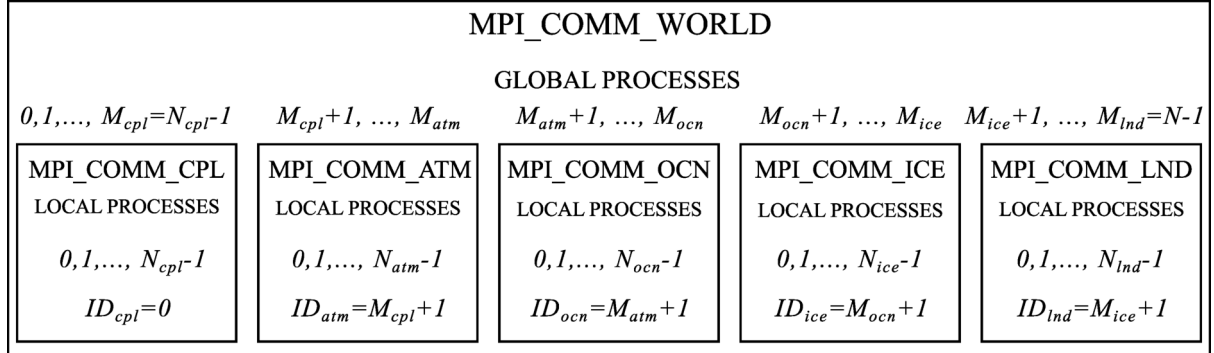
Next, it is necessary to create subroutines for sending calculated model data to the coupler and for receiving data from the coupler, similar to subroutines *send\_atmos\_data* and *get\_atm\_state* (see [3]: Coupler\_8/coupler.F), for example.

The send subroutine should pack a set of two-dimensional arrays into one two-dimensional array with the number of rows equal to the number of arrays to be sent and send it to the coupler. The rows themselves are the original two-dimensional arrays, flattened into one-dimensional arrays. The receive subroutine must get a two-dimensional array from the coupler and reshape each of its rows back into a two-dimensional array. In this case, it is necessary to ensure the consistency of the transmitted data between the coupler and the model component.

These subroutines must then be used before the main model cycle to send the model grid and initial data to the coupler and receive the necessary data from the other models through the coupler.

## 1.2. Initialization of Model Components

The subroutine *setup\_mpi('model\_name')* (see [3]: MPI/mpi\_tools.F) should be called at the beginning of the model components and coupler programs. It initializes all model components in global communicator MPI.COMM\_WORLD, creates the internal communicators and distribute all processes over them (Fig. 2).



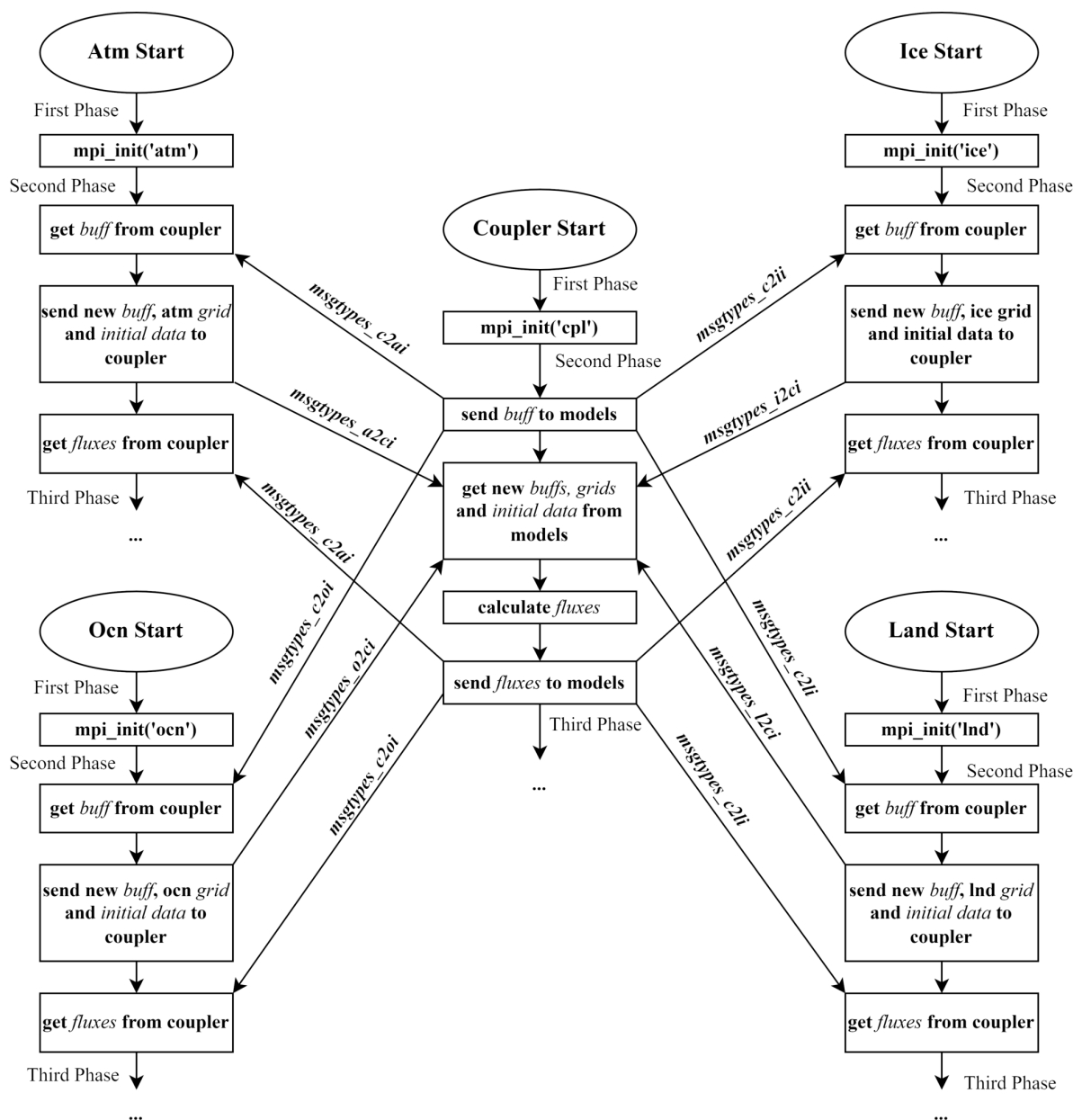
**Figure 2.** Scheme of the *setup\_mpi('model\_name')* subroutine result

Each model component process has a global and a local number. Let  $N$  be total number of processes used by model, and  $N_{cpl}, N_{atm}, N_{ocn}, N_{ice}, N_{lnd}$  are the number of a process used by each model component providing equality  $N_{cpl} + N_{atm} + N_{ocn} + N_{ice} + N_{lnd} = N$ . These numbers also define the maximum number of local processes for each component. Enumeration of global processes is changed for every program run, therefore for certainty we assume that  $M_{cpl}, M_{atm}, M_{ocn}, M_{ice}, M_{lnd}$  are the global numbers of the last process of each component, such that  $N_{cpl} - 1 = M_{cpl} < M_{atm} < M_{ocn} < M_{ice} < M_{lnd} = N - 1$ . Procedures MPI\_GROUP\_RANGE\_INCL and MPI\_COMM\_CREATE are used to create the processes groups and individual communicators (MPI\_COMM\_CPL, MPI\_COMM\_ATM, etc.) for all model components. Within these communicators, model components can perform their own internal communications. Finally, the master processes for each component are defined as global processes with numbers  $ID_{cpl} = 0, ID_{atm} = M_{cpl} + 1, ID_{ocn} = M_{atm} + 1, ID_{ice} = M_{ocn} + 1, ID_{lnd} = M_{ice} + 1$ .

## 1.3. Initial Data and Configuration Exchange

The second phase starts with configuration exchange (Fig. 3). First, the coupler sends the start and end date of the experiment via the *buff* array of size 100 to the model components. Tags *msgtypes\_c2ii, msgtypes\_c2ai, msgtypes\_c2li, msgtypes\_c2oi* (see [3]: Coupler\_8/msgtypes.F) show that the coupler sends initial data to the ice, atmospheric, land and oceanic components, respectively. Similarly, parameters *msgtypes\_i2ci, msgtypes\_a2ci, msgtypes\_l2ci, msgtypes\_o2ci* shows that model components send initial data to coupler. Then coupler receives updated array *buff* and grid configuration from each model. The data structures that the coupler uses will be described below. (see [3]: Coupler\_8/struct.F for details)

The first are structures of type “grid” and “state” that store buff array data and grid configuration. They are defined for each model component. A structure of “state” type contains the next fields: the current date *idate*, the time in seconds elapsed since the beginning of the



**Figure 3.** Flow chart diagram of the first and second phases of coupler and model components

model’s day *sec*, the size of the computation grid *imt*, *jmt*, the number of time steps per day *nadv* and the next time moment of model component *next*.

A structure of “grid” type contains the next two-dimensional arrays: geographical coordinates *lat*, *lon*, the area of the box surrounding the grid points *tarea*, domain mask *mask* and an auxiliary array *work* that can store any additional grid data.

The next step is receiving the initial data of each model component. This data is written in data structures of types “from\_ocn”, “from\_ice”, “from\_atm” and “from\_lnd”. These structures must be modified manually according to the data being sent by the model components. Fluxes after their computation are stored in structures “to\_ice”, “to\_ocn”, “to\_lnd”, “to\_atm” types. The final step of second phase is sending the new data to the model components.

The second phase of any model component includes receiving the coupler configuration, sending its configuration, grid and initial data to the coupler and receiving the necessary data from the other model components (Fig. 3).

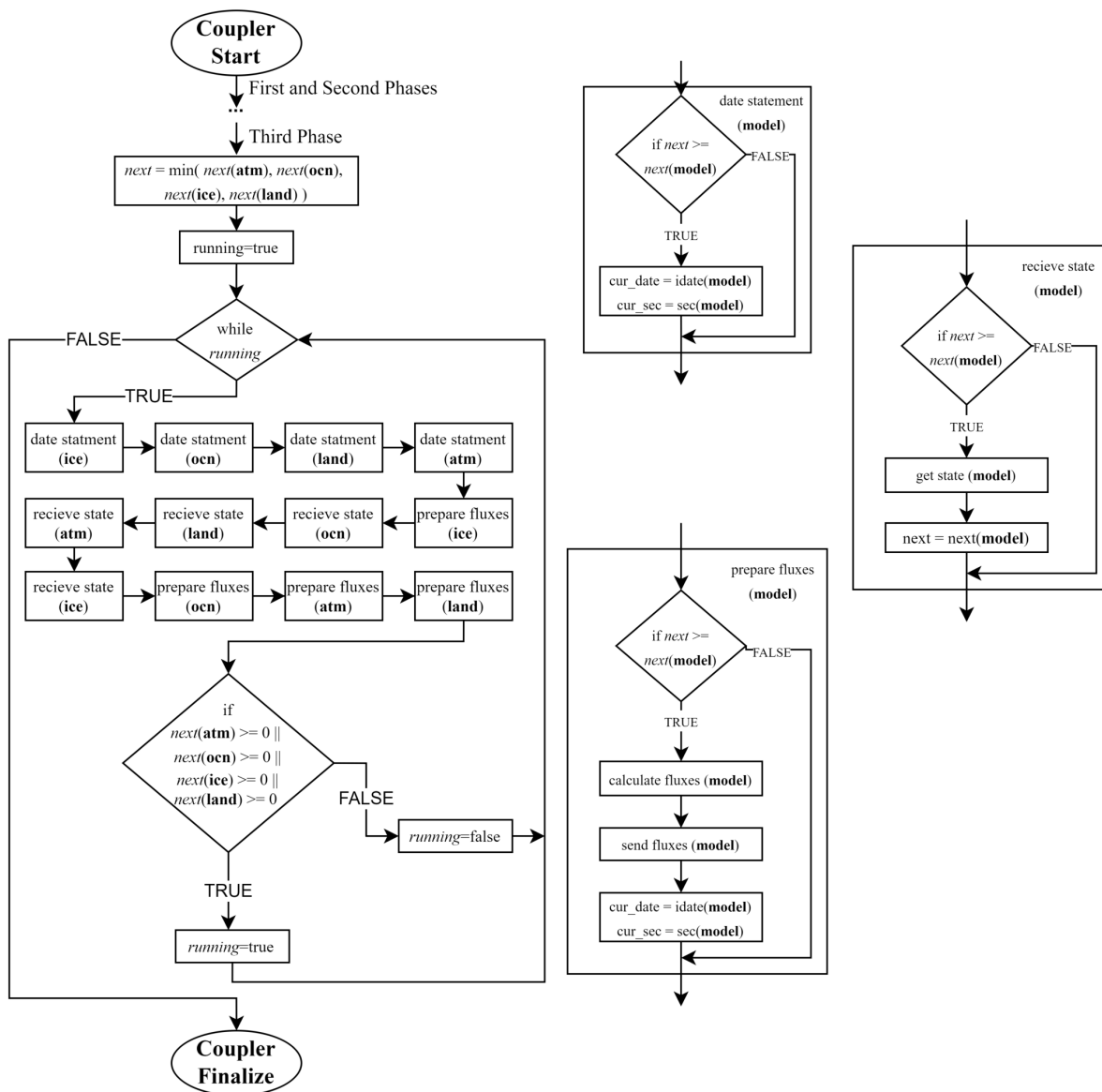


Figure 4. Flow chart diagram of the coupler third phase

### 1.4. Data Exchange Synchronization

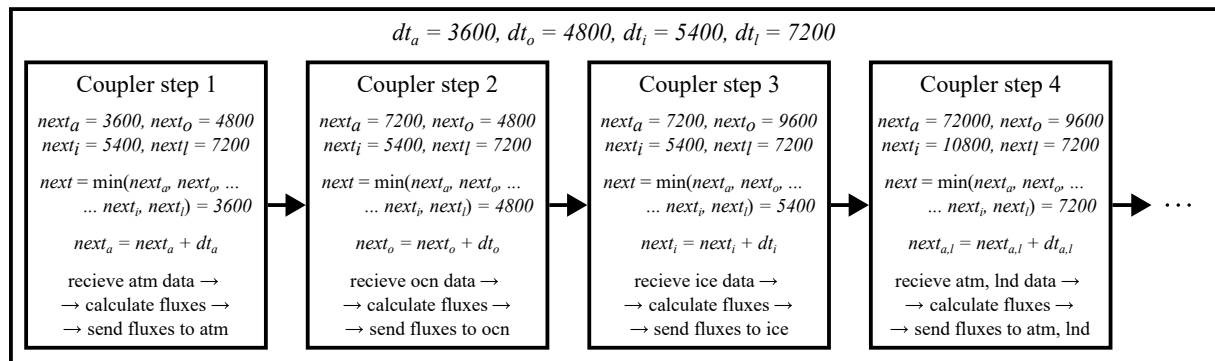
The third phase of coupler includes a “while” loop (Fig. 4). The exit from the loop occurs in two cases: after reaching the final simulation date (the “state” structure field *next* is assumed to be  $-999$  for any model component) or after stopping any of the model components for any reason.

At the same time, the third phase of any model component includes “do” loop where the synchronized data exchange occurs. Moreover, the order of interaction with the coupler remains the same as before the loop (Fig. 3). Within the loop, the *buff* is sent, the current model state is sent, and the necessary data is received from other components.

The coupler knows next time moments  $next(\mathbf{atm})$ ,  $next(\mathbf{ice})$ ,  $next(\mathbf{ocn})$ ,  $next(\mathbf{land})$  for each step of the “while” loop. Among all these moments, the minimum moment  $next$  is chosen, and further data exchange and flux calculation are performed for model components where this minimum moment is greater or equal to one of the next time moments. One should note that this condition can be satisfied for several components at the same time.

Fluxes are calculated using the data from each model component obtained at different time steps, i.e., if  $next \geq next(\mathbf{ice})$ , then ice fluxes are calculated using the atmospheric, land and ocean data from the previous time step and ice data received at the current moment.

Since a coupler is a sequential program, the order established for receiving data, calculating fluxes, and sending updated data is very important. The ocean model is the most time-consuming one; these types of operations are a priority for it. For other model components, these operations take much less time. If the order is changed, the waiting time for other components increases. And this slows down the model operation.



**Figure 5.** Example of the data exchange synchronization in SCM coupler

Finally, let us consider an example of how the process of data exchanges synchronization by the coupler is performed (Fig. 5). Let  $dt_a = 3600$ ,  $dt_o = 4800$ ,  $dt_i = 5400$ ,  $dt_l = 7200$  are time steps (in seconds) of each model component. Let us also assume that  $next_a = next(\mathbf{atm})$ ,  $next_o = next(\mathbf{ocn})$ ,  $next_i = next(\mathbf{ice})$  and  $next_l = next(\mathbf{land})$ . Before the while loop starts, we have  $next_a = dt_a$ ,  $next_o = dt_o$ ,  $next_i = dt_i$ ,  $next_l = dt_l$ . At the first step,  $next = next_a = 3600$ , i.e.,  $next \geq next_a$ . Therefore, data exchange and flux calculation are performed only for the atmospheric component. The next time moment for atmosphere is  $next_a = next_a + dt_a = 7200$ . A similar process takes place for the second and third steps within the ocean and ice components, respectively. For the fourth step, we have  $next \geq next_a$  and  $next \geq next_l$ , thus, in this step, data exchange and flux calculation are performed first for the atmosphere, then for the land.

## 2. SCM Coupler Functionality

This chapter describes other functions of the SCM coupler, such as interpolation, flux calculation and tidal effects accounting. These functions are used in the data preparation subroutines before being sent to the model components. The preparation subroutines are called *prepare\_atmos\_support*, *prepare\_ocean\_support*, *prepare\_ice\_support* and *prepare\_land\_support* (see [3]: Coupler\_8/prepare.F).



## 2.1. Fluxes Calculation

Flux is the exchange rate of some media characteristics specified for a unit area per unit time. Fluxes determine how the climate components interact with each other. For each pair of model components, different fluxes are calculated e.g., heat and radiation fluxes from the atmosphere to the ocean ice, greenhouse gases fluxes from the land to the atmosphere, fresh water fluxes from ice to ocean, etc.

The flux calculation problem can be solved either by the coupler, which has information about the state of all components at any given time, or by the major component at the moment when it has all the necessary data from the other components. Note that in the general case it is difficult to determine where the fluxes should be calculated, either in the coupler or in any model component, because it depends on the features of a particular model. Therefore, the solution of this problem is left to the model developers.

The current version of the SCM coupler fully computes fluxes from atmosphere to ocean (see [3]: Coupler\_8/flux\_ao.F). Fluxes from atmosphere to ice are computed by the ice model, because the CICE model has its own flux calculation procedures. Moving these procedures from the ice component to the coupler requires a significant rearrangement in the model code and may cause incorrect model behavior. Nevertheless, the SCM coupler provides the option of the atmosphere-to-ice fluxes computation.

At the atmosphere-ocean interface, the values of wind stress  $\vec{\tau}$ , humidity  $E$ , sensible heat flux  $H$ , upward long-wave radiation flux  $L_{\uparrow}$  and  $CO_2$  flux from the atmosphere to ocean must be calculated. The bulk aerodynamic formulae are used to evaluate  $\vec{\tau}$ ,  $E$  and  $H$ . The general expressions are the following:

$$\begin{aligned} \vec{\tau} &= \rho_A (u^*)^2 \frac{\Delta\vec{U}}{|\Delta\vec{U}|}, & E_s &= \rho_A u^* Q^*, & H_s &= \rho_A C_{P_A} u^* \theta^*, \\ u^* &= \sqrt{C_U} |\Delta\vec{U}|, & Q^* &= \frac{C_E |\Delta\vec{U}| \Delta q}{u^*}, & \theta^* &= \frac{C_T |\Delta\vec{U}| \Delta\theta}{u^*}, \\ \Delta\vec{U} &= \vec{U} - \vec{U}_A, & \Delta q &= q - q_A, & \Delta\theta &= T - \theta_A, \end{aligned}$$

where  $\rho_A$  is the atmospheric surface density,  $C_{P_A}$  is the specific heat capacity of atmosphere,  $C_U, C_E, C_T$  are dimensionless drag-coefficients for wind stress, humidity and sensible heat, respectively,  $\vec{U}_A, q_A, \theta_A$  are the horizontal wind velocity vector, the specific air humidity and the potential atmospheric temperature at some reference height  $\zeta_r$ , respectively, and  $\vec{U}, q, T$  are the vector of horizontal ocean current velocity, the saturation humidity at the atmosphere-ocean interface, and the ocean surface temperature, respectively. The differences  $\Delta\vec{U}, \Delta q, \Delta\theta$  are defined so that the downward fluxes are positive. It is worth noting that the velocities in the atmosphere are considerably higher than in the ocean, so one can neglect the ocean velocity in the formulas and put  $\Delta\vec{U} = -\vec{U}_A$ .

All bulk formulas differ from each other in a manner how the parametric coefficients  $C_U, C_E, C_T$  are evaluated. Two evaluation approaches are implemented in the SCM coupler: one is based on the COARE3.0 algorithm [1, 11], the other is based on the algorithm presented in the CSM Flux coupler (cpl4) user guide [23].

The general formula to calculate  $L_{\uparrow}$  is as in [23]

$$L_{\uparrow} = -\varepsilon\sigma T^4 + \alpha^L L_{\downarrow}, \quad (1)$$

where  $\sigma = 5.67 \times 10^{-8} \frac{W}{m^2 K^4}$  is the Stefan-Boltzmann constant,  $\varepsilon$  is the emissivity of the interface, and  $\alpha^L$  is the surface albedo for incident longwave radiation  $L_\downarrow$ . The value of  $L_\uparrow$  also depends on the air humidity, cloud cover and difference between the surface (ocean or ice) temperature  $T_S$  and near- surface air temperature  $T_A$ . Thus, the formula (1) becomes (see [5]):

$$L_\uparrow = L_\uparrow^0 + 4\varepsilon\sigma T_S^3 (T_S - T_A). \quad (2)$$

Two ways to determine  $L_\uparrow^0$  can be used in the coupler:

$$\begin{aligned} L_\uparrow^0 &= \varepsilon\sigma T_S^4 (0.39 - 0.05\sqrt{e_a}) (1 - 0.6c^2), \\ L_\uparrow^0 &= \varepsilon\sigma T_S^4 (0.39 - 0.05\sqrt{e_a}) (1 - 0.8c), \end{aligned}$$

where  $e_a$  is the water vapor pressure in atmosphere, and  $c$  is the fractional cloud cover. The first expression is described in [13], the second one in [32]. The second expression was used by various Arctic Ocean models, including SibCIOM, in the framework of the AOMIP (Arctic Ocean Inter comparison Project).

There are also two ways to account  $L_\uparrow$ . In the first case, the fluxes  $L_\uparrow^I$  and  $L_\uparrow^O$  for the ocean surface and sea ice are calculated separately by formula (2) and the total flux is calculated as  $L_\uparrow = \beta L_\uparrow^I + (1 - \beta) L_\uparrow^O$ , where  $\beta$  is ice compactness. In the second case, it is assumed that  $T_S = \varepsilon_I T_I \beta + \varepsilon_O T_O (1 - \beta)$  and  $\varepsilon = \varepsilon_I \beta + \varepsilon_O (1 - \beta)$ , where  $\varepsilon_I, \varepsilon_O$  are emissivity for ice and ocean. Then the calculation is carried out according (2).

## 2.2. Data Interpolation

Let for some model component, for example, the ice model, the influence of atmospheric, ocean and land components have been described. These components send their current states and fluxes, based on these states, to the ice model using the coupler. Each component has its own computational grid. Let us call the grids of the atmosphere, ocean, and land models the source grids, and the ice model grid as the destination grid. To use all necessary data by the ice component, the data from the source grids must be interpolated onto the destination grid. This requires an interpolation subroutine.

The subroutine used in the coupler is based on the interpolation function described in [26]. Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be a function of  $n$  variables and  $\{\vec{x}_k \in \mathbb{R}^n, k = 1, \dots, M\}$  be a grid of any domain. The value of  $f(\vec{x})$  in some grid point  $\vec{x}_k$  is denoted by  $f_k = f(\vec{x}_k)$ . The model component grid can include both points used and not used for calculation, e.g., the ocean model grid can include land points that are not used in the calculation (Fig. 2). Therefore, let us introduce additionally the mask of the domain as a function  $m : \mathbb{R}^n \rightarrow \{0, 1\}$  such that  $m(\vec{x}_k) = m_k$ , where  $m_k = 1$  if  $\vec{x}_k$  is a computational point and  $m_k = 0$  if  $\vec{x}_k$  is not a computational point. The interpolation function is defined by the formula:

$$u(\vec{x}) = \frac{\sum_{k=1}^M W_k(\vec{x}) f_k}{\sum_{k=1}^M W_k(\vec{x})}, \quad W_k(\vec{x}) = \begin{cases} \exp\left(-E \frac{r_k^2}{R^2}\right), & r_k \leq R \vee m_k = 1 \\ 0, & r_k > R \vee m_k = 0 \end{cases},$$

where  $R$  is the radius of the circle centered at  $\vec{x} \in \mathbb{R}^n$ ,  $r_k = \rho(\vec{x}, \vec{x}_k) = |\vec{x} - \vec{x}_k|$  and  $E = 4$ .

The interpolation subroutine is time consuming enough, so if the model grids are fixed, it is better to compute the coefficients  $W_k(\vec{x})$  once and use them for every time step. During the first

fluxes calculation, the coupler creates the files containing the matrices of interpolation coefficients  $W_k(\vec{x})$  for each pair of model components. Then these matrices are stored in memory and used during computation. If the grid configuration for another model simulation has not changed, the interpolation coefficients are read from the saved files during the first flux calculation and used further. If the grid of any component has changed, it is necessary to delete manually the files with interpolation coefficients, related to this component. In this case, only part of the matrices will be recalculated.

Note that in addition to scalar data, vector data can also be sent to the components. Since the source grid and the destination grid can be rotated relative to each other by a certain angle, there is another interpolation procedure for vector data, based on the same interpolation function: it can rotate the vector on the destination grid by the required angle.

### 2.3. Tidal Model

The tidal phenomena have a significant impact on the dynamics of sea ice in the Arctic. For example, they can affect the sea ice velocity, its formation and degradation, etc. A more detailed description of the tide effect can be found in [17]. The coupler has a subroutine for describing the tidal phenomena (see [3]: Coupler\_8/tides.F). It is based on the barotropic inverse model of the Northern Ice Ocean [29]. This module is called in the ice flux calculation procedure and it is universal for any ice model attached to the coupler, since it uses the sea ice model mask and the current date as input data.

## 3. Constructing New Climate Models Using SCM Coupler

One of the important tasks in Arctic climate research is to investigate the causal relationships between the Earth's and Arctic's climate changes. The SibCIOM model is useful for solving these kinds of problems. However, this model does not use dynamical atmosphere and land models, which makes it impossible to perform more detailed research. Therefore, it was decided to modify the SibCIOM model using the existing GACM instead of pure data components. The atmospheric model PUMA (Portable University Model of Atmosphere) of the intermediate complexity climate model PlaSim was taken first, and then the GACM of the INMCM48 climate model.

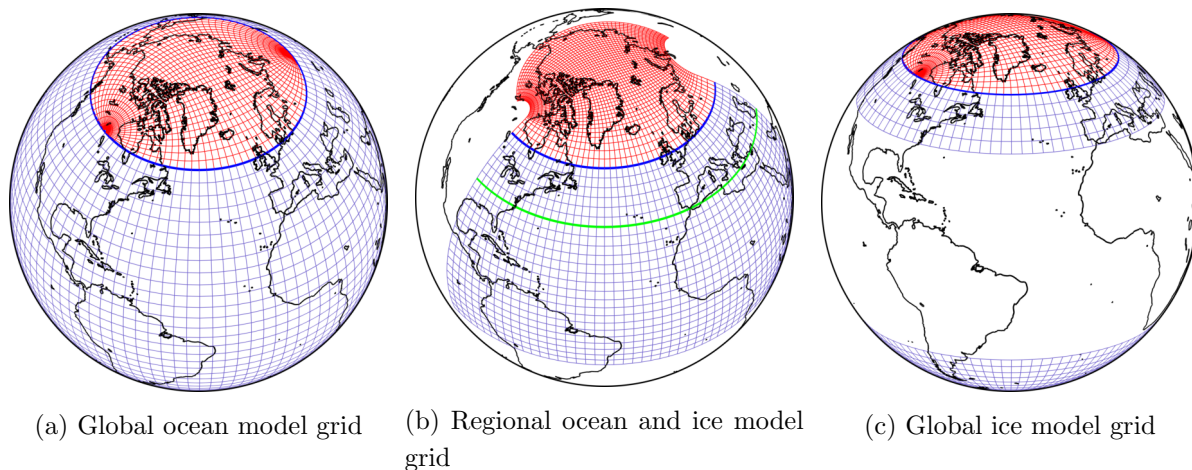
### 3.1. Overview of Basic Models

#### SibCIOM model

The SibCIOM model is a coupled ice-ocean model and has four independent model components: oceanic, atmospheric, sea-ice and land components interacting through the SCM coupler (Fig. 1a). There exist regional and global versions model.

The grid for the ocean component is a composite tripolar grid [28] composed of two components. The first component is spherical grid with a pole at 90°S. The second component is an asymmetric bipolar grid (“polar cap”) with poles on land: one in North America and the second in Eurasia.

The grid of the global version (Fig. 6a) has a horizontal resolution of  $1^\circ \times 1^\circ$  in longitude and latitude in the spherical part and from 35 to 94 km within the polar cap. The regional version uses a grid (Fig. 6b) that covers only the Arctic Ocean and a part of the Atlantic Ocean above



**Figure 6.** The SibCIOM composite grid composed of a spherical grid (blue) with a pole at  $90^{\circ}\text{S}$  and a bipolar asymmetric grid (red) with two poles allocated in North America and Eurasia

the latitude of  $20^{\circ}\text{S}$ . There are grids with a resolution of  $0.5^{\circ} \times 0.5^{\circ}$  and  $0.25^{\circ} \times 0.25^{\circ}$  in latitude and longitude directions in the spherical part and from 7 to 55 km and from 3 to 27 km in the polar cap, respectively. Both versions of the model have 38 vertical levels to a depth 5500 m.

The ice model uses the same grid as the ocean model for both model versions, though this is not necessary. The spherical area from  $35^{\circ}\text{S}$  to  $35^{\circ}\text{N}$  (Fig. 6c) excluded in the global version because there is no ice in this area. The regional version uses the ocean model grid (Fig. 6b) above  $35^{\circ}\text{N}$  (green line).

Current model configurations use different time steps  $\Delta T_{ocn}$  and  $\Delta T_{ice}$ . Time steps for the global version with resolution  $1^{\circ} \times 1^{\circ}$  are  $\Delta T_{ocn} = 4800\text{ s}$  and  $\Delta T_{ice} = 5400\text{ s}$ . The time steps of the regional version of the ocean model at resolutions  $0.5^{\circ} \times 0.5^{\circ}$  and  $0.25^{\circ} \times 0.25^{\circ}$  are  $\Delta T_{ocn} = 2400\text{ s}$  and  $\Delta T_{ocn} = 1800\text{ s}$ , respectively. At the same time, the step of the ice model in both cases is  $\Delta T_{ice} = 3600\text{ s}$ . The model components are synchronized according to the diagram described in Section 1.4.

SibCIOM is written in FORTRAN90 programming language using MPI. Oceanic and ice components are parallelized using MPI while atmospheric, land components and coupler are serial programs. MPI functions are also used for data exchange between coupler and other model components.

### INMCM48 model

INMCM48 is a global climate model developed at the INM RAS. It is able to simulate the state of the atmosphere, ocean, sea ice, land and vegetation, taking into account the greenhouse gases. This model has the atmospheric (GACM) and oceanic (GOCM) components with mutual interaction.

The atmospheric component calculates the state of the atmosphere, land, vegetation and dynamics of aerosols. Here the horizontal resolution is  $2^{\circ} \times 1.5^{\circ}$  in longitude and latitude, respectively. Vertically, 21  $\sigma$ -levels up to  $\sigma = 0.01$  are used. The oceanic component simulates the state of the ocean and sea ice. It has a resolution  $1^{\circ} \times 0.5^{\circ}$  in longitude and latitude and 40 vertical  $\sigma$ -levels. The maximum depth is 5500 m. In current version time step of the GACM is  $\Delta T_{atm} = 240\text{ s}$ , while the GOCM time step is  $\Delta T_{atm} = 1800\text{ s}$ .

The GACM and GOCM components of the INMCM48 model can be run either in stand-alone mode (Fig. 7a) or in a coupled mode (Fig. 7b). The coupled run is controlled by a simple driver that distributes all MPI processes within the MPI\_COMM\_WORLD communicator to the four communicators: atmospheric (MPI\_COMM\_ATM), oceanic (MPI\_COMM\_OCEAN), atmospheric-trace (MPI\_COMM\_ATM\_XX) and ocean-trace (MPI\_COMM\_OCEAN\_XX). The tracer models are optional and work only with their corresponding main models (Fig. 7). For stand-alone running, there are individual drivers for each component.

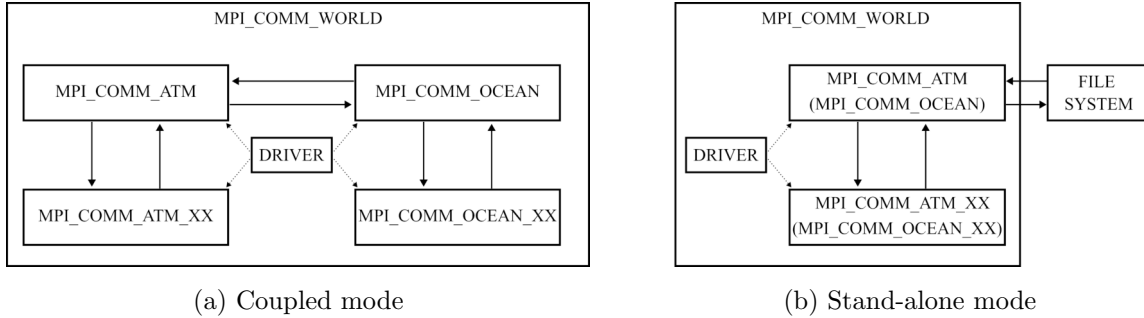


Figure 7. INMCM48 model interaction diagrams

As mentioned in the introduction, the GACM and GOCM components are coupled directly through master processes without any coupler. The GACM sends to GOCM data every 2 hours, while GOCM sends data every 1 hour. At the same time, the GACM each next time step after the ocean data received from the GOCM uses the same data for the computations.

INMCM48 is coded with FORTRAN77 language. GACM and GOCM components parallelized using MPI. This model is unique and has been used in various climate researches. It has also been verified many times during the CMIP experiments.

### PlaSim model

Intermediate complexity climate models are the models allowing various simplifications in describing some physical processes. Such models are suitable in investigating particular physical processes and relationships between them when some details can be neglected, e.g. subscale processes. One such model is the PlaSim (Planet Simulator) global climate model developed at the Institute of Meteorology (the University of Hamburg).

PlaSim model includes atmospheric, ocean, sea ice, land and biosphere components. The dynamic core of the model is the PUMA (Portable University Model of the Atmosphere). The other components are nested as subroutines in the atmospheric model.

The PlaSim model works with fine resolution grids and has a limited number of grid resolution configurations. The number of grid points by latitude is  $2^n$ , by longitude is  $2^{n+1}$ , where  $n = 1, \dots, 8$ . The vertical coordinate is  $\sigma$ , and the minimum number of vertical levels is 5. Note, that all components use the same grid. Default time step of the atmosphere model is  $\Delta T_{atm} = 2700 s$ , while the sea ice and ocean model use a time step of one day.

The model does not have an external coupler and works in SIMD mode. Atmosphere and sea ice models coupled directly every 32 time steps and the sea ice and ocean model are coupled every time step.

**Table 1.** Arrays sent by the atmospheric component to the coupler

Atmosphere → Coupler				
Variable name	Name in model	SibCIOM	PlaSim-ICMMG1.0	INMCM-SibCIOM
Wind velocity height	zlvl	+	+	+
Temperature and humidity height	zvlv	+	+	+
<i>U</i> -wind velocity components	uatm	+	+	+
<i>V</i> -wind velocity components	vatm	+	+	+
Potential temperature	potT	+	+	+
Air temperature	Tair	+	+	+
Specific humidity	Qa	+	+	+
Air density	rhoa	+	+	+
Downward shortwave radiation	Fsw	+	+	+
Downward longwave radiation	Flw	+	+	+
Precipitation rate (rain)	Frain	+	+	+
Precipitation rate (snow)	Fsnow	+	+	+
Cloudiness	cld	+	+	–
CO <sub>2</sub> flux to ocean	cco2	+	+	+
Greenland river discharge transport	small_riv	+	+	+
<i>U</i> -wind stress component	taux	–	–	+
<i>V</i> -wind stress component	tauy	–	–	+
Upward longwave radiation	lwup	–	–	+
Sensible heat flux	hsn	–	–	+
Latent heat flux	hlt	–	–	+

**Table 2.** Arrays delivered by the coupler to the atmospheric component

Coupler → Atmosphere				
Variable name	Name in model	SibCIOM	PlaSim-ICMMG1.0	INMCM-SibCIOM
Surface temperature	tsurf	–(+)	+	+
Drag-coefficient for wind stress	cu	–(+)	+	–
Drag-coefficient for sensible heat flux	ct	–(+)	+	–
Drag-coefficient for evaporation flux	ce	–(+)	+	–
Surface albedo	albed	–(+)	+	–
Ice compactness	dicec	–(+)	+	+
CO <sub>2</sub> flux to the atmosphere	fco2	–(+)	+	+
Sensible heat flux	hsn	–(+)	–	+
Latent heat flux	hlt	–(+)	–	+

### 3.2. Construction of New Climate Models

To couple a new model component to the coupler, the component must be configured as described in Section 1.1. Then, the coupling is carried out through MPI data exchange procedures.

During data exchange step, each model component sends to the coupler its current state described by a set of two-dimensional arrays. The current states of the models are required to compute the fluxes. After the fluxes have been computed, the coupler sends back to each component the other sets of two-dimensional arrays. For example, in the SibCIOM model, the atmospheric component sends to the coupler 15 arrays (Tab. 1), which are sufficient for calculating the atmospheric fluxes between atmosphere and ocean and atmosphere and ice. The coupler may send any data to the atmospheric component, but the data will not be used by it since the atmospheric component is a pure model component (Tab. 2). In the general case, the set of arrays is not fixed and depends on the coupled model. Note that the data transfer from the coupler to the atmospheric component can be enabled when testing the coupler's communication capabilities.

Two examples of constructing the climate models using the SCM coupler are described below. The main principle is replacing the atmospheric component of the SibCIOM model with a General Atmospheric Circulation Model. The first model is obtained by coupling the GACM of INMCM48 model [36], the second model is obtained by using the GACM of PlaSim model [12].

#### INMCM-SibCIOM model

When building the INMCM-SibCIOM model, the original driver of the GACM component was modified by replacing the original initialization subroutines with the *setup\_mpi('model\_name')* (see Section 1.2). The MPI communicator of the GACM becomes the sub-communicator created in this subroutine.

The data exchange process between the components was also changed. In the original INMCM48 model, the GACM component received 5 two-dimensional arrays, and then sent 11 two-dimensional arrays, one one-dimensional array and one constant. To synchronize the exchanges between coupler and GACM, the order of exchanges was reversed: sending to coupler first, then receiving from coupler. The original subroutines were replaced with the new ones using functions from [3]: *MPI/mpi\_tools.F*. The arrays sent to the coupler are shown in Tab. 1. The arrays received from the coupler remained the same as for the original model (see. Tab. 2). The coupling also began to be performed every step of the atmosphere model instead of two, as in the original model.

It is worth noting that flux calculation functions from the atmosphere to the ocean were disabled in the coupler to avoid a significant heat imbalance. The program code of the atmosphere model is too complex to make various changes to it, so the best solution was to send the fluxes calculated by GACM.

Thus, the INMCM-SibCIOM model is a global climate model composed of three model components coupled with each other through the SCM coupler. Since the GACM includes a nested land model, the interaction diagram of the model components corresponds to diagram from Fig. 1a with the missing land component. The configurations of the grids and time steps correspond to ones that global SibCIOM model and GACM of the INMCM48 model use (Section 3.1).

The availability of such a climate model gives more opportunities to investigate a wide range of problems related, for example, to the study of the response of atmospheric circulation to climate change in the Arctic.

### **PlaSim-ICMMG1.0 model**

The PUMA model is coupled to the SCM coupler in the same way as the GACM of the INMCM48 model. Subroutines for data exchange, consistent with the SCM coupler protocol, were developed and the necessary synchronization points were created. The process of model development is described in more detail in [31].

The PUMA also has a nested land model like the GACM of INMCM48. Therefore, the interaction diagram corresponds to that of the INMCM-SibCIOM model. The set of arrays to be exchanged is presented in Tab. 1 and Tab. 2. In this model, the fluxes from the atmosphere to the ocean are computed by the coupler in the same way as for the SibCIOM model.

The grid configurations and time steps for the ocean and ice models are the same as in the global version of the SibCIOM model. The grid of the atmospheric model has a resolution  $2.8125^\circ \times 2.8125^\circ$  in latitude and longitude (64 by 128 nodes), respectively. The coupling is performed each time step  $\Delta T_a = 1200$  s.

Thus, the PlaSim-ICMMG1.0 model is an intermediate complexity model since it has a PUMA as the atmospheric component, but with complete ocean and sea ice models from the SibCIOM model. Therefore, this model can be used, e.g., to study both changes in the large-scale atmospheric dynamics occurring on large time scales due to changes in the Arctic climate and, conversely, the long-term effect of large-scale atmospheric phenomena on the climate of the Arctic region. In particular, this model was used to investigate the mechanisms that influence the recently observed Arctic climate warming [31].

## **Conclusions**

The SCM coupler performs its primary functions, such as exchange synchronization and interpolation, and has other optional functions, such as flux calculation and tidal effect accounting. But, it is less functional concerning the already existing couplers. For example, the OASIS coupler is implemented in several programming languages (Fortran, and C), has coupling library and integrated coupling framework versions, and includes several types of interpolation functions. Nevertheless, having our software product makes it significantly easier to control its versions and allows us to modify it with less effort.

The SCM coupler allows coupling different models of climate system components, as was demonstrated by the GACM components of the INMCM48 and PlaSim models. Thus, climate model INMCM-SibCIOM and climate model of intermediate complexity PlaSim-ICMMG1.0 were built. These models allow solving various problems of climate variability on different time scales. Moreover, they enable us to analyze and understand in more detail the drivers of climate change in the Arctic region and estimate its impact on the global climate.

Several modifications are planned for the next version of SCM coupler. First is the implementation of its parallelized version to improve performance. In particular, it would be useful to implement parallel flux calculation and parallel interpolation in the coupler as it is done in modern couplers. This is especially important when increasing the grid resolution of the model. Also, multiple MPI processes can concurrently exchange data with model components, which will reduce coupler idle time. Second is adding an interpolation procedure with the conservation



property for arbitrarily shaped grids, since the standard interpolation procedure may over- or underestimate the interpolated value in a grid cell. It is crucial, for example, when calculating radiation and heat fluxes since there is a balance of heat and radiation in the simulated climate system, and disbalance can lead to errors. The third task is creating more flexible versions of the coupler suitable for coupling to any number of different components.

The source code SCM coupler is available on Github [3].

## Acknowledgements

This work was conducted within the framework of the budget project 0251-2022-0003 for ICMMG SB RAS. We would like to thank the reviewers for their time and effort in reviewing the manuscript. We sincerely appreciate all the meaningful comments and suggestions that have helped us to improve the quality of the manuscript.

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

## References

1. COARE algorithm, Version 3.0, <https://github.com/NOAA-PSL/COARE-algorithm.git>
2. Flexible Modeling System (FMS) Coupler, <https://github.com/NOAA-GFDL/FMSCoupler.git>
3. SCM (SibCIOM Coupling Module) coupler, <https://github.com/SibCIOM/SCM-coupler.git>
4. Bitz, C.M., Lipscomb, W.H.: An energy-conserving thermodynamic model of sea ice. *JGR: Oceans* 104(C7), 15669–15677 (1999). <https://doi.org/10.1029/1999JC900100>
5. Budyko, M.I.: *Climate and Life*. International geophysics series, Academic Press (1974), <https://books.google.ru/books?id=U0IazQEACAAJ>
6. Collins, N., Theurich, G., DeLuca, C., *et al.*: Design and implementation of components in the earth system modeling framework. *The International Journal of High Performance Computing Applications* 19(3), 341–350 (2005). <https://doi.org/10.1177/1094342005056120>
7. Craig, A., Valcke, S., Coquart, L.: Development and performance of a new version of the OASIS coupler, OASIS3-MCT\_3.0. *Geoscientific Model Development* 10(9), 3297–3308 (2017). <https://doi.org/10.5194/gmd-10-3297-2017>
8. Craig, A., Vertenstein, M., Jacob, R.: A new flexible coupler for earth system modeling developed for CCSM4 and CESM1. *The International Journal of High Performance Computing Applications* 26(1), 31–42 (2012). <https://doi.org/10.1177/1094342011428141>
9. Cummins, D.P., Stephenson, D.B., Stott, P.A.: Optimal Estimation of Stochastic Energy Balance Model Parameters. *Journal of Climate* 33(18), 7909–7926 (2020). <https://doi.org/10.1175/JCLI-D-19-0589.1>

10. Dukowicz, J.K., Baumgardner, J.R.: Incremental Remapping as a Transport/Advection Algorithm. *Journal of Computational Physics* 160(1), 318–335 (2000). <https://doi.org/10.1006/jcph.2000.6465>
11. Fairall, C.W., Bradley, E.F., Hare, J.E., *et al.*: Bulk Parameterization of AirSea Fluxes: Updates and Verification for the COARE Algorithm. *Jour. of Climate* 16(4), 571–591 (2003). [https://doi.org/10.1175/1520-0442\(2003\)016<0571:BPOASF>2.0.CO;2](https://doi.org/10.1175/1520-0442(2003)016<0571:BPOASF>2.0.CO;2)
12. Fraedrich, K., Jansen, H., Kirk, E., *et al.*: The Planet Simulator: Towards a user friendly model. *Meteorologische Zeitschrift* 14(3), 299–304 (2005). <https://doi.org/10.1127/0941-2948/2005/0043>
13. Gill, A.: *Atmosphere-Ocean Dynamics*. International Geophysics Series, Academic Press (1982), <https://books.google.ru/books?id=UOIazQEACAAJ>
14. Golubeva, E.N.: Numerical modeling of the Atlantic Water circulation in the Arctic Ocean using QUICKEST scheme. *Computational technologies* 13(5), 11–24 (2008), <http://www.ict.nsc.ru/jct/annotation/1166?l=eng>, (in Russian)
15. Golubeva, E.N., Platov, G.A.: On improving the simulation of Atlantic Water circulation in the Arctic Ocean. *Journal of Geophysical Research: Oceans* 112(C4) (2007). <https://doi.org/10.1029/2006JC003734>
16. Haggag, M., Yamashita, T., Lee, H., Kim, K.: A coupled atmosphere and multi-layer land surface model for improving heavy rainfall simulation. *Hydrology and Earth System Sciences Discussions* 5, 1067–1100 (2008). <https://doi.org/10.5194/hessd-5-1067-2008>
17. Holloway, G., Proshutinsky, A.: Role of tides in Arctic ocean/ice climate. *Journal of Geophysical Research: Oceans* 112(C4) (2007). <https://doi.org/10.1029/2006JC003643>
18. Hunke, E.C., Dukowicz, J.K.: An ElasticViscousPlastic Model for Sea Ice Dynamics. *Journal of Physical Oceanography* 27(9), 1849–1867 (1997). [https://doi.org/10.1175/1520-0485\(1997\)027<1849:AEVPMF>2.0.CO;2](https://doi.org/10.1175/1520-0485(1997)027<1849:AEVPMF>2.0.CO;2)
19. Jackson, L.S., Maycock, A.C., Andrews, T., *et al.*: Errors in Simple Climate Model Emulations of Past and Future Global Temperature Change. *Geophysical Research Letters* 49(15), e2022GL098808 (2022). <https://doi.org/10.1029/2022GL098808>
20. Jacob, R., Larson, J., Ong, E.:  $M \times N$  Communication and Parallel Interpolation in Community Climate System Model Version 3 Using the Model Coupling Toolkit. *The International Journal of High Performance Computing Applications* 19(3), 293–307 (2005). <https://doi.org/10.1177/1094342005056116>
21. Kalmykov, V.V., Ibrayev, R.A., Kaurkin, M.N., Ushakov, K.V.: Compact Modeling Framework v3.0 for high-resolution global oceaniceatmosphere models. *Geoscientific Model Development* 11(10), 3983–3997 (2018). <https://doi.org/10.5194/gmd-11-3983-2018>
22. Kalnay, E., Kanamitsu, M., Kistler, R., *et al.*: The NCEP/NCAR 40-Year Reanalysis Project. *Bulletin of the American Meteorological Society* 77(3), 432–472 (1996). [https://doi.org/10.1175/1520-0477\(1996\)077<0437:TNYRP>2.0.CO;2](https://doi.org/10.1175/1520-0477(1996)077<0437:TNYRP>2.0.CO;2)
23. Kauffman, B.G., Large, W.G.: NCAR CSM Flux Coupler, version 4.0. <https://www2.cesm.ucar.edu/models/cpl/doc4> (1998), accessed: 2023-01-15

24. Larson, J., Jacob, E., Ong, E.: The Model Coupling Toolkit: A New Fortran90 Toolkit for Building Multiphysics Parallel Coupled Models. *The International Journal of High Performance Computing Applications* 19(3), 277–292 (2005). <https://doi.org/10.1177/1094342005056115>
25. Leach, N.J., Jenkins, S., Nicholls, Z., *et al.*: FaIRv2.0.0: a generalized impulse response model for climate uncertainty and future scenario exploration. *Geoscientific Model Development* 14(5), 3007–3036 (2021). <https://doi.org/10.5194/gmd-14-3007-2021>
26. Levitus, A., Boyer, T.P.: World ocean atlas 1994. Vol. 4, Temperature. NOAA atlas NESDIS; 4 (1994), <https://repository.library.noaa.gov/view/noaa/1381>
27. Lipscomb, W.H., Hunke, E.C.: Modeling Sea Ice Transport Using Incremental Remapping. *Monthly Weather Review* 132(6), 1341–1354 (2004). [https://doi.org/10.1175/1520-0493\(2004\)132<1341:MSITUI>2.0.CO;2](https://doi.org/10.1175/1520-0493(2004)132<1341:MSITUI>2.0.CO;2)
28. Murray, R.J.: Explicit Generation of Orthogonal Grids for Ocean Models. *Journal of Computational Physics* 126(2), 251–273 (1996). <https://doi.org/10.1006/jcph.1996.0136>
29. Padman, L., Erofeeva, S.: A barotropic inverse tidal model for the Arctic Ocean. *Geophysical Research Letters* 31(2) (2004). <https://doi.org/10.1029/2003GL019003>
30. Perezhugin, P., Chernov, I., Iakovlev, N.: Advanced parallel implementation of the coupled ocean–ice model FEMAO (version 2.0) with load balancing. *Geoscientific Model Development* 14(2), 843–857 (2021). <https://doi.org/10.5194/gmd-14-843-2021>
31. Platov, G., Krupchatnikov, V., Martynova, Y., *et al.*: A new earths climate system model of intermediate complexity, PlaSim-ICMMG-1.0: description and performance. *IOP Conference Series: Earth and Environmental Science* 96(1), 012005 (2017). <https://doi.org/10.1088/1755-1315/96/1/012005>
32. Rosati, A., Miyakoda, K.: A General Circulation Model for Upper Ocean Simulation. *Journal of Physical Oceanography* 18(11), 1601–1626 (1988). [https://doi.org/10.1175/1520-0485\(1988\)018<1601:AGCMFU>2.0.CO;2](https://doi.org/10.1175/1520-0485(1988)018<1601:AGCMFU>2.0.CO;2)
33. Stute, M., Clement, A., Lohmann, G.: Global climate models: Past, present, and future. *Proceedings of the National Academy of Sciences* 98(19), 10529–10530 (2001). <https://doi.org/10.1073/pnas.191366098>
34. Valcke, S.: The OASIS3 coupler: a European climate modelling community software. *Geoscientific Model Development* 6(2), 373–388 (2013). <https://doi.org/10.5194/gmd-6-373-2013>
35. Volodin, E.M., Gritsun, A.S.: Simulation of Possible Future Climate Changes in the 21st Century in the INM-CM5 Climate Model. *Izvestiya, Atmospheric and Oceanic Physics* 56(3), 218–228 (2018). <https://doi.org/10.1134/S0001433820030123>
36. Volodin, E.M., Mortikov, E.V., Kostykin, S.V., *et al.*: Simulation of the modern climate using the INM-CM48 climate model. *Russian Journal of Numerical Analysis and Mathematical Modelling* 33(6), 367–374 (2018). <https://doi.org/10.1515/rnam-2018-0032>