2023, **24** (4), 440–449. doi 10.26089/NumMet.v24r430

doi 10.26089/NumMet.v24r430

Comparison of computational efficiency of two versions of a terrain-following ocean climate model

Dmitry V. Blagodatskikh
Marchuk Institute of Numerical Mathematics RAS, Moscow, Russia
ORCID: 0009-0008-1402-9038, e-mail: blago1958@mail.ru

Abstract: This work presents the results of modification of the ocean climate model of the Institute of Numerical Mathematics (INM) RAS. The main code changes are related to the revision of the dissipation calculation of the horizontal velocity components, isopycnal diffusion and also to modifications of schemes of time integration of scalars and the components of velocity. Test runs conducted with the CORE-II protocol demonstrated that the proposed changes lead to a significant increase in the computational efficiency of the ocean climate model.

Keywords: climate modelling, terrain-following ocean model, parallel scalability.

Acknowledgements: The work was funded under The Russian Federation research and technical development program in ecological strategy and climate change through grant FFMG–2023–0001 "Development of an extended version of the Earth system INM RAS model within a new computational framework"

For citation: D. V. Blagodatskikh, "Comparison of computational efficiency of two versions of a terrain-following ocean climate model," Numerical Methods and Programming. **24** (4), 440–449 (2023). doi 10.26089/NumMet.v24r430.

Сравнение вычислительной эффективности двух версий климатической модели океана с вертикальной сигма-системой координат

Д. В. Благодатских
Институт вычислительной математики РАН имени Г. И. Марчука, Москва, Российская Федерация
ORCID: 0009-0008-1402-9038, e-mail: blago1958@mail.ru

Аннотация: В данной работе представлены результаты модификации климатической модели океана Института Вычислительной Математики (ИВМ) РАН. Описаны основные изменения, произведенные в процедурах расчета диссипации горизонтальных компонент скорости, изопикнической диффузии, а также модификации схем интегрирования скаляров и компонент скорости по времени. На расчетах по протоколу CORE-II продемонстрировано, что внесенные изменения ведут к существенному улучшению вычислительной эффективности климатической модели океана.

Ключевые слова: моделирование климата, сигма-система координат, параллельное масштабирование.

Благодарности: Работа выполнена в рамках тематики "Создание расширенной версии модели Земной системы ИВМ РАН на базе новой вычислительной платформы" (грант FFMG–2023–0001) Федеральной научно-технической программы в области экологического развития Российской Федерации и климатических изменений.

Для цитирования: Благодатских Д.В. "Сравнение вычислительной эффективности двух версий климатической модели океана с вертикальной сигма-системой координат," Вычислительные методы и программирование". 2023. **24**, № 4. 440–449. doi 10.26089/NumMet.v24r430.

2023, **24** (4), 440–449. doi 10.26089/NumMet.v24r430

1. Introduction. The ocean model of the INM RAS has been developed within the framework of the INM RAS Earth System Model (ESM) called the Institute of Numerical Mathematics Climate Model (INMCM) [1]. According to a modern paradigm of climate modelling, a correct representation of the decadal and multidecadal variability of climate requires realistic simulation of the ocean circulation formation and a proper description of its interplay with an atmosphere model [2].

Regretfully, due to the lack of observational data, it is difficult to conduct a rigorous full analysis of the long-term climate modelling to provide its physical correctness. In this regard, the international community of climate modelers resorted to a practice of cross-verification of projections (intercomparison) obtained by various ESMs. Therefore, scenarios of the future climate change are derived as the statistical mean of the results of multiple models. The validity of these scenarios strongly depends on how much the participating intercomparison models differ from each other with regard to the computational algorithms they use [3].

Unlike other ocean climate models [1], the ocean model of INMCM utilizes the σ-vertical coordinate system (terrain-following approach), which is one of the reasons why this model is an attractive tool that can provide results statistically independent from other ESMs.

Thus, INMCM model has been an only participant of the Coupled Model Intercomparison Project on behalf of the Russian Federation including the last two phases 5 and 6 (CMIP5, CMIP6) [4, 5]. The quantitative analysis of simulations carried out in accordance with CMIP6 experimental design showed a good quality of the results obtained by INMCM [6].

Though CMIPs protocols do not hold any compulsory requirements on ESMs degree of complexity, modern trends in climate modelling vividly demonstrate a steady increase in both spatial resolution and the number of ESM components [7]. Hence, the contemporary ESM has a spatial resolution about $1^\circ \times 1^\circ$ of the atmosphere component and $1/4^\circ \times 1/4^\circ$ of the ocean model. As for the number of ESMs components, it has grown from four — atmosphere, land, ocean, sea ice — up to eight components with addition of aerosols, carbon cycle, vegetation, atmospheric chemistry and land ice [8].

Therefore, the ever-growing physical complexity of ESMs makes climate modelling a computationally demanding task. The development of INMCM model lay within the same trend: for instance, an addition of several new software modules (e.g. upper atmosphere chemistry) can be attributed here [9]. Thus, it has become necessary to conduct a thorough profiling of the INMCM model to find the bottlenecks and search for possible replacement or modernization of these parts of the INCM model.

This work is devoted to the consideration of issues related to the ocean component because profiling of other components has already been studied in the article [10]. In order to simplify profiling, test runs are performed within the framework of the Coordinated Ocean-ice Reference Experiments phase II (CORE-II) protocol without adherence to an atmosphere model.

The presentation of this paper is organized in the following way. In section 2 two the main governing equations, physical parameterization, numerical schemes are described. In section 3 the time stepping of the model is given. The results of profiling is represented and the bottlenecks are discussed in section 4. In section 5 modifications of the original model code to deal with the issues revealed in section 3 are described, in section 6 conclusions and plans of future work are given.

2. Model description.

2.1. Governing equations. The dynamical core of the ocean model consists of the solution of the hydrostatic primitive equations, discretized on the C-grid according to Arakawa classification using the vector invariant form under the assumption of a spherical Earth, a thin shell and the Boussinesq approximations [1]:

$$\frac{\partial \boldsymbol{U}_h}{\partial t} = -[(\nabla \times \boldsymbol{U}) \times \boldsymbol{U} + \frac{1}{2}\nabla(\boldsymbol{U}^2)]_h - \ell \boldsymbol{k} \times \boldsymbol{U}_h - \frac{1}{\rho_0}(\nabla p)_h + D^{\boldsymbol{U}}, \tag{1}$$

$$\frac{\partial \theta}{\partial t} = -\nabla(\theta \boldsymbol{U}) + D^\theta + R^\theta, \tag{2}$$

$$\frac{\partial S}{\partial t} = -\nabla(S\boldsymbol{U}) + D^S \tag{3}$$

$$\nabla \boldsymbol{U} = 0, \tag{4}$$

$$\frac{\partial p}{\partial z} = \rho g, \tag{5}$$

$$\rho = \rho(\theta, S, p_0), \tag{6}$$

where $\sigma = z/H$ stands for the general vertical coordinate; x, y are the coordinates in the horizontal plane; $z = z(x, y, \sigma)$ denotes the depth of the point (x, y, σ); $H(x, y)$ is the depth of the ocean in the point (x, y);

2023, **24** (4), 440–449. doi 10.26089/NumMet.v24r430

$\boldsymbol{U} = u\boldsymbol{i} + v\boldsymbol{j} + w\boldsymbol{k}$ represents the velocity vector in the point (x, y, σ); $\boldsymbol{i}, \boldsymbol{j}$ are the orthogonal horizontal vectors parallel to the horizontal surface; $\boldsymbol{k}$ stands for the local downward vertical vector orthogonal to $\boldsymbol{i}, \boldsymbol{j}$; u, v, w are the horizontal and vertical components of the vector $\boldsymbol{U}$; $[\cdot]_h$ denotes the projection operator on the $(\boldsymbol{i}, \boldsymbol{j})$ subspace; θ is the potential temperature; S is the practical salinity; ℓ is the Coriolis parameter; $D^{\boldsymbol{U}}, D^S, D^\theta$ represent the operators of subgrid mixing of the momentum, salinity and heat; p is the anomaly of pressure with respect to p_0; $p_0 = \rho_0 g z$ is the linearized hydrostatic pressure under Boussinesg approximation; ρ_0 denotes the constant reference density; R^θ is the source term of the solar penetrative radiation; g is the gravity acceleration.

The horizontal computational grid is generated using the analytical transformation of the spherical grid [11] into a curvilinear orthogonal grid with the "North pole" displaced into Siberia. The horizontal resolution in basic configuration is $1/2^\circ \times 1/4^\circ$ in longitude and latitude directions respectively.

In the vertical direction the σ-system coordinate is used with 40 levels specially adjusted to be close to 33 standard layers of the World Ocean Atlas [12] in the majority of the World Ocean. The zero-level coincides with the sea surface and the vertical coordinate axis is directed downward.

The advection terms $\nabla(S\boldsymbol{U})$, $\nabla(\theta\boldsymbol{U})$, $\left[(\nabla \times \boldsymbol{U}) \times \boldsymbol{U} + \frac{1}{2}\nabla(\boldsymbol{U}^2)\right]_h$ in the scalar and momentum transport equations (1)–(3) are discretized applying the second order central-difference scheme.

The source term in the momentum equation (1) is a sum of two operators: the vertical turbulent mixing and the lateral dissipation. The latter having no physical meaning is used to dampen non-physical oscillations and make the solution numerically stable. The source terms in scalar transport equations (2)–(3) represent the vertical turbulent mixing and the isopycnal diffusion.

The vertical turbulent mixing coefficients in case of the stable potential density profile are calculated according to Philander-Pacanowsky parameterization [13]. The convection processes are introduced into the system as vertical mixing with large mixing coefficients [1].

The lateral dissipation operator is specified as a 6th-order Laplacian diffusion operator acting along the σ-levels of the ocean model [1]. Thus, the dissipation does not mandatory occur along horizontal surfaces.

2.2. Isopycnal diffusion. In contrast to the lateral dissipation operator, the isopycnal diffusion operator represents a real physical phenomenon. It realizes mixing of scalars along the surfaces tangent at each point to the surface where the locally referenced potential density is constant [14, 15] (the so-called isopycnal or isoneutral surface). In fact, the isopycnal diffusion operator can be considered as a kind of subgrid parameterization of mesoscale mixing in the ocean, which is needed in case of using a coarse horizontal resolution (more than $1/10^\circ \times 1/10^\circ$) [16]. As was mentioned in the introduction the latter is typical for climate modelling.

Originally, at the early stage of development of the ocean model, the discretized isopycnal diffusion operator was realized using the rotated tensor approach: assuming that the isopycnal mixing in the Cartesian coordinate system with the $x'y'$ plane, aligned along the isopycnal surface, is described by the diagonal tensor, we can project it into the xyz-system, where xy-plane is horizontal. Thereby, one can define a tensor (Redi tensor) describing the process of the isopycnal diffusion in the physical space [15, 17]. Further discretization leads to a local computational stencil. In practice, its number of points is limited up to nine [18].

It was shown in [19], that in case of the ocean model with the σ vertical coordinate system (in general, any terrain-following model with a coarse horizontal resolution), the rotated approach leads to appearance of significant spurious modes. If a terrain-following model is considered, then the instability arises due to a large deviation of model levels from a horizontal. This is schematically illustrated in Fig. 1.

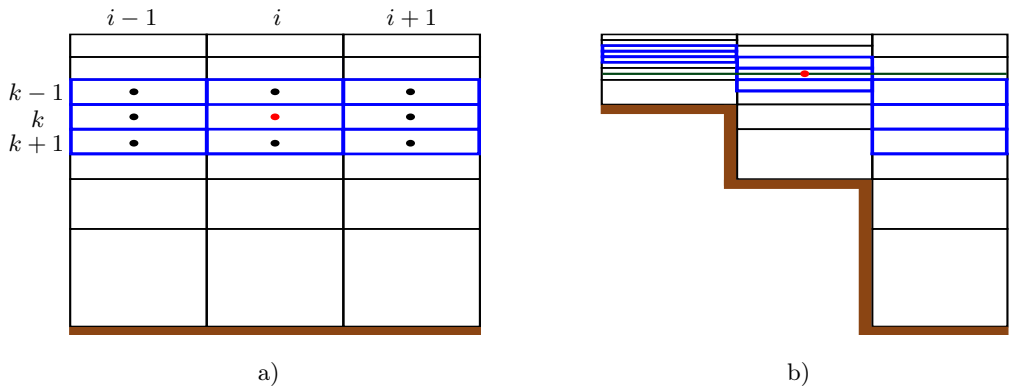

Figure 1. Schematic illustration of a nine-point computational stencil realizing the rotated tensor approach in a) the model space; b) the physical space

2023, **24** (4), 440–449. doi 10.26089/NumMet.v24r430

In Fig. 1a a nine-point computational stencil is shown in the model σ-space, i.e. $\sigma = const$ along any horizontal line. For the sake of simplicity, the number of vertical levels is reduced to seven levels and only a vertical cross-section (x, σ) is shown. There i is the index in x direction and k — in the vertical σ direction. The scalar cells that are part of the stencil are drawn with blue lines, the rest — with black. The thick brown line indicates the ocean bottom. The center of scalar cells are designated by circles. The tendency of scalar in the center cell (red circle) generally depends on the values of scalars in the other cells (black circles) of the stencil.

Fig. 1b shows the same configuration of cells in the physical space. As one can see, the stencil (blue lines) demonstrates severe discontinuities in the physical space in case of a steep ocean shelf. Since the isopycnal surfaces in most regions of the ocean just slightly deviate from horizontal levels, it can lead to the situation when the isopycnal surface (green line) originating from the central point of the stencil (red circle) does not intersect other cells of the stencil (blue lines). Thus, isopycnal mixing occurring across the isopycnal surface cannot be correctly represented within a local stencil ideology [19].

In order to overcome this deficiency, a non-local approach realizing the isopycnal diffusion was proposed in [19]. The key point of this method is the fact that it is not restrained with a local stencil as in the rotated tensor approach. In the non-local method the isopycnal mixing can occur between cells even if they are situated at model levels far beyond the local stencil (Fig. 2). Thus, for a given scalar cell O (red circle, green lines) in the central column, the points P and P' (brown circles) are located in the left and right columns respectively with the same value of the locally referenced potential density as in the center of the cell O. Then, the isopycnal fluxes between three cells (the cells O, A and B drawn with blue lines) in the left column and three cells (O, A', B') in the right column are calculated. The details of the fluxes calculation are given in the paper [19].

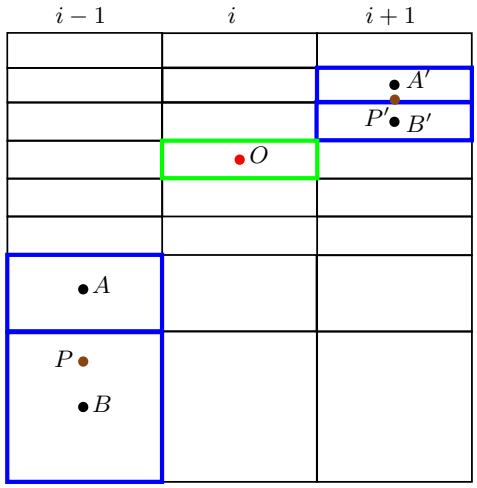

Figure 2. Illustration of the main principle of the non-local approach

The isopycnal diffusion is realized in the original version in routine **isopyc(φ)**, where φ is one of the scalars θ or S. It consists of two logical parts (see Algorithm 1):

- for the current central cell (i, k_O), a point P is searched in the adjacent column with the same locally referenced potential density ρ_{i,k_O} applying à-la bisection method ($n_{\text{iter}} = 6$, $n_z = 40$);
- redistribution of the isopycnal flux between the cells (i, k_O), $(i-1, k_A)$ and $(i-1, k_B)$.

Algorithm 1. Routine **isopyc**(φ)

```
1:  do k = 1, n_z
2:      k_O := k
3:      k_min := 1; k_max := n_z
4:      do k_L = 1, n_iter
5:          k_A := (k_min + k_max) /2
6:          if ρ_{i,k_O} > ρ_{i-1,k_A} then
7:              k_min := k_A
8:          else
9:              k_max := k_A
10:         end if
11:     end do
12:     k_A := k_min
13:     ...
14:     calculation of Δφ, ξ_1 and ξ_2
15:     ...
16:     φ_{i,k_O} := φ_{i,k_O} + Δφ
17:     φ_{i-1,k_A} := φ_{i-1,k_A} + Δφ · ξ_1
18:     φ_{i-1,k_B} := φ_{i-1,k_B} + Δφ · ξ_2
19: end do
```

2023, **24** (4), 440–449. doi 10.26089/NumMet.v24r430

3. Time-stepping. The solution of the discretized governing equations is obtained by splitting into physical processes. There are three main stages:

- advection, vertical mixing of the scalars (θ, S) and the horizontal components of velocity (u, v) plus the isopycnal diffusion for scalars;
- a getting the new baroclinic velocity taking into account the barocline gradient and the Coriolis force term;
- a solving of the matrix equation, where the sea surface height η^{n+1} and the barotropic components of the horizontal velocity $\bar{u}^{n+1}$, $\bar{v}^{n+1}$ are unknown variables.

In the following subsections the algorithmic structure of the original version of the model is given. For brevity, the details of spatial discretization are omitted here because they are thoroughly discussed in the monograph [1]. The subsections are listed in the same order in which the described routines are executed in the model during one time step.

3.1. Sea ice model. The sea ice model presents a traditional approach to ice modelling and consists of the thermodynamics and the ice dynamics modules [1]. In the ice dynamics a set of two-dimensional viscous–plastic model equations [20] is solved on the same grid as the governing equations (1)–(6). The sea ice thermodynamics is based on the zero-dimensional approach [21]. The sea ice model enables to calculate two-dimensional fields of the sea ice compactness and freshwater flux as output data.

3.2. Advection, vertical turbulent mixing, isopycnal diffusion and lateral dissipation. First of all, the tendency of φ (where φ is one of the variables θ, S, u, v) caused by advection is calculated accordingly to the following relations:

$$\begin{aligned} \frac{\varphi^* - \varphi^n}{\Delta t} &= F_{\text{adv}}(\varphi^n), \\ \frac{\varphi^{n+1/2} - \varphi^n}{\Delta t} &= F_{\text{adv}}(\varphi^*). \end{aligned} \tag{7}$$

Here as the time integration method the Matsuno scheme [1] was used; the quantity F_{adv} is the advection term in the equation for scalars (2)–(3) or momentum (1). The advection of scalars is realized in routine **trants_matsuno_unidif**.

Then, for the temperature equation (2) the solar penetration radiation is taken into account by means of solving a local one-dimensional set of equations with help of the tridiagonal matrix algorithm. This procedure is implemented in routine **swradpen**.

At the next stage, a vertical turbulent mixing of φ is conducted (routine **diff_z_impl** for scalars). After that, the isopycnal diffusion of the scalars is calculated in routine **isopyc**. The lateral dissipation of u and v is performed by a six times application of the iso-level Laplacian operator.

Therefore, the advection, the vertical turbulent mixing, the isopycnal diffusion and the solar penetration radiation (for temperature) are realized in the two routines (see Algorithm 2 below):

Algorithm 2. Calculation of scalars θ, S

Temperature, θ	Salinity, S
1: call **trants_matsuno_unidif**(θ)	1: call **trants_matsuno_unidif**(S)
2: call **swradpen**(θ)	2: ...
3: call **diff_z_impl**(θ)	3: call **diff_z_impl**(S)
4: call **isopyc**(θ)	4: call **isopyc**(S)

3.3. Barocline component. The horizontal velocity components $u^{n+1/2}$, $v^{n+1/2}$ are split into the baroclinic u', v' and barotropic $\bar{u}$, $\bar{v}$ components:

$$\begin{aligned} u^{n+1/2} &= \left(u'(x,y,\sigma)\right)^{n+1/2} + \left(\bar{u}(x,y)\right)^{n+1/2}, \\ v^{n+1/2} &= \left(v'(x,y,\sigma)\right)^{n+1/2} + \left(\bar{v}(x,y)\right)^{n+1/2}, \end{aligned} \tag{8}$$

where

$$\bar{u} = \int_0^1 u\,\Delta\sigma; \quad \bar{v} = \int_0^1 v\,\Delta\sigma. \tag{9}$$

At the new time level the baroclinic components are found taking into account the Coriolis force. The

2023, **24** (4), 440–449. doi 10.26089/NumMet.v24r430

Coriolis term is discretized using the technique described in [1]. As a result, the baroclinic components of velocity satisfy the following relations:

$$\begin{aligned} \frac{(u')^{n+1} - (u')^{n+1/2}}{\Delta t} - l\,(v')^{n+1} = 0, \\ \frac{(v')^{n+1} - (v')^{n+1/2}}{\Delta t} + l\,(u')^{n+1} = 0. \end{aligned} \tag{10}$$

Then, the tendency of the horizontal velocity caused by the barocline gradient is calculated.

3.4. Barotropic component. The barotropic $\bar{u}^{n+1}$, $\bar{v}^{n+1}$ components of velocity and the sea surface height η^{n+1} are calculated using the implicit time scheme:

$$\begin{aligned} \frac{\bar{u}^{n+1} - \bar{u}^{n+1/2}}{\Delta t} - l\bar{v}^{n+1} = g\frac{\eta^{n+1} - \eta^n}{\Delta x}, \\ \frac{\bar{v}^{n+1} - \bar{v}^{n+1/2}}{\Delta t} + l\bar{u}^{n+1} = g\frac{\eta^{n+1} - \eta^n}{\Delta y}, \\ \frac{\eta^{n+1} - \eta^n}{\Delta t} = \operatorname{div}(\boldsymbol{u}^{n+1}), \end{aligned} \tag{11}$$

where $\operatorname{div}(\boldsymbol{u})$ is the "horizontal" divergence operator described in [1].

The system (11) can be considered as a matrix equation with a seven-diagonal matrix and solved by the GMRES iterative method from PETSc library [22]. The horizontal components of velocity u^{n+1}, v^{n+1} is calculated by means of summing the baroclinic $(u')^{n+1}$, $(v')^{n+1}$ and the barotropic $\bar{u}^{n+1}$, $\bar{v}^{n+1}$ components as in (8).

4. Performance of the original version. Since in this paper the focus is on the study of the ocean model performance, test runs were conducted according to CORE-II protocol [23]. Within the framework of the CORE-II phase, the atmosphere forcing is determined using the CORE-II IAF atmospheric data sets without adherence to an atmosphere model.

The tests were run at the INM RAS cluster equipped with the 20-core Intel Xeon Gold 6230v2 (two processors per node). In Fig. 3 the dependency of the number of simulated years per day (SYPD) on the number of computational cores used for a test run is shown.

In Fig. 4 the acceleration of the model compared to performance at 34 cores is demonstrated.

One can see from Fig. 3, 4 that the ocean model scalability is limited to approximately 300 cores. The scalability just slightly deviates from the linear one up to 10 SYPD.

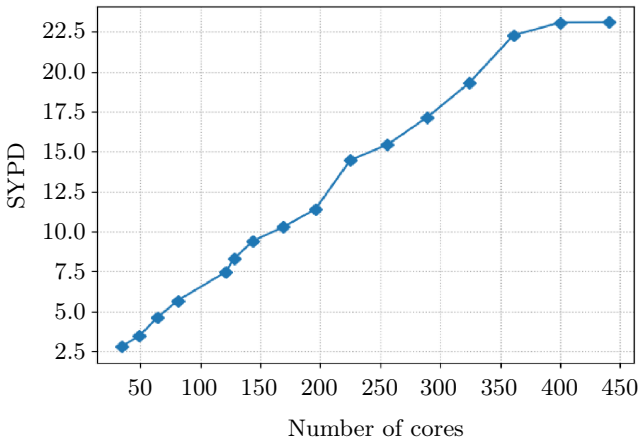

Figure 3. Model throughput curve (in Simulated Years per Day (SYPD))

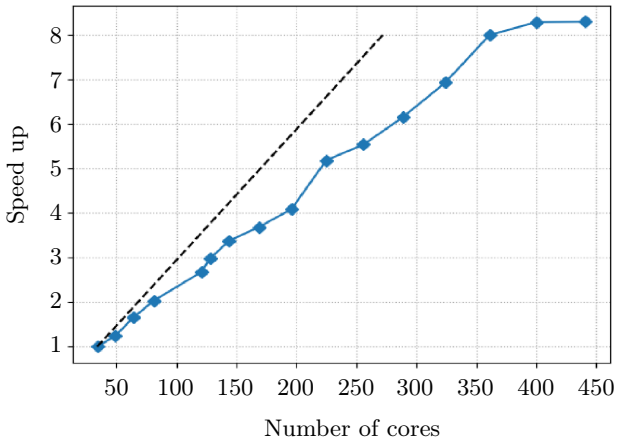

Figure 4. Dependence of the model speedup on the number of computational cores. Dashed line stands for a linear speedup

2023, **24** (4), 440–449. doi 10.26089/NumMet.v24r430

In order to find bottlenecks of the model, the distribution of time consumed by different routines of the model was investigated as well. The result of this study is presented in Fig. 5, where the most computationally consuming parts of the model are shown.

The first column in Fig. 5 stands for a isopycnal diffusion of scalars T and S, the second one corresponds to a time integration by the Matsuno scheme of both scalars and momentum (see (7)), the third column describes a dissipation of the horizontal components of velocity mentioned in Section 2.1, the fourth one meets a solution of the matrix equation (10), the fifth column stands for a vertical turbulent mixing and the sixth one corresponds to the sea ice module.

One can deduce from Fig. 5 that from the viewpoint of size of a time-stepping fraction, the first three parts of the model are distinguished. Thus, the main efforts were dedicated to deal with these issues to accelerate the model.

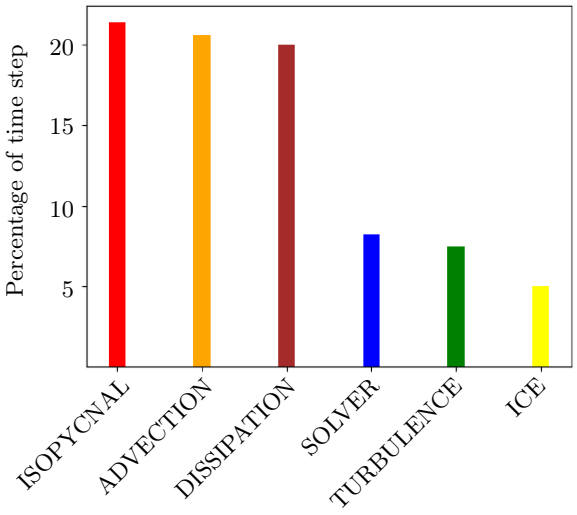

Figure 5. The first six most costly parts of the model

As it can be seen from (7), the advection term is calculated twice for each time step. This problem can be mitigated by application of a multi-step method in which the advection term is taken from the previous time step, e.g. by application of the Adams-Bashforth method.

The 6th order Laplace operator, used in fact as a spatial filter for removing computational noise from the horizontal velocity field, can be successfully replaced by an operator acting along horizontal planes as shown in [24].

5. Performance of the modernized version. There were considered three issues highlighted in the section 3 in the modernized version of the ocean model.

The Matsuno scheme was changed to the second-order Adams-Bashforth method. The 6th-order dissipation operator was replaced by the Laplace operator acting along horizontal planes [24].

For dealing with the issue of the isopycnal diffusion, the code refactoring was made. The update of scalars was merged into one program unit **scalars_transport**, where the routine **isopyc** was called once instead of two times as in the original version (see Algorithm 3).

Algorithm 3. Calculation of scalars θ, S in the modernized version

```
1: subroutine scalars_transport
2:     call trants_matsuno_unidif(θ)
3:     call swradpen(θ)
4:     call diff_z_impl(θ)
5:     call trants_matsuno_unidif(S)
6:     call diff_z_impl(S)
7:     call isopyc(θ,S)
8: end routine
```

The routine realizing the isopycnal diffusion was reprogrammed to take as input parameters two scalar fields (θ and S) instead of just one. In this routine the calculation of fluxes for both scalars was carried out after performing the search algorithm (see Algorithm 4).

Algorithm 4. Routine **isopyc**(θ, S)

```
1: do search for an isopycnal surface
2:     if an isopycnal surface is found then:
3:         ...
4:         calculation of Δθ, ΔS, ξ1 and ξ2
5:         ...
6:         θ_{i,k_O} := θ_{i,k_O} + Δθ
7:         θ_{i-1,k_A} := θ_{i-1,k_A} + Δθ · ξ1
8:         θ_{i-1,k_B} := θ_{i-1,k_B} + Δθ · ξ2
```

2023, **24** (4), 440–449. doi 10.26089/NumMet.v24r430

$S_{i,k_O} := S_{i,k_O} + \Delta S$
$S_{i-1,k_A} := S_{i-1,k_A} + \Delta S \cdot \xi_1$
$S_{i-1,k_B} := S_{i-1,k_B} + \Delta S \cdot \xi_2$
end if
end do

The analogue of the bisection method was changed to the search algorithm taking into account the position of the isopycnal surface that was found previously. The main idea underlying the new search algorithm is illustrated in Fig. 6. The isopycnal surface starts at the center of a scalar cell (black circle) and ends up at the point with the same locally referenced potential density (green circle). Having found the first isopycnal surface (for example, let it starts from A), the densities of the cells lying below the point A in the left column are compared with those of the point B'. If there is no cells that are lighter than the point B' (as it is shown in Fig. 6), then the densities of the cells in the right column lying below B' are compared with those of the point B. If there is a point lighter than B, then an isopycnal surface is defined (the surface starting from C'). Thus, the search is conducted until the bottom has been reached (the surface starting from E').

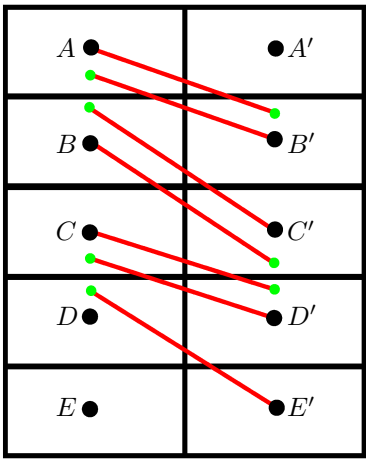

Figure 6. Search for isopycnal surfaces in the new algorithm

An advantage of the modernized version of the considered model compared to the original one, expressed in terms of SYPD, is shown in Fig. 7.

As it follows from Fig. 8, the scalability of the new version is not worsened during the modification process.

6. Conclusions. In this paper, the search for a way to accelerate the climate ocean model was described. The modification of the model leads to almost 40% increase in the model throughput in terms of SYPD. It was achieved by taking a more physically correct momentum dissipation into account and by means of the application of the new time integration scheme, which is less computationally demanding than previous ones. In addition, a refactoring of the code, which realizes the isopycnal diffusion, facilitated to improving the model performance.

At the next stage of development of this model, a marine biochemistry module with about 60 additional passive tracers will be implemented. The results concerning the isopycnal diffusion acceleration and new time integration scheme will be generalized for the case of the marine biochemistry as well.

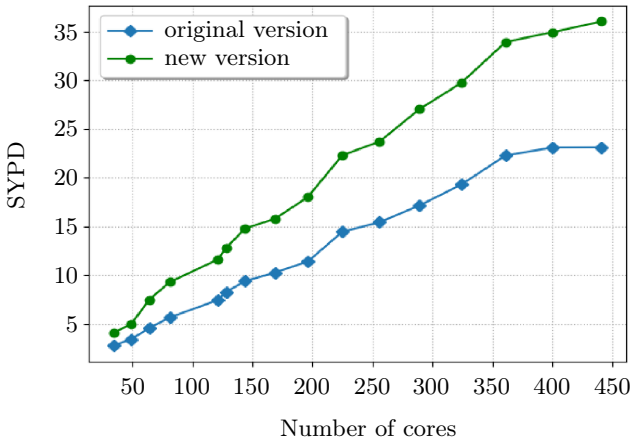

Figure 7. The model throughput curves obtained in the original and the modernized versions

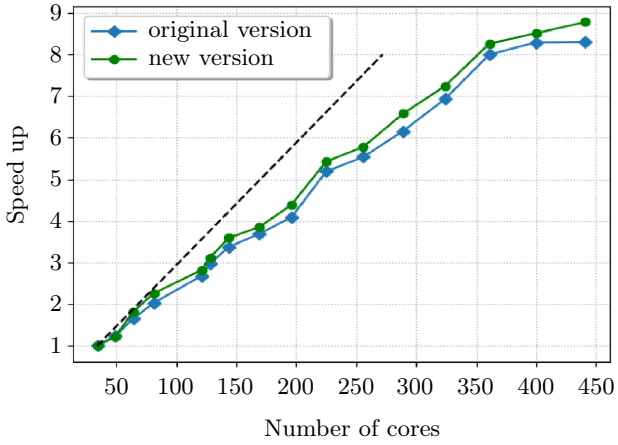

Figure 8. Comparison of speedups obtained in the original and the modernized versions

2023, **24** (4), 440–449. doi 10.26089/NumMet.v24r430

References

1. E. M. Volodin, V. Ya. Galin, A. S. Gritsun, et al., *Mathematical Modeling of the Earth System* (MAKS Press, Moscow, 2016) [in Russian].
2. G. Flato, J. Marotzke, B. Abiodun, et al., *Evaluation of Climate Models* in *Climate Change 2013* (Cambridge Univ. Press, Cambridge, 2014), pp. 741–866. doi 10.1017/CBO9781107415324.020.
3. R. Knutti, D. Masson, and A. Gettelman, "Climate Model Genealogy: Generation CMIP5 and How We Got There," Geophys. Res. Lett. **40** (6), 1194–1199 (2013). doi 10.1002/grl.50256.
4. K. E. Taylor, R. J. Stouffer, and G. A. Meehl, "An Overview of CMIP5 and the Experiment Design," Bull. Am. Meteorol. Soc. **93** (4), 485–498 (2012). doi 10.1175/BAMS-D-11-00094.1.
5. V. Eyring, S. Bony, G. A. Meehl, et al., "Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) Experimental Design and Organization," Geosci. Model Dev. **9** (5), 1937–1958 (2016). doi 10.5194/gmd-9-1937-2016.
6. Y.-H. Kim, S.-K. Min, X. Zhang, et al., "Evaluation of the CMIP6 Multi-Model Ensemble for Climate Extreme Indices," Weather Clim. Extremes **29**, Article Number 100269 (2020). doi 10.1016/j.wace.2020.100269.
7. H. Le Treut, R. Somerville, U. Cubasch, et al., *Historical Overview of Climate Change Science* in *Climate Change 2007* (Cambridge Univ. Press, Cambridge, 2007), pp. 93–127.
8. J. Jebeile and A. Barberousse, "Model Spread and Progress in Climate Modelling," Euro. J. Phil. Sci. **11**, Article Number 66 (2021). doi 10.1007/s13194-021-00387-0.
9. N. E. Chubarova, A. S. Pastukhova, E. Y. Zhdanova, et al., "Effects of Ozone and Clouds on Temporal Variability of Surface UV Radiation and UV Resources over Northern Eurasia Derived from Measurements and Modeling," Atmosphere **11** (1), Article Number 59 (2020). doi 10.3390/atmos11010059.
10. M. Tarasevich, A. Sakhno, D. Blagodatskikh, et al., "Scalability of the INM RAS Earth System Model," accepted in Russian Journal of Numerical Analysis and Mathematical Modelling (2023).
11. J. L. Roberts, P. Heil, R. J. Murray, et al., "Pole Relocation for an Orthogonal Grid: An Analytic Method," Ocean Model. **12** (1–2), 16–31 (2006). doi 10.1016/j.ocemod.2005.03.004.
12. R. A. Locarnini, A. V. Mishonov, J. I. Antonov, et al., *World Ocean Atlas 2009*, Vol. 1: *Temperature* (U.S. Department of Commerce, Washington, D.C., 2010) https://www.ncei.noaa.gov/sites/default/files/2020-04/woa09_vol1_text.pdf. Cited November 23, 2023.
13. R. C. Pacanowski and S. G. H. Philander, "Parametrization of Vertical Mixing in Numerical Models of Tropical Oceans," J. Phys. Oceanogr. **11** (11), 1443—1451 (1981). doi 10.1175/1520-0485(1981)011<1443:POVMIN>2.0.CO;2.
14. T. J. McDougall, "Neutral Surfaces," J. Phys. Oceanogr. **17** (11), 1950–1964 (1987). doi 10.1175/1520-0485(1987)017<1950:NS>2.0.CO;2.
15. T. J. McDougall, S. Groeskamp, and S. M. Griffies, "On Geometrical Aspects of Interior Ocean Mixing," J. Phys. Oceanogr. **44** (8), 2164—2175 (2014). doi 10.1175/JPO-D-13-0270.1.
16. T. L. Delworth, A. Rosati, W. Anderson, et al., "Simulated Climate and Climate Change in the GFDL CM2.5 High-Resolution Coupled Climate Model," J. Clim. **25** (8), 2755—2781 (2012). doi 10.1175/JCLI-D-11-00316.1.
17. M. H. Redi, "Oceanic Isopycnal Mixing by Coordinate Rotation," J. Phys. Oceanogr. **12** (10), 1154—1158 (1982). doi 10.1175/1520-0485(1982)012<1154:OIMBCR>2.0.CO;2.
18. S. M. Griffies, A. Gnanadesikan, R. C. Pacanowski, et al., "Isoneutral Diffusion in a z-Coordinate Ocean Model," J. Phys. Oceanogr. **28** (5), 805—830 (1998). doi 10.1175/1520-0485(1998)028<0805:IDIAZC>2.0.CO;2.
19. D. V. Blagodatskikh, N. G. Iakovlev, E. M. Volodin, and A. S. Gritsun, "Non-Local Discretization of the Isoneutral Diffusion Operator in a Terrain-Following Climate Ocean Model," accepted in Russian Journal of Numerical Analysis and Mathematical Modelling (2023).
20. E. C. Hunke and J. K. Dukowicz, "An Elastic-Viscous-Plastic Model for Sea Ice Dynamics," J. Phys. Oceanogr. **27** (9), 1849—1867 (1997). doi 10.1175/1520-0485(1997)027<1849:AEVPMF>2.0.CO;2.
21. A. J. Semtner, "A Model for the Thermodynamic Growth of Sea Ice in Numerical Investigations of Climate," J. Phys. Oceanogr. **6** (3), 379—389 (1976). doi 10.1175/1520-0485(1976)006<0379:AMFTTG>2.0.CO;2.
22. S. Abhyankar, J. Brown, E. M. Constantinescu, et al., "PETSc/TS: A Modern Scalable ODE/DAE Solver Library," https://arxiv.org/abs/1806.01437. Cited November 23, 2023.
23. S. M. Griffies, A. Biastoch, C. Böning, et al., "Coordinated Ocean–Ice Reference Experiments (CORE s)," Ocean Model. **26** (1–2), 1–46 (2009). doi 10.1016/j.ocemod.2008.08.007.

2023, **24** (4), 440–449. doi 10.26089/NumMet.v24r430

24. M. Gurvan, R. Bourdall'e-Badie, J. Chanut, et al., *NEMO Ocean Engine*, Scientific Notes of Climate Modelling Center, Issue 27, ISSN 1288-1619 (Pierre-Simon Laplace Institute, Guyancourt, 2022). https://zenodo.org/records/6334656. Cited November 23, 2023. doi 10.5281/zenodo.1464816.

Received
September 25, 2023

Accepted for publication
November 17, 2023

Information about the author

Dmitry V. Blagodatskikh — Junior Scientific Researcher; Marchuk Institute of Numerical Mathematics RAS, Gubkina ulitsa, 8, 119333, Moscow, Russia.