https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

swLICOM: the multi-core version of an ocean general circulation model on the new generation Sunway supercomputer and its kilometer-scale application

Kai Xu[1,†], Maoxue Yu[1,†], Jiangfeng Yu[2,5], Jingwei Xie[1], Xiang Han[3], Jiaying Song[6], Mingyao Geng[6], Jinrong Jiang[3,*], Hailong Liu[1,2,*], Pengfei Wang[2,4], and Pengfei Lin[2,5]

[1]Laoshan Laboratory, Qingdao 266237, China
[2]Institute of Atmospheric Physics (IAP), Chinese Academy of Sciences (CAS), Beijing 100029, China
[3]Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China
[4]Center for Monsoon System Research (CMSR), Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100190, China
[5]University of Chinese Academy of Sciences, Beijing 100049, China
[6] Gosci Technology Group,Qingdao 266237, China

Correspondence: Jinrong Jiang (jjr@sccas.cn) and Hailong Liu (lhl@laasg.iap.ac.cn)

Abstract. The global ocean general circulation model (OGCM) with kilometer-scale resolution is of great significance for understanding the climate effects of mesoscale and submesoscale eddies. To address the computational and storage demands of exponential growth associated with kilometer-scale resolution simulation for global OGCMs, we develop an enhanced and deeply optimized OGCM, namely swLICOM, on the new generation Sunway supercomputer. We design a novel split I/O scheme that effectively partitions tripole grid data across processes for reading and writing, resolving the IO bottleneck encountered in kilometer-scale resolution simulation. We also develop a new domain decomposition strategy that removes land points effectively to enhance the simulation capability. In addition, we upgrade the code translation tool swCUDA to convert the LICOM3 CUDA kernels to Sunway kernels efficiently. By further optimization using mixed precision, we achieve a peak performance of 453 Simulated Days per Day (SDPD) with 59% parallel efficiencies at 1 km resolution, scaling up to 25 million cores. The result of simulation with a 2 km horizontal resolution shows swLICOM is capable of capturing the vigorous mesoscale eddies and active submesoscale phenomena.

1 Introduction

Improving the horizontal resolution of global ocean general circulation models (OGCMs) to the kilometer scale has been a cutting-edge direction in current physical oceanography research. The core goal is to resolve the nonlinear interactions of

†Authors contributed equally to the paper.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

oceanic multi-scale dynamical processes. Though traditional high-resolution models with ~10 km horizontal resolution can capture mesoscale eddy activities (Hallberg, 2013; Kjellsson and Zanna, 2017; Hewitt et al., 2020), they fail to directly resolve submesoscale processes with horizontal scales of 100 m-10 km (Gula et al., 2022; Xie et al., 2024), which contribute to key mechanisms such as energy cascade, vertical mixing, and material transport (Zhang et al., 2023; Chassignet and Xu, 2017; Schubert et al., 2019; Balwada et al., 2018; Soufflet et al., 2016). In recent years, with the breakthrough of supercomputing technology, global OGCMs with kilometer-scale resolution have been gradually realized. For example, the 1/48° MITgcm LLC4320 simulation (Su et al., 2018), the ocean component of ICON-Sapphire with a 1.25 km resolution (Hohenegger et al., 2023), and the 1/100° global model LICOMK++ (Wei et al., 2024; Xie et al., 2025). Such OGCMs show great capability on simulating submesoscale eddies and the related multi-scale interactions. However, limited by computing resources and storage costs, long-term integration (such as on the interannual scale) of kilometer-scale global OGCMs still faces challenges.

The rapid development of the heterogeneous supercomputers provides potential opportunities to solve the problem. The development of new climate and ocean models on heterogeneous supercomputers has advanced rapidly, demonstrating significant performance (Eirund et al., 2024; Silvestri et al., 2024), particularly in the GPU or DCU architectures. Fuhrer et al. rewrote the dynamical framework of the COSMO model, which implements solutions to the non-hydrostatic Euler equations, from Fortran to C++ (Fuhrer et al., 2018; Thaler et al., 2019). They developed the Stencil Loop Language (STELLA), a domain-specific programming C++ templates-based language (Gysi et al., 2015). STELLA provides performance-portable optimizations for stencil computations and can track data access patterns and kernel data dependencies. Although STELLA simplifies code porting, COSMO conversion still requires a significant amount of hand-written code. POM.gpu (Xu et al., 2015), a GPU version of mpiPOM, converts Fortran code to CUDA C and enhances its performance through loop and function fusion, better use of read-only data cache and L1 cache, multi-GPUs communication optimization, and overlapping I/O and computation. Kinaco focused on improving the sparse matrix-vector multiplication of its MGCG solver on GPUs, primarily optimizing data access and employing instruction-level parallelism and mixed precision computing (Yamagishi and Matsumura, 2016). For the porting of Veros to GPUs, Python with JAX backend was used (Häfner et al., 2021). PETSc and petsc4py were used to support distributed linear solvers. Veros provided a hand-written implementation of the Thomas algorithm, written in Cython for CPUs and CUDA for GPUs, to enhance efficiency. Oceananigans is the first ocean model designed specifically for GPUs rather than being ported from existing CPU code (Silvestri et al., 2023). Oceananigans utilized Oceananigans.jl, a new software written in the Julia programming language. It leverages Just-In-Time (JIT) compilation and LLVM to implement optimizations, including kernel fusion, on-the-fly computation for intermediate quantities, and overlapping computation and communication. In addition, there have been efforts to optimize POP on accelerators like Intel Xeon Phi. Key techniques include explicit and implicit fusion of data locality and vectorization, along with asynchronous offloading strategies for scheduling computations asynchronously (Aketh et al., 2016).

On the other hand, as another important heterogeneous supercomputer, significant model development of kilometer-scale resolutions has been achieved on Sunway architecture processors. Xu et al. developed SWSLL, a domain-specific language for stencil computations optimization of WRF's dynamical framework (Xu et al., 2019). SWSLL automates loop fusion and dependency analysis and partitions computations based on memory access counts, guided by the Roofline model, indicating memory-bound computations for stencils. CESM (Community Earth System Model) is ported to Sunway by manual kernel coding for the dynamical framework and OpenACC for physical process optimization (Zhang et al., 2020b). Ye et al. developed swNEMO, which implements simulation with a resolution of 1 km, achieving a performance of 1.43 SYPD (Ye et al., 2022).

In this study, we redesign and develop the LASG/IAP Climate system Ocean Model version 3.0 (LICOM3; Li et al., 2020; Lin et al., 2020) on the new generation Sunway supercomputer. LICOM3 was developed by the State Key Laboratory of Numerical Modeling for Atmospheric Sciences and Geophysical Fluid Dynamics (LASG), Institute of Atmospheric Physics (IAP), Chinese Academy of Sciences. It has been optimized on several heterogeneous supercomputers, including LICOM2-gpu (Jiang et al., 2019), LICOM3-HIP (Wang et al., 2021), and LICOM3-CUDA (Wei et al., 2023). LICOM2-GPU utilized OpenACC to port LICOM2 to GPUs, employing techniques such as local memory blocking, loop fusion, and communication optimization. LICOM3-HIP utilized AMD's HIP interface for AMD GPU portability, facilitating code conversion and execution on both NVIDIA and AMD GPUs. LICOM3-CUDA, a CUDA C version of LICOM3, enhanced fine-grained parallelism, decoupled data dependencies, and prevented memory write conflicts, particularly optimizing heterogeneous architecture communication. LICOM3-CUDA outperforms LICOM3-HIP, making it the preferred choice for LICOM3 porting to the Sunway platform. The latest achievement is from our previous work named LICOMK++, a performance-portable OGCM that uses Kokkos. It achieves 1.05 SYPD performance for 1 km resolution on the new generation Sunway supercomputer (Wei et al., 2024).In this study, we explore new automatic code transformation methods beyond Kokkos and achieve better performance metrics than Kokkos for LICOM3 at the kilometer-scale resolution, contributing to ocean model development.

To support the kilometer-scale resolution simulation of LICOM3 and leverage the performance advantages of heterogeneous computing, we redesign and optimize LICOM3 on the new-generation Sunway supercomputer, naming the optimized model swLICOM. Section 2 introduces the new-generation Sunway supercomputer and the LICOM3 model. Our innovative methods and optimizations are detailed in Section 3. In Section 4, we evaluate the performance and scalability of swLICOM and present the ultra-high-resolution simulation results.

2 Model Description and HPC

2.1 The Ocean General Circulation Model

LICOM3 is a global OGCM to simulate the oceanic physical state and general circulation. It also serves as the ocean component of coupled climate models, such as the Flexible Global Ocean Atmosphere-Land System model (FGOALS; Bao et al., 2013; Li et al., 2013) and the Chinese Academy of Sciences Earth System Model (CAS-ESM; Zhang et al., 2020). In addition, LICOM3 has participated in the Ocean Model Intercomparison Project Phases 1 and 2 (OMIP1 and OMIP2, respectively; Lin et al., 2020; Tsujino et al., 2020).

LICOM3 adopts the Arakawa-B grid horizontally and eta coordinates vertically (Mesinger et al., 1988) with free sea surface height. It utilizes orthogonal curvilinear coordinates and the tripolar grid (Smith et al., 1992, 2010). Its dynamical core also employs a two-step shape-preserving advection scheme and a semi-implicit mode-splitting (barotropic-baroclinic) time-stepping scheme.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

If the coordinate system (ξ_1,ξ_2,ξ_3) is an orthogonal curvilinear coordinate system on a spherical surface and we only consider the changes in the horizontal grid coordinates on the sphere's surface while keeping the vertical coordinate system unchanged, the model primitive equations in any orthogonal curvilinear coordinate system are given by:

$$\begin{cases} \frac{\partial u_x}{\partial t} + L(u_x) = -\frac{1}{\rho_0 h_1}\frac{\partial p}{\partial \xi_1} - f^* u_y + F_{mx} + \frac{\partial}{\partial \xi_3}\left(K_m \frac{\partial u_x}{\partial \xi_3}\right), \\ \frac{\partial u_y}{\partial t} + L(u_y) = -\frac{1}{\rho_0 h_2}\frac{\partial p}{\partial \xi_2} + f^* u_x + F_{my} + \frac{\partial}{\partial \xi_3}\left(K_m \frac{\partial u_y}{\partial \xi_3}\right), \\ \frac{\partial p}{\partial z} = -\rho g, \\ \frac{1}{h_1 h_2}\left(\frac{\partial h_2 u_x}{\partial \xi_1} + \frac{\partial h_1 u_y}{\partial \xi_2}\right) + \frac{\partial w}{\partial \xi_3} = 0, \\ \frac{\partial T}{\partial t} + L(T) = A_T \nabla^2 T + \frac{\partial}{\partial \xi_3}\left(K_T \frac{\partial T}{\partial \xi_3}\right) + C_T + Q_{\text{pen}}, \\ \frac{\partial S}{\partial t} + L(S) = A_S \nabla^2 S + \frac{\partial}{\partial \xi_3}\left(K_S \frac{\partial T}{\partial \xi_3}\right) + C_S, \\ \rho = \rho(T,S,p), \end{cases} \tag{1}$$

where θ is the colatitude, λ is the longitude, T and S are temperature and salinity respectively, ρ is the in-situ density, p is the pressure, f^* is the Coriolis parameter, Q_{pen} is the shortwave radiation penetration term, K_m is the vertical turbulent viscosity coefficient for velocity, K_T and K_S are the vertical turbulent diffusion coefficients for temperature and salinity respectively, A_T and A_S are the horizontal turbulent diffusion coefficients for temperature and salinity respectively, C_T and C_S are the vertical advection terms, F_{mx} and F_{my} are the horizontal turbulent viscosity terms, ∇^2 is the horizontal Laplacian operator, L is the advection operator, u_x and u_y are the horizontal velocities in the directions of the orthogonal curvilinear basis vectors along the spherical surface, where h_i is known as the Lamé coefficient. The physical meaning of the Lamé coefficients is the length of the unit basis vectors in an orthogonal curvilinear coordinate system when projected onto a Cartesian coordinate system, which represents the true length in physical space. Therefore, the Lamé coefficients are also known as map scale factors. The horizontal viscosity terms in the momentum equations can be written as (Griffies, 2009):

$$F_{mx} = A_m\left(\nabla^2 u_x - u_x\left(\frac{\partial K_x}{h_1\partial\xi_1} + \frac{\partial K_y}{h_2\partial\xi_2} + 2K_x^2 + 2K_y^2\right) + u_y\left(\frac{\partial K_x}{h_2\partial\xi_2} - \frac{\partial K_y}{h_1\partial\xi_1}\right) + 2K_y\left(\frac{\partial u_y}{h_1\partial\xi_1}\right) - 2K_x\left(\frac{\partial u_y}{h_2\partial\xi_2}\right)\right), \tag{2}$$

$$F_{my} = A_m\left(\nabla^2 u_y - u_y\left(\frac{\partial K_x}{h_1\partial\xi_1} + \frac{\partial K_y}{h_2\partial\xi_2} + 2K_x^2 + 2K_y^2\right) + u_x\left(\frac{\partial K_y}{h_1\partial\xi_1} - \frac{\partial K_x}{h_2\partial\xi_2}\right) + 2K_x\left(\frac{\partial u_x}{h_2\partial\xi_2}\right) - 2K_y\left(\frac{\partial u_x}{h_1\partial\xi_1}\right)\right), \tag{3}$$

where A_m is the horizontal turbulent viscosity coefficient, $K_x \equiv \frac{1}{h_1 h_2}\frac{\partial h_2}{\partial \xi_1}$, $K_y \equiv \frac{1}{h_1 h_2}\frac{\partial h_1}{\partial \xi_2}$.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

Except for the Laplace term in equations (2) and (3), all other terms are geometric curvature terms, which arise due to the non-uniformity of the coordinate grid in physical space. Additionally, under any orthogonal curvilinear coordinates, the expressions for the advection operator and the two-dimensional Laplacian operator are as follows:

$$\begin{cases} L = u_x \dfrac{\partial}{h_1 \partial \xi_1} + u_y \dfrac{\partial}{h_2 \partial \xi_2} + w \dfrac{\partial}{\partial \xi_3}, \\ \nabla^2 = \dfrac{1}{h_1 h_2} \left(\dfrac{\partial}{\partial \xi_1} \left(h_2 \dfrac{\partial}{h_1 \partial \xi_1} \right) + \dfrac{\partial}{\partial \xi_2} \left(h_1 \dfrac{\partial}{h_2 \partial \xi_2} \right) \right). \end{cases} \tag{4}$$

Equations (1) to (4) represent a system of differential equations in arbitrary orthogonal curvilinear coordinates (Y.Q. et al., 2018).

LICOM3 essentially solves the equations described above. The main procedure for solving these equations is illustrated in Figure 1. Seven compute-intensive modules operate within the time integration loop: “readyt”, “readyc”, “barotr”, “bclinc”, “tracer”, “icesnow”, and “convadj”. These modules are the primary targets for optimization. The first two modules compute terms in the barotropic and baroclinic equations. The following three modules ("barotr," "bclinc," and "tracer") are responsible for solving the barotropic, baroclinic, and temperature/salinity equations. The prescribed data are used for ice module. The "icesnow" module has been simplified to only handle temperatures higher or lower than the freezing point of the seawater, usually using -1.8C in the model. The "convadj" module is the convective adjustment, which ensures that the density of the lower layer of seawater is greater than that of the upper layer.

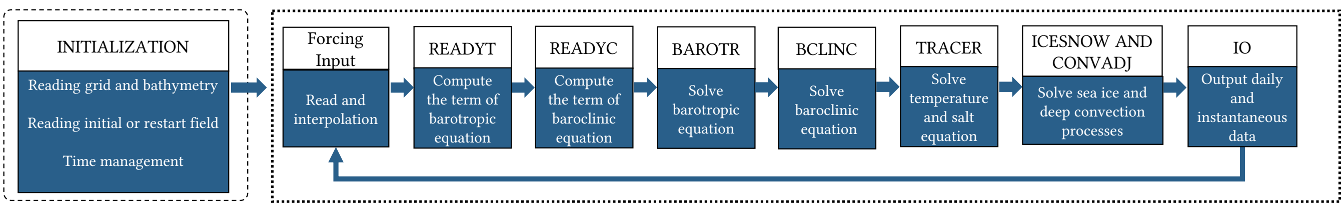


Figure 1. The overview of LICOM3 architecture. The INITIALIZATION module is responsible for model initial preparation, including reading grid and bathymetry file, reading initial or restart field, and time management. The main procedure includes seven main compute-intensive modules and IO module: READYT for computing terms of the barotropic equation, READYC for computing terms of the baroclinic equation, BAROTR for solving the barotropic equations, BCLINC for solving the baroclinic equations, TRACER for solving the tracer (temperature and salinity) equations, ICESNOW and CONVADJ for solving sea ice and deep convection process, IO module for outputting daily and instantaneous data.

2.2 SW26010 Pro

SW26010 Pro is a new generation of Sunway heterogeneous many-core processors developed by China. It builds upon the basic architecture of the SW26010 and serves as an improved version of the processor featured in the Sunway TaihuLight supercomputer.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

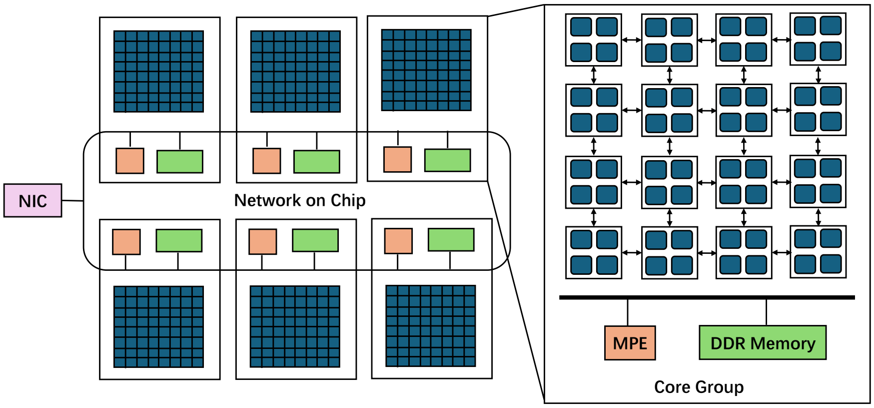


Figure 2. The architecture of SW26010 Pro many-core processor. Each SW26010 Pro processor contains 6 Core Groups (CGs). Within each CG, there are one management processing element (MPE) and 64 computing processing elements (CPEs) organized in an 8 × 8 cluster and 16 GB of attached DDR4 main memory. These 6 CGs and network interface card (NIC) are connected via the network on chip (NoC).

The architecture of SW26010 Pro is shown in Figure 2. The SW26010 Pro is divided into six core clusters, known as Core Groups (CGs). Each CG consists of one Management Processing Element (MPE) and 64 Computing Processing Elements (CPEs). MPE is responsible for handling communication, IO, and spawning threads on CPEs. CPEs in each CG are organized as an 8x8 mesh and communicate with each other using Remote Memory Access (RMA). The MPE features a 32KB L1 Instruction cache, 32KB L1 Data Cache, and 512KB L2 cache. Each CPE has a 32KB L1 Instruction cache and a 256 KB software-managed scratchpad called local data memory (LDM) with up to half configurable as cache. Each CG has its own memory controller and a 16GB memory capacity, providing the SW26010 Pro chip access to a total of 96GB of main memory. Each CG has its own memory controller and has 16GB of memory capacity. In total, a SW26010 Pro chip can access 96GB of main memory. CPEs can access main memory through Load/Store instructions or direct memory access (DMA). These 6 CGs and network interface cards (NICs) are connected via the network on-chip (NoC). Each node contains a single chip with its own network connectivity. A Supernode connects 256 nodes via a high-speed switch. Each Supernode is further connected to a cross-supercomputer interconnect with 48 ports. Compared to traditional multi-core architectures, the heterogeneous many-core architecture poses more complex programming challenges, necessitating significant efforts to redesign and optimize applications to harness its full computational potential.

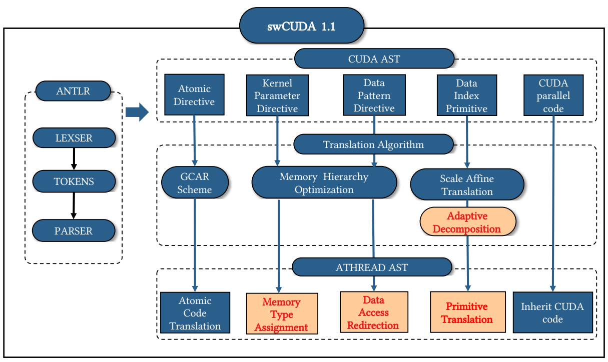
3 Method and Optimzations

3.1 swCUDA

LICOM3 was ported to different supercomputer clusters in recent years (Wang et al., 2021). Every time, porting LICOM3 and getting superior performance takes a huge effort. In the GPU version of LICOM3, there are a total of more than 70 kernels that are manually ported and optimized. To address this issue, we adopt an auto-parallel code translation tool, swCUDA,

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

to automatically translate the mature CUDA kernel of LICOM GPU version to Sunway Athread kernel from our previous work (Yu et al., 2024). swCUDA constructs a hardware mapping model and a scaled affine algorithm between GPU and Sunway CPE. Meanwhile, a set of high-level directives is designed, which is easy to get started. For general usage, only the attribute and pattern of data are required to be described. Considering the characteristics of LICOM3 kernels with general stencil computation, We optimized the swCUDA to version 1.1, automatically achieving 3D grid decomposition and memory optimization. This upgrade gives translated kernels better parallel scalability. swCUDA 1.1 supports various memory transfers to accomplish auto-translation based on the high-level directive. The architecture of swCUDA 1.1 is shown in Figure 3a. Firstly, high-level directives are embedded to illustrate the attributes and patterns of the data in the CUDA kernel, as shown in Figure 3b. Secondly, the code generator ANTLR (Parr et al., 2014) is utilized to parse the CUDA kernel to an abstract syntax tree (AST). Thirdly, our translation algorithms are utilized in corresponding AST nodes to accomplish CUDA primitive translation and memory redirection. By swCUDA 1.1, the Sunway kernel is translated not only with high efficiency but also with high performance.



(a) The workflow of swCUDA 1.1.

```
//kernel parameter directive
userDef paraVar(double,h0,sizeof(double)*jmt*imt,outonly,dma,j*imt+i);
userDef paraVar(double,h0p,sizeof(double)*jmt*imt,readonly,cache,j*imt+i);
userDef paraVar(double,work,sizeof(double)*jmt*imt,readonly,cache,j*imt+i);
//data redirection directive
dataPattern(blockIdx.x * blockDim.x + threadIdx.x,
                blockIdx.y * blockDim.y + threadIdx.y, NONE);
//CUDA Kernel Funciton
__global__ void barotr2_cu1(double *h0, double *h0p, double *work){
  int j = blockIdx.x * blockDim.x + threadIdx.x;
  int i = blockIdx.y * blockDim.y + threadIdx.y;
  if (i < imt && j < jmt){
    h0[j * imt + i] = h0p[j * imt + i]+ work[j * imt + i] * dtb;
  }
}
```

(b) An example of swCUDA 1.1 usage.

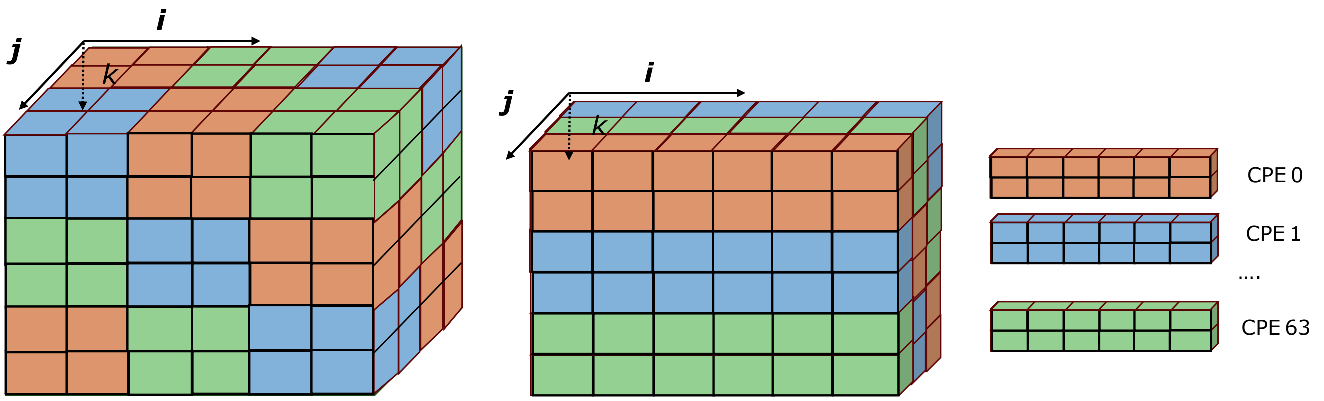
Figure 3. swCUDA 1.1 overview. swCUDA 1.1 automatically translates the CUDA kernel to the Athread kernel by using high-level directives. (a) swCUDA adopts a hardware mapping model and scaled affine algorithm between GPU and SW26010 Pro. ANTLR is used to parse CUDA kernels to CUDA AST. Our algorithm translates the CUDA AST to the Athread AST and generates the ATHEAD kernel. We promote adaptive decomposition and memory accessing optimization in swCUDA 1.1. (b) Kernel parameter directive `userDef paraVar` and data redirection directive `dataPattern` are specified to implement memory accessing optimization in translation.

3.1.1 Adaptive Decomposition

LICOM3 uses finite-difference approximation to solve the Navier–Stokes equations, the general stencil computation in a multidimensional grid. In the GPU version of Licom3, the grid is decomposed into two or three-dimensional thread blocks. Each thread block has fixed threads depending on the hardware limit, which is divided as a cube, as shown in Figure 4a. However, this decomposition is not adapted to the Sunway platform since it conducts data swapping in the cache to increase the cache miss rate. Hence, we designed an integrated JK decomposition to improve Sunway parallel efficiency, as shown in Figure 4b.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

By this decomposition, the whole grid is divided into multiple i-directional blocks. With data stride, blocks with the same color are sent to the same CPE. Data access in each CPE is continuous, which is the most effective cache utilization.



(a) 3D block decomposition for CUDA kernel.

(b) Integrated JK decomposition for Sunway Athread kernel.

Figure 4. The decomposition translation between CUDA on NVIDIA GPU and Athread on Sunway processor. (a) Three dimensions of a 3D block decomposition for the CUDA kernel. (b) An integrated JK decomposition of a 3D block data decomposition for the Athread kernel.

To automatically translate GPU decomposition to Sunway-integrated JK decomposition, we develop an adaptive decomposition method in swCUDA. In the original CUDA thread hierarchy, there is a limit to the number of threads per block, which may contain up to 1024 threads. Here, we break through this limit by directly assigning i-dimensional data to thread blocks as one-dimensional blocks and decomposing the grid into j and k-dimensional blocks, automatically according to different parallel scales. Each CPE is assigned with continuous blocks for load balance. Each block occupies continuous i-directional data for access and calculation. Based on this translation, the original CUDA parallel algorithm can be inherited in Sunway kernels and achieves the most optimal performance.

3.1.2 Memory Accessing Optimization

In the previous swCUDA, we designed a general memory optimization strategy. It provides a flexible memory type interface, which is automatically distributed according to the variable memory size. However, the variable memory size of LICOM3 is changed in different parallel scales. Meanwhile, many stencil kernel computations require access to data from its surrounding grids with three dimensions in LICOM3, which means discrete memory access is a general characteristic. In Sunway CPE, slave L1 cache direct access is suitable for continuous data access, and discrete memory access should be leveraged by DMA transfer. Selecting appropriate memory usage is essential to promote kernel performance. Hence, we redesign our high-level directive to explicitly illustrate the data memory usage and localization of data indexing for specific data arrays by the format $paraVarAttr(type, varname, size, inout, memUsage, glbInx)$, as shown in Figure 3b. Each data array has a detailed attribute. $Inout$ the attribute is used to illustrate if this array is read-only or must be modified in the kernel. $memUsage$ is used to describe if it uses cache or DMA transfer. $glbInx$ is required to combine with $dataPattern$ for usage, which represents

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

the global thread indexing pattern for i, j, and k directions in the CUDA kernel. $glbInx$ denotes the global indexing pattern of the current array. If the array uses DMA transfer, swCUDA translates the global indexing to local indexing using our new JK decomposition algorithm. Otherwise, swCUDA directly translates its index to global indexing. Based on this new strategy, most kernels perform better due to the appropriate memory usage.

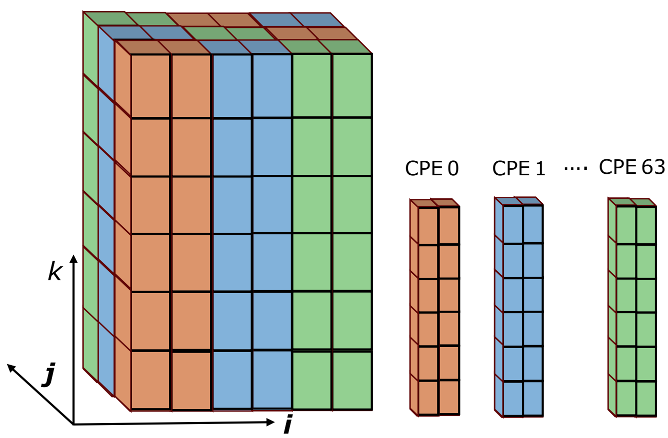
3.1.3 Manually Optimization

In CUDA kernels, stencil computation is not concerned with the order of the three-dimensional loop since all data has been loaded into the GPU device memory. However, Sunway kernels are very sensitive to the k-loop position. If the k loop is in the innermost layer, the efficiency of the Sunway kernel is worse; even the method of using DMA transfer can't address this issue. In this case, we need to manually optimize the kernels. There are two ways. One is to extract the k loop to the outer based on trading space for time. Another method is to decompose data by IK decomposition, as shown in Figure 5. From our example in Figure 5a, WKK is extended to a three-dimensional data array $wktmp$ to record data in every grid point, which can extract k-loop to outer. Then, the Sunway kernel can use JK decomposition with continuous data access. As shown in Figure 5b, data is decomposed from the i- and k-directions. Each CPE is assigned part of the i-directional and total k-directional data using stride DMA transfer. Data is reorganized and loaded to Sunway CPE with the continuous array. Hence, data indexing is required for redirection. The manual optimizations are used in several kernels to improve the performance significantly.

```
DO J = 1,jmt
 DO I = 1,IMT
    WKK (1)= (PSA (I,J)* OD0+ WORK (I,J)* G)* VIT (I,J,1)
    DO K = 1,KM
       WKK (K +1)= WKK (K) - GG (I,J,K)* DZP (K)* VIT (I,J,K)
    END DO
    DO K = 1,KM
       WKA (I,J,K)= P5* (WKK (K) + WKK (K +1))
    END DO
 END DO
END DO
```

```
DO K = 2,KM+1
DO J = 1,jmt
  DO I = 1,IMT
    if(K .eq. 2) then
      wktmp(i,j,1)= (PSA (I,J)* OD0+ WORK (I,J)* G)* VIT (I,J,1)
    endif
    wktmp(i,j,k)= wktmp(i,j,k-1) - GG (I,J,K-1)* DZP (K-1)* VIT (I,J,K-1)
    WKA (I,J,K-1)= P5* (wktmp(i,j,k-1) + wktmp(i,j,k))
  END DO
END DO
END DO
```

(a) An example of extracting k loop to outer.

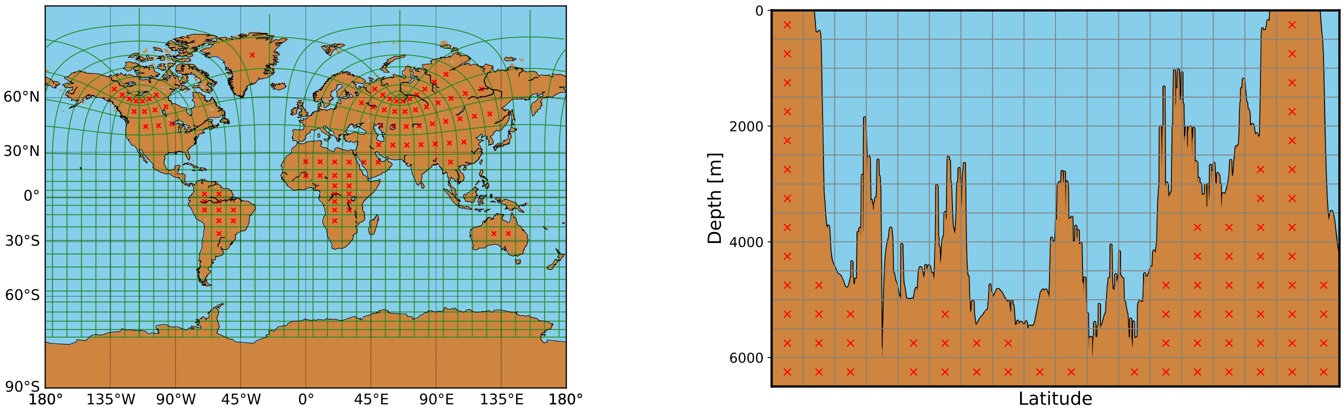


(b) The IJ directional decomposition.

Figure 5. Manual optimization is needed to change the loop order from jik to kji. (a) An example of extracting the k loop to the outer. The upper code snippet is the original stencil computation with k in the inner loop, and the lower one is the optimized code, extracting the k loop to the outer based on trading space for time. (b) Part of the i and j directional data is assigned and loaded to each CPE.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

3.2 Remove Land Grid Points



(a) 3D block decomposition for CUDA kernel.

(b) Integrated JK decomposition for Sunway Athread kernel.

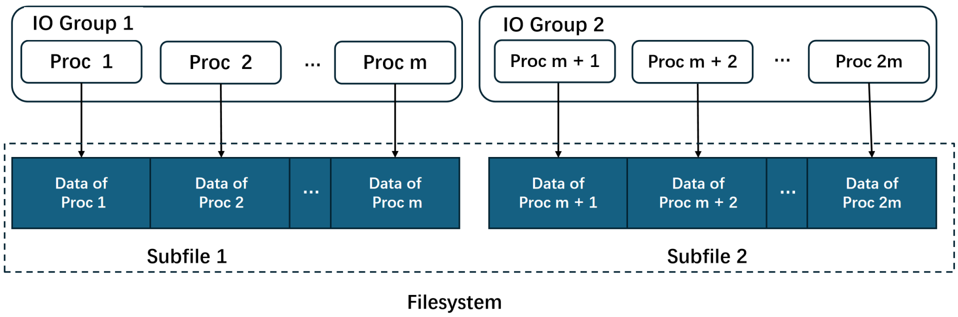
Figure 6. Manual optimization for removing land grid points. (a) The MPI process decomposition of the LICOM3 tripole grid (green lines). Each block contains many grid points for both ocean and land, and the blocks are labeled by a red x, indicating that the block contains only land grids. (b)The grid points in one block are ocean or land in the vertical dimension. The land points in one block are labeled by a red x.

To accelerate computations and reduce resource usage, we optimize the LICOM grid partition. The total land area globally accounts for 29% of the Earth's total surface area. When partitioning the Earth's surface into grids, the number of grid points on the land represents approximately 29% of all grid points. However, land points have no effect on marine points and do not contribute to the final results. Therefore, we design an algorithm to remove land grid points to reduce resource usage and improve computing efficiency. First, the input data is partitioned according to the number of processes. Each process holds one block of data, and each block holds tens of grid points. Then, we count the marine grid points of each block. If no marine grid points exist in one block, we will remove the process corresponding to the block. A pre-processing procedure is implemented to tackle the problem. During the actual computation, we will launch the processes containing marine points. For accurate calculations, we need to map the process rank that only includes marine points to the original process rank that did not exclude land points. In this way, we can correctly read and write the data. At the same time, we will regenerate a new communication topology based on the new process rank to facilitate the exchange of boundary data. Through this approach, we can reduce 29% in computational resources while ensuring that the results are completely consistent with the original results, thus increasing computational speed and resource efficiency at high resolution.

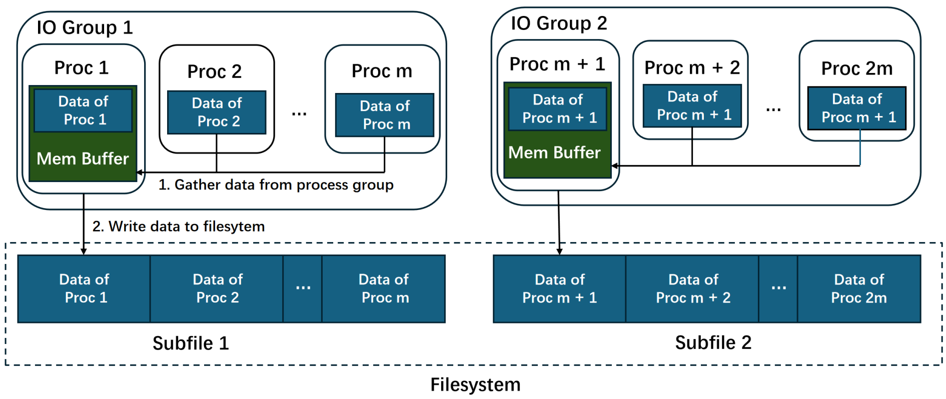
3.3 IO Optimizations

LICOM utilizes the NetCDF format to read input, write model output, and restart files. Initially, data are read by a single process, which then distributes them among others. Similarly, output data from multiple processes is gathered by one process and written to the file system. However, as the number of processes increased to hundreds of thousands, input reading and

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025



(a) Read data.



(b) Write data.

Figure 7. IO strategy. The big files have been separated into small subfiles. All MPI processes are divided into IO process groups that read from or write to a subfile. (a) Read data. Processes read their data based on their address offset in the subfile. (b) Write data. The master process in the IO process group first gathers data from other processes. Then, the master process writes data to a subfile.

output writing performance became inadequate. To address this issue, we adopted a strategy that divides the data into multiple subfiles. To minimize the number of subfiles, each set of one hundred processes handles input reading from a single subfile, and similarly, one hundred processes collectively write model output and restart files to a single file. Furthermore, we opted for a binary format for both input and output files. Compared to the original NetCDF-based implementation, this optimized approach has significantly improved IO performance across tens of thousands of machines.

For analysis and archival storage purposes, a post-processing step involves several nodes that convert the binary format data into the NetCDF format.

3.4 Mixed Precision

The ocean model default uses double resolution to guarantee the simulation accuracy, regardless of the need. The computational cost and memory bandwidth increase exponentially along with the in-depth application of kilometer resolution in the ocean

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

model. The approach of reducing numerical precision has been proven to be an effective strategy to improve performance (Tintó Prims et al., 2019; Chen et al., 2024). We implement single precision mainly focusing on equation **??** and equation **??** of LICOM3, making a balance of performance improvement and fidelity.

3.4.1 Selecting Variable of Reduced Precision

In swLICOM, we divide the set of variables into three groups: dynamic core, tracer formulation, and physical package. The dynamic core mainly consists of barotropic and baroclinic processes. The tracer primarily calculates temperature and salinity equations. The physical package mainly includes vertical viscosity and diffusivity schemes (Canuto et al., 2002). Most of the variables in these three groups are relatively independent. For these three groups, the tracer formulation occupies almost 50% of the computational cost (Wang et al., 2021), Canuto parametrization only calculates vertical diffusivities for momentum, heat, and salt. Hence, our reduced precision is aimed at the variables of tracer formulation and physical package, which occupy more than 75% amount of the total variables. Furthermore, Canuto parametrization is heavily computing-intensive and load-unbalanced. In this case, implementing reduced precision can significantly promote its performance. On the other hand, advection momentum calculation in dynamic core requires higher precision since its equation may encounter the circumstance of multiplying a small value by a big value, which causes a significant loss of precision or even a NAN value, as shown in shown in the equations 5 and 6. Here, adv_uu and adv_vv represent the advection momentum component in the θ and λ direction. Hence, the tracer formulation and the Canuto parametrization are implemented in single precision in swLICOM.

$$adv\_uu_{i,j,k} = -(u\frac{\partial u}{\partial \theta})_{i,j,k} - (v\frac{\partial u}{\partial \lambda})_{i,j,k} - (\frac{\partial wu}{\partial z})_{i,j,k} \quad (5)$$

$$adv\_vv_{i,j,k} = -(u\frac{\partial v}{\partial \theta})_{i,j,k} - (v\frac{\partial v}{\partial \lambda})_{i,j,k} - (\frac{\partial wv}{\partial z})_{i,j,k} \quad (6)$$

3.4.2 Design Accuracy Test

The metric used to evaluate the similarity between double and mixed precision versions is root mean square deviation (RMSD) (Ye et al., 2022). However, directly using RMSD to evaluate the accuracy is not appropriate. LICOM is developed to a variable resolution gridpoint model by adopting tripolar grids (Yu et al., 2018). Hence, the area of each grid point is different. Due to this reason, we need to utilize the area of each grid for our accuracy formula instead of the original number of gridpoints to get a more accurate deviation:

In this formula, i, j are the spatial grid indice, t is the time index, $dbl_{i,j}(t)$ is the value in double precision version at the given i, j gridpoint and given t time. $mix_{i,j}(t)$ is the corresponding value in the mixed precision version. $area_{i,j}$ is the area of each grid with a unit of square kilometer.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

4 Experiments and test results

4.1 Model Configuration

All of our evaluations are conducted in a new generation of Sunway HPC. Comprehensive experiments on a kilometer-scale are performed to evaluate the performance and fidelity of our new swLICOM with 1 km, 2 km, and 10km resolution, respectively. The model configuration is shown in Table 1. Despite the different resolution, the region stays the same, from 76°S to 90°N. Forcing field is adopted by Japanese 55-year Reanalysis for driving ocean–sea-ice models (JRA55-do) with 3-hour frequency (Tsujino et al., 2018), except runoff and sea-ice with daily frequency. We use the bilinear method to interpolate the forcing field to map with different model resolutions. The case of 10km resolution is a classical case based on the previous LICOM experiment (Wang et al., 2021; Wei et al., 2023). It is used to break down our innovations to quantify the performance improvement of new domain decomposition, I/O optimization, and mixed precision, etc. The case of 1-km and 2-km ultra-high resolution is used to evaluate the strong extensibility of swLICOM. Meanwhile, we use the case of 2-km resolution to evaluate the fidelity of swLICOM by hindcast simulation. The performance indicators are represented by using SDPD (simulated-days-per-day).

Table 1. Configuration of our grids and timesteps. The case of 10-km resolution is used to implement performance metrics to quantify the performance improvement of new domain decomposition. The case of 1-km and 2-km resolution is used to evaluate the fidelity of swLICOM and the performance of swLICOM. The maximum grid point we adopted is beyond 63 billion.

Label	Resolution (km)	Latitude grids	Longitude grids	Layers	Timestep		
					Barotropic	Baroclinic	Tracer
D1	1	36000	22081	80	2	30	30
D2	2	18000	11511	244	2	30	30
D3	10	3600	2302	80	9	90	90

4.2 Performance Metric

To evaluate our innovations, a set of software versions is listed in Table 2. The version of LICOM-Org is the original version ported from the Intel platform, which is calculated by MPE cores. Each innovative method is separately implemented into different software versions for evaluation. To perform performance analysis and profiling accurately, we port the GPTL library (Version 8.1.1) into LICOM software and manually insert GPTL instrumentation. We use the max time interval of MPI ranks for statistics. Whenever the module computation process or I/O process, the time consumption can be accurately counted. We utilize 10km resolution with time steps set to 9, 90, and 90s for the barotropic, baroclinic, and tracer, respectively. We implement three different parallel scales for each comparison. The performance improvement is shown in Figure 8.

As shown in Figure 8a, our accelerated LICOM-CPE version gets a maximum 12.3 times speedup compared with the original LICOM version with 1170 ranks. Along with the parallel scale's growth, speedup times are decreased since more time is spent on MPI communication.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

Table 2. LICOM software version list. LICOM-Org is the original version as a benchmark. The others are optimized versions implementing our innovative method separately, which are used to verify performance improvement.

Implementations	CPE acceleration	Removing land point	IO optimization	Mixed precision
LICOM-Org	✖	✖	✖	✖
LICOM-CPE	✔	✖	✖	✖
LICOM-Domain	✔	✔	✖	✖
LICOM-IO	✔	✔	✔	✖
swLICOM	✔	✔	✔	✔

To fairly evaluate the performance of our new removing land point domain decomposition, we use three same parallel scales for verification. We adopt original Cartesian domain decomposition for LICOM-CPE and new Cartesian_noland decomposition for LICOM-Domain version. From the different parallel scales, we can get an average above 30% speedup. This significant improvement is because there is no need to implement meaningless calculations on land points and boundary data exchange with land points. Especially in a super large parallel scale, the new domain decomposition will save more than 10 million cores.

Mix precision is implemented in our final swLICOM version. We get an average of 1.5 times speedup compared with the double precision version LICOM-IO. The main contribution is that the parametrization scheme of Canuto in LICOM is totally implemented with a single precision. It decreases the load balance among ranks, which indirectly causes less time spent in the following halo update in the barotropic process.

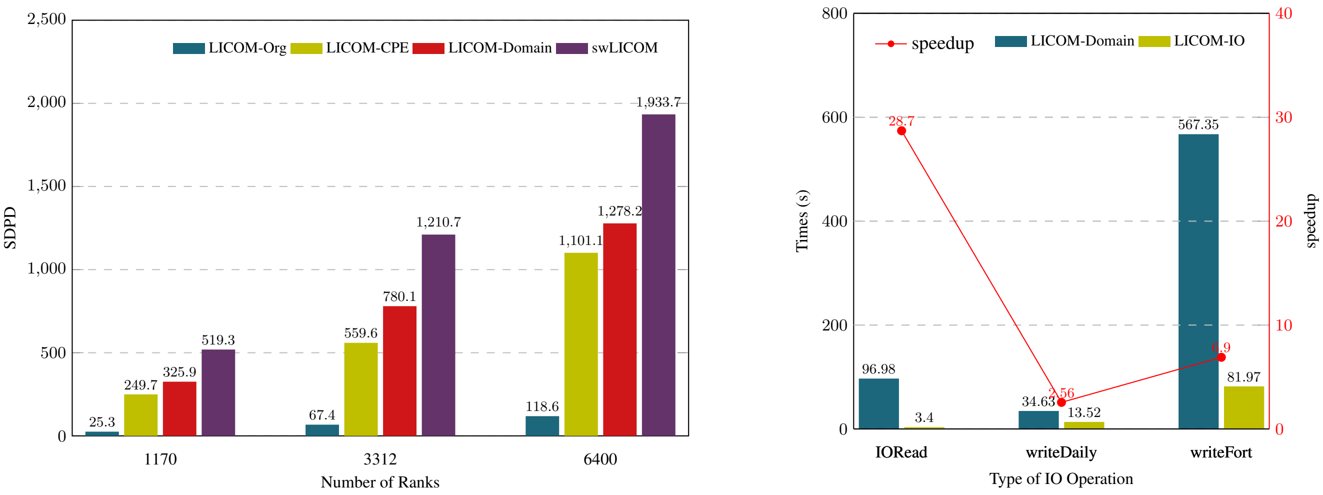
Our breakdown experiment demonstrates that each innovation method has significantly improved performance. Ultimately, we get a maximum speedup of 20 times for swLICOM compared with the LICOM-Org version.

IO experiment is implemented to simulate 1 day on 1170 ranks, which is configured as startup mode, writing history file every 6 hours and a daily file per day, as shown in Figure 8b. The IORead is the total time of reading the initial field, forcing field, and grid data. The writeDaily and writeFort represent the time of writing instantaneous files and daily files, respectively. The speedup of IORead is up to 28.7. The reason is that in the original version, data is read by a single core and then scattered to the whole ranks, according to tripole boundary conditions. If this original reading method is used in the 1-km/2-km experiment with super high parallel scales, the time spent will exponentially increase. By using our split reading strategy, the time spent will be kept the same at 10km resolution. Although PNetCDF (Parallel I/O Library for NetCDF File Access) is used in the original LICOM version for writing, our new split writing scheme still shows its advantages with 2.56 and 6.9 times speedup than PNetCDF.

4.3 Weak Scaling

The weak scalability test is evaluated by our final version swLICOM. We choose six different resolutions with the exact latitude and longitude block per rank and the same barotropic and baroclinic steps, as shown in Table 3. The horizontal resolutions range from 10km to 1 km, with cores increasing from 284,310 to 25,326,145. As shown in Figure 9, we achieve 92% parallel

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025



(a) Metric of Performance improvement

(b) IO optimization

Figure 8. The performance improvement and IO optimizations. (a) The performance (Simulated Days per Day, SDPD) for the original version of LICOM (LICOM-Org, blue bar) as a reference and three main optimizing versions, including porting to CPE (LICOM-CPE, yellow bar), removing land points (LICOM-Domain, red bar), and the mix-precision (swLICOM, purple bar) on three scales run with 1170, 3312, and 6400 ranks. (b) The performance of the IO time in a single simulation day for LICOM-Domain (blue bar) as a reference and the IO-optimized version LICOM-IO (yellow bar) for the reading files (IORead), writing daily mean files and restart files (writeDaily), and writing instantaneous files (writeFort) on single scales with 1170 ranks. The red line is the speedup. The LICOM-Domain version here uses serial NetCDF reading and PNetCDF parallel writing.

efficiency under the maximum 25,326,145 cores. The linear tendency chart along different resolutions demonstrates good weak scalability of swLICOM.

Table 3. Weak Scaling Configuration and timesteps. We select six different resolutions, ranging from 10 km to 1 km, to implement a weak scalability test. Each resolution is adopted with the same latitude and longitude block per rank and the same barotropic and baroclinic steps.

Label	Resolution (km)	Latitude grids	Longitude grids	Layers	Decomposition		Timestep		Sunway cores
					Latitude Block	Longitude Block	Barotropic	Baroclinic	
WD1	1	36000	22081	80	48	28	2	30	25326145
WD2	2	18000	11511	80	48	28	2	30	6621290
WD3	3.33	10800	6907	80	48	28	2	30	2421055
WD4	5	7200	4605	80	48	28	2	30	1094795
WD5	6.66	5400	3453	80	48	28	2	30	623805
WD6	10	3600	2302	80	48	28	2	30	284310

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

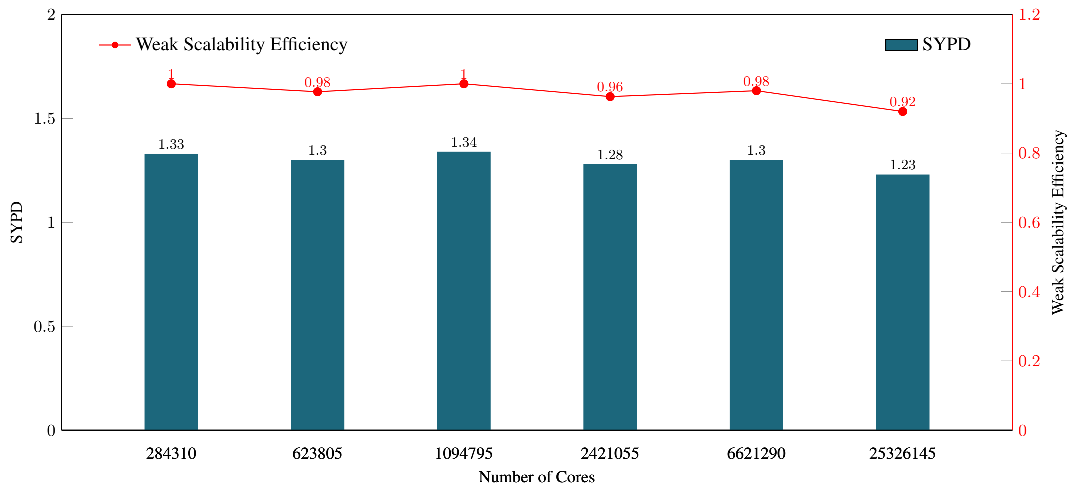


Figure 9. The weak scaling for swLICOM (Simulated Years per Day, SYPD) with 6 different resolutions of 1km, 2km, 3.33km, 5km, 6.66km, and 10km and the scaling of 284,310, 623,805, 1,094,795, 2,421,055, 6,621,290 and 25,326,145 cores, respectively. The red line is the efficiency.

4.4 Strong Scaling

We conduct a strong scalability test with 1-km and 2-km resolution, respectively, which is configured as D1/D2, shown in Table 1. The 2-km test starts from 1,277,770 cores to 24,315,850 cores, and the 1-km test starts from 1,201,070 to 26,995,020. The strong scaling test result is shown in Figure 10. We achieve peak performance of 410 SDPD with 45% parallel efficiency at 2 km resolution and 453 SDPD with 59% parallel efficiency at 1 km resolution. Although the total grid points of 1 km resolution are 1.25 times more than 2 km resolution, the parallel efficiency and SDPD of 1 km resolution are obviously better than 2 km resolution. The main reason is that the more vertical layers conduct the more memory access, and lower the performance.

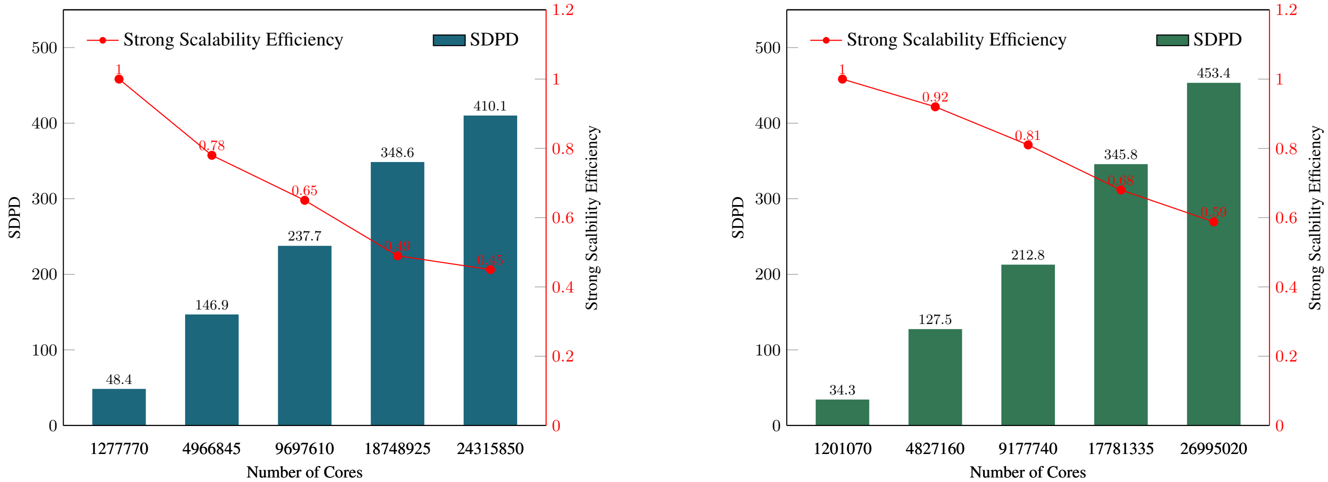
4.5 Accuracy Test of Mix Precision

To verify the accuracy of mix precision, we conducted a one-month simulation on both the double precision version of LICOM-Domain and the mix precision version of silicon. As introduced in section 3.4.2, we calculate the average value of 30 daily data points for temperature, salt, and sea surface height. Then, we use our accuracy formula MIXRMSD to evaluate the accuracy, as shown in Figure 11. The MIXRMSD of temperature, salt, and sea surface height is around 0.018, 0.0098, and 0.0005, respectively.

4.6 The Ultra-high-resolution Simulation

We use swLICOM with a horizontal resolution of 2 km to conduct a short-term (50 days) simulation test in the new generation Sunway cluster. Figures 12 and 13 show the snapshots of sea surface variables in the Kuroshio and Gulf Stream regions,

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025



(a) Strong scalability of 2-km resolution

(b) Strong scalability of 1-km resolution

Figure 10. The strong scaling of swLICOM (Simulated Days per Day, SDPD) for (a) 2-km test ranging from 1,277,770 to 24,315,850 cores and (b) 1-km test ranging from 1,201,070 to 26,995,020 cores. The red lines are the efficiency.

respectively. The normalized vertical vorticity field, defined as $(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y})/f$, demonstrates active submesoscale processes with the magnitude of O(1), especially along the energetic jets and around the mesoscale eddies. The sea surface temperature field can also show clues of stirring submesoscale activities.

5 Conclusions

In conclusion, swLICOM represents a significant advancement in kilometer-scale resolution ocean general circulation models on heterogeneous computing architectures. By optimizing LICOM3 for the new generation Sunway supercomputer, we successfully address several key challenges associated with ultra-high-resolution global simulations. Through our effort, swLICOM demonstrates a peak performance of 453 SDPD with 59% efficiencies at a 1-km resolution on 26,995,020 cores and 410 SDPD with 45% efficiencies at a 2-km resolution on 24,315,850 cores in terms of strong scalability. Moreover, swLICOM captures the vigorous mesoscale eddies and active submesoscale phenomena with 2 km horizontal resolution.

Our optimization efforts addressed a series of challenges that are particularly crucial for high-resolution modeling. The automatic parallel code translation tool, swCUDA, is optimized to enhance computational performance and simplify the porting process for future applications, thereby promoting productivity and scalability. The initialization phase suffers from scalability limitations due to extreme parallelism processes and point-to-point (send-recv) communication patterns. Meanwhile, output operations require gathering hundreds of gigabytes of data into a single process, creating severe bottlenecks in write performance. To tackle the problem, a split I/O scheme is developed to improve IO bandwidth for large-scale computation and simulation capability at kilometer-scale resolutions. The time for both initialization and output operations are reduced

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

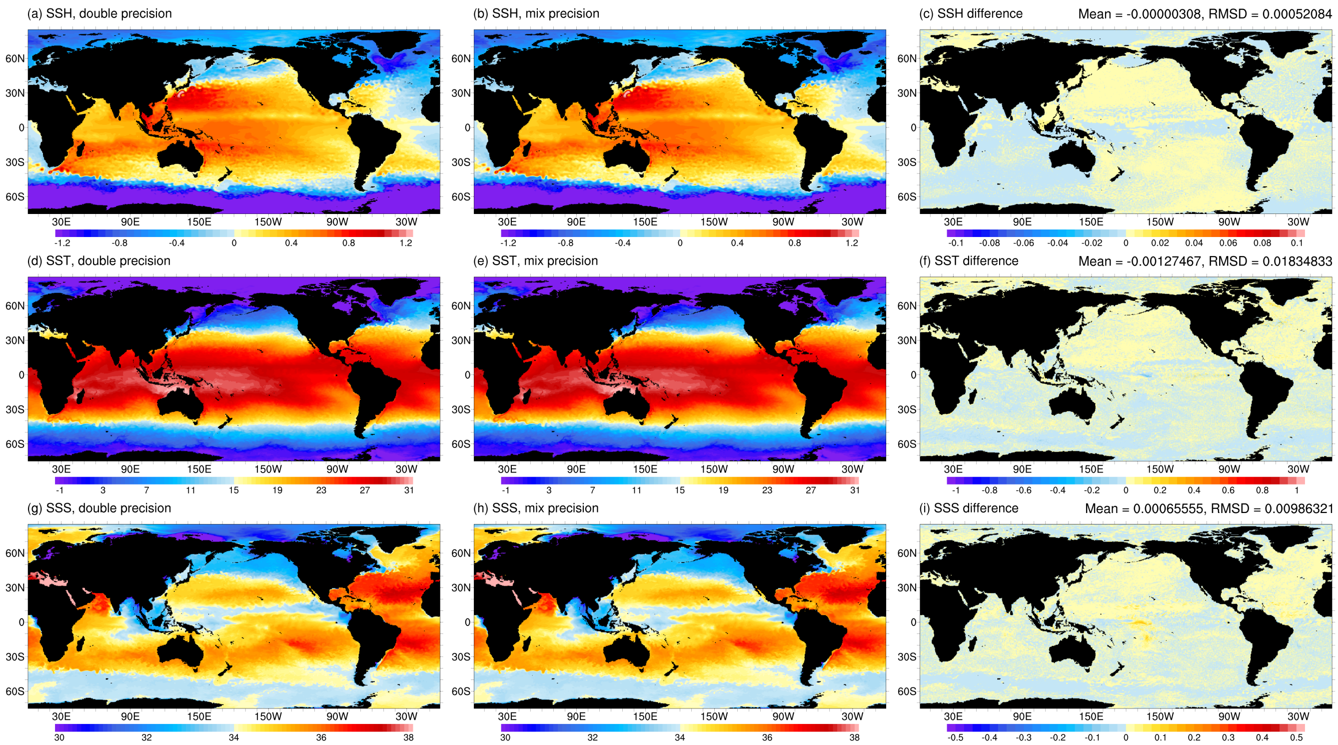


Figure 11. The monthly mean sea surface height for (a) double precision, (b) mix precision, and (c) their difference in January 2016, simulated by swLICOM at a 10 km resolution. (d-f) are for sea surface temperature and (g-i) are for sea surface salinity. The upper right side shows the global mean and root-mean-square difference of the corresponding variables

from several hours to approximately 10 minutes for ultra-high-resolution simulations. Moreover, removing land points from computations significantly reduces the computational resources required for high-resolution simulation while maintaining the integrity of marine-related computations. This method can achieve an average about 30% speedup which corresponds to the proportion of land area relative to Earth's total surface area. Furthermore, mixed precision is implemented to reduce memory usage and improve communication bottlenecks for such a large scale. The proposed implementation can gain about 50% speedup versus the double-only implementation. The adaptive decomposition method and memory access optimizations further contribute to the overall efficiency by maximizing cache utilization and improving data transmission speed. The adoption of bilinear interpolation or a forcing field significantly improves simulation accuracy.

Implementing swLICOM on the Sunway supercomputer sets a benchmark for future OGCM developments, highlighting the potential of heterogeneous computing in advancing climate and oceanographic studies. We use swLICOM with a horizontal resolution of 2 km to conduct a short-term (50 days) simulation test. The 2-km resolution global simulation shows the high capacity of swLICOM to capture the oceanic meso- to submesoscale processes.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

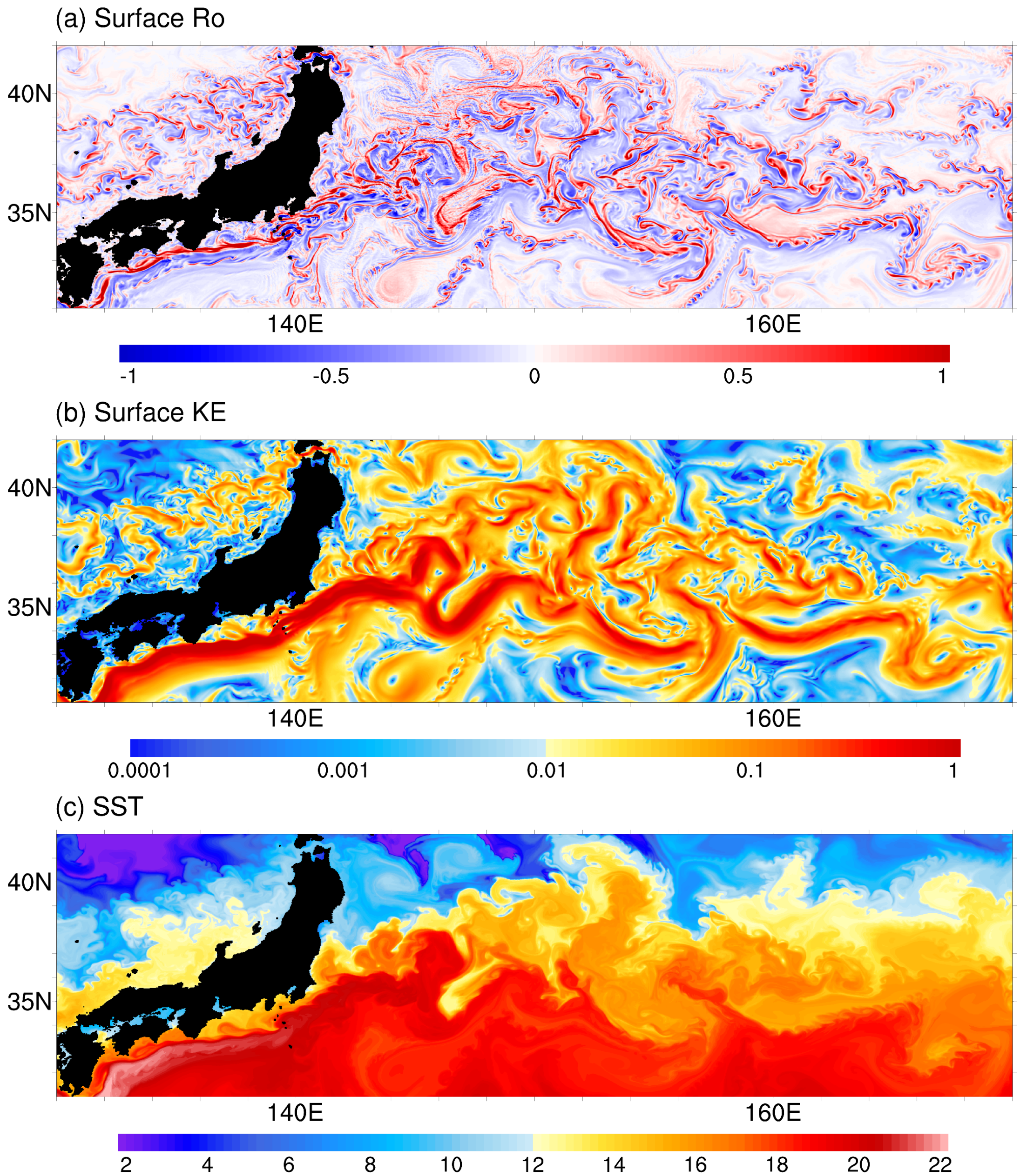


Figure 12. Snapshots of (a) surface normalized vertical vorticity, (b) surface kinetic energy, and (c) sea surface temperature in the Kuroshio region on February 20, 2016 simulated by swLICOM with a 2 km horizontal resolution.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

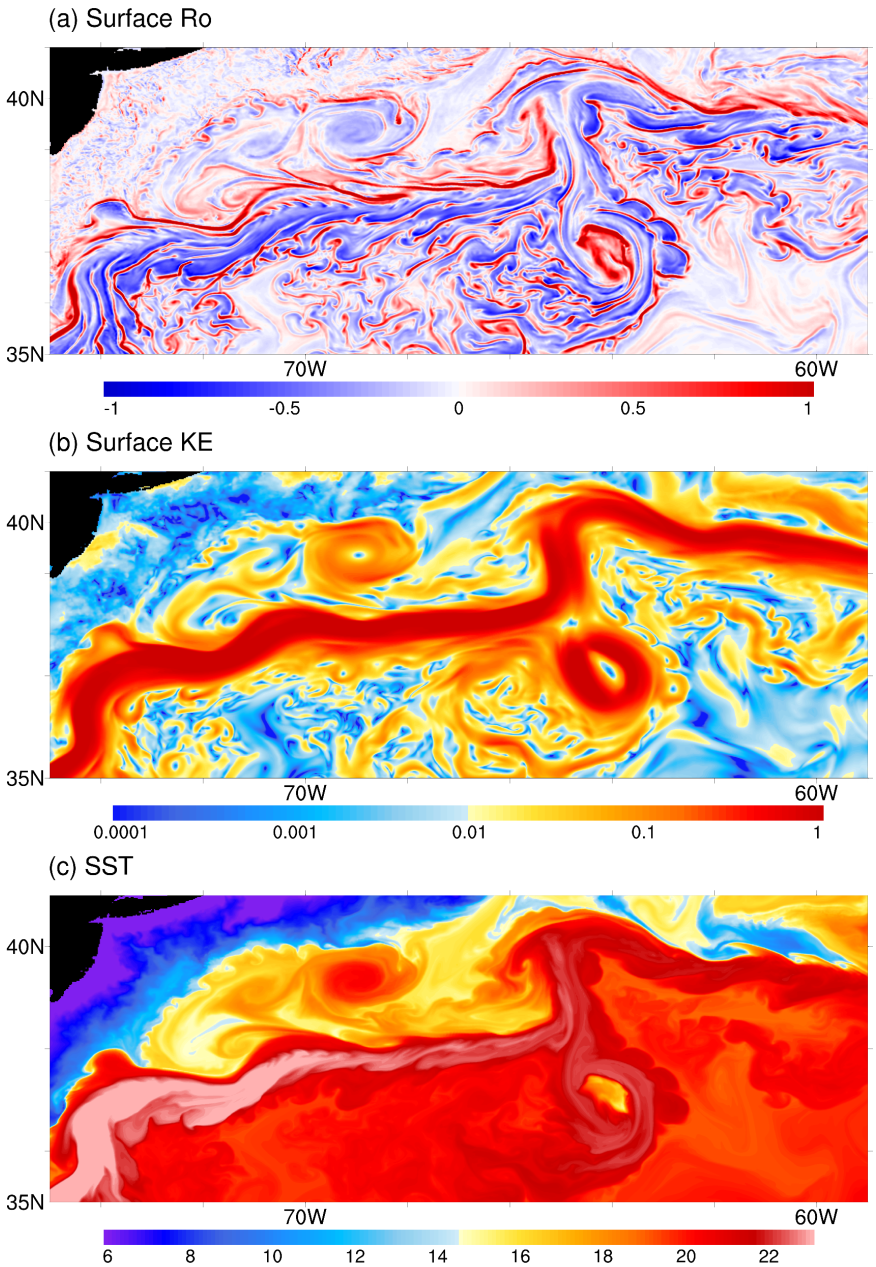


Figure 13. Same as Figure 12, but for the Gulf Stream.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

6 Code and data availability

The current version of swCUDA is available from the project website https://zenodo.org/records/15494635 under the Creative Commons Attribution 4.0 International licence. The exact version of the model used to produce the results used in this paper is archived on repository under 10.5281/zenodo.15494635 (Xu (2025)), as are data and scripts to run the model and produce the plots for all the simulations presented in this paper.

Acknowledgements. This study is funded by the National Natural Science Foundation of China (U2242214 and 92358302), the National Key R&D Program for Developing Basic Sciences (2022YFC3104802), the Tai Shan Scholar Program (Grant No. tstp20231237), and Laoshan Laboratory Project (LSKJ202300301).

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

References

Aketh, T. M., Vadhiyar, S., Vinayachandran, P. N., and Nanjundiah, R.: High Performance Horizontal Diffusion Calculations in Ocean Models on Intel® Xeon Phi™ Coprocessor Systems, in: 2016 IEEE 23rd International Conference on High Performance Computing (HiPC), pp. 203–211, https://doi.org/10.1109/HiPC.2016.032, 2016.

Balwada, D., Smith, K. S., and Abernathey, R.: Submesoscale vertical velocities enhance tracer subduction in an idealized Antarctic Circumpolar Current, Geophysical Research Letters, 45, 9790–9802, https://doi.org/10.1029/2018GL079244, 2018.

Bao, Q., Lin, P. F., Zhou, T. J., Liu, Y. M., Yu, Y. Q., Wu, G. X., He, B., He, J., Li, L. J., Li, J. D., Li, Y. C., Liu, H. L., Qiao, F. L., Song, Z. Y., Wang, B., Wang, J., Wang, P. F., Wang, X. C., Wang, Z. Z., Wu, B., Wu, T. W., Xu, Y. F., Yu, H. Y., Zhao, W., Zheng, W. P., and Zhou, L. J.: The Flexible Global Ocean-Atmosphere-Land system model, Spectral Version 2: FGOALS-s2, Adv. Atmos. Sci., 30, 561–576, https://doi.org/10.1007/s00376-012-2113-9, 2013.

Canuto, V. M., Howard, A., Cheng, Y., and Dubovikov, M. S.: Ocean Turbulence. Part II: Vertical Diffusivities of Momentum, Heat, Salt, Mass, and Passive Scalars, Journal of Physical Oceanography, 32, 240 – 264, https://doi.org/10.1175/1520-0485(2002)032<0240:OTPIVD>2.0.CO;2, 2002.

Chassignet, E. P. and Xu, X.: Impact of horizontal resolution (1/12 to 1/50) on Gulf Stream separation, penetration, and variability, Journal of Physical Oceanography, 47, 1999–2021, https://doi.org/10.1175/JPO-D-17-0031.1, 2017.

Chen, S., Zhang, Y., Wang, Y., Liu, Z., Li, X., and Xue, W.: Mixed-precision computing in the GRIST dynamical core for weather and climate modelling, Geoscientific Model Development, 17, 6301–6318, https://doi.org/10.5194/gmd-17-6301-2024, 2024.

Eirund, G. K., Leclair, M., Muennich, M., and Gruber, N.: ROMSOC: A regional atmosphere-ocean coupled model for CPU-GPU hybrid system architectures, EGUsphere, 2024, 1–27, https://doi.org/10.5194/egusphere-2024-2922, 2024.

Fuhrer, O., Chadha, T., Hoefler, T., Kwasniewski, G., Lapillonne, X., Leutwyler, D., Lüthi, D., Osuna, C., Schär, C., Schulthess, T. C., and Vogt, H.: Near-global climate simulation at 1 km resolution: establishing a performance baseline on 4888 GPUs with COSMO 5.0, Geoscientific Model Development, 11, 1665–1681, https://doi.org/10.5194/gmd-11-1665-2018, 2018.

Griffies, S. M.: Elements of mom4p1, GFDL Ocean Group Tech., 6, 444, 2009.

Gula, J., Taylor, J., Shcherbina, A., and Mahadevan, A.: Chapter 8: Submesoscale processes and mixing, in: Ocean Mixing, edited by Meredith, M. and Garabato, A. N., pp. 181–214, Elsevier, ISBN 9780128215128, https://doi.org/10.1016/B978-0-12-821512-8.00015-3, 2022.

Gysi, T., Osuna, C., Fuhrer, O., Bianco, M., and Schulthess, T. C.: STELLA: a domain-specific tool for structured grid methods in weather and climate models, in: SC '15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–12, https://doi.org/10.1145/2807591.2807627, 2015.

Hallberg, R.: Using a resolution function to regulate parameterizations of oceanic mesoscale eddy effects, Ocean Modelling, 72, 92–103, https://doi.org/10.1016/j.ocemod.2013.08.007, 2013.

Hewitt, H. T., Roberts, M., Mathiot, P., Biastoch, A., Blockley, E., Chassignet, E. P., Fox-Kemper, B., Hyder, P., Marshall, D. P., Popova, E., et al.: Resolving and parameterising the ocean mesoscale in earth system models, Current Climate Change Reports, 6, 137–152, https://doi.org/10.1007/s40641-020-00164-w, 2020.

Hohenegger, C., Korn, P., Linardakis, L., Redler, R., Schnur, R., Adamidis, P., Bao, J., Bastin, S., Behravesh, M., Bergemann, M., et al.: ICON-Sapphire: simulating the components of the Earth system and their interactions at kilometer and subkilometer scales, Geoscientific Model Development, 16, 779–811, https://doi.org/10.5194/gmd-16-779-2023, 2023.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

Häfner, D., Nuterman, R., and Jochum, M.: Fast, Cheap, and Turbulent—Global Ocean Modeling With GPU Acceleration in Python, Journal of Advances in Modeling Earth Systems, 13, e2021MS002 717, https://doi.org/https://doi.org/10.1029/2021MS002717, e2021MS002717 2021MS002717, 2021.

Jiang, J., Lin, P., Wang, J., Liu, H., Chi, X., Hao, H., Wang, Y., Wang, W., and Zhang, L.: Porting LASG/ IAP Climate System Ocean Model to Gpus Using OpenAcc, IEEE Access, 7, 154 490–154 501, https://doi.org/10.1109/ACCESS.2019.2932443, 2019.

Kjellsson, J. and Zanna, L.: The impact of horizontal resolution on energy transfers in global ocean models, Fluids, 2, 45, https://doi.org/10.3390/fluids2030045, 2017.

Li, L. J., Lin, P. F., Yu, Y. Q., Wang, B., Zhou, T. J., Liu, L., Liu, J. P., Bao, Q., Xu, S. M., Huang, W. Y., Xia, K., Pu, Y., Dong, L., Shen, S., Liu, Y. M., Hu, N., Liu, M. M., Sun, W. Q., Shi, X. J., Zheng, W. P., Wu, B., Song, M. R., Liu, H. L., Zhang, X. H., Wu, G. X., Xue, W., Huang, X. M., Yang, G. W., Song, Z. Y., and Qiao, F. L.: The flexible global ocean-atmosphere-land system model, Grid-point Version 2: FGOALS-g2, Adv. Atmos. Sci., 30, 543–560, https://doi.org/10.1007/s00376-012-2140-6, 2013.

Li, Y. W., Liu, H. L., Ding, M. R., Lin, P. F., Yu, Z. P., Yu, Y. Q., Meng, Y., Li, Y. L., Jian, X. D., Jiang, J. R., Chen, K. J., Yang, Q., Wang, Y. Q., Zhao, B. W., Wei, J. L., Ma, J. F., Zheng, W. P., and Wang, P. F.: Eddy-resolving Simulation of CAS-LICOM3 for Phase 2 of the Ocean Model Intercomparison Project, Adv. Atmos. Sci., 37, 1067–1080, https://doi.org/10.1007/s00376-020-0057-z, 2020.

Lin, P., Yu, Z., Liu, H., Yu, Y., Li, Y., Jiang, J., Xue, W., Chen, K., Yang, Q., Zhao, B., Wei, J., Ding, M., Sun, Z., Wang, Y., Meng, Y., Zheng, W., and Ma, J.: LICOM Model Datasets for the CMIP6 Ocean Model Intercomparison Project, Acta Oceanol. Sin., 37, 662–662, https://doi.org/10.1007/s00376-020-2005-3, 2020.

Mesinger, F., Janjić, Z. I., Ničković, S., Gavrilov, D., and Deaven, D. G.: The Step-Mountain Coordinate: Model Description and Performance for Cases of Alpine Lee Cyclogenesis and for a Case of an Appalachian Redevelopment, Mon. Wea. Rev., 116, 1493 – 1518, https://doi.org/10.1175/1520-0493(1988)116<1493:TSMCMD>2.0.CO;2, 1988.

Parr, T., Harwell, S., and Fisher, K.: Adaptive LL(*) Parsing: The Power of Dynamic Analysis, SIGPLAN Not., 49, 579–598, https://doi.org/10.1145/2714064.2660202, 2014.

Schubert, R., Schwarzkopf, F. U., Baschek, B., and Biastoch, A.: Submesoscale impacts on mesoscale Agulhas dynamics, Journal of Advances in Modeling Earth Systems, 11, 2745–2767, https://doi.org/10.1029/2019MS001724, 2019.

Silvestri, S., Wagner, G., Hill, C., Ardakani, M. R., Blaschke, J., Campin, J.-M., Churavy, V., Constantinou, N., Edelman, A., Marshall, J., Ramadhan, A., Souza, A., and Ferrari, R.: Oceananigans.jl: A model that achieves breakthrough resolution, memory and energy efficiency in global ocean simulations, https://arxiv.org/abs/2309.06662, 2023.

Silvestri, S., Wagner, G. L., Constantinou, N. C., Hill, C. N., Campin, J.-M., Souza, A. N., Bishnu, S., Churavy, V., Marshall, J. C., and Ferrari, R.: A GPU-based ocean dynamical core for routine mesoscale-resolving climate simulations, ESS Open Archive eprints, 823, 171708158.82342448, https://doi.org/10.22541/essoar.171708158.82342448/v1, 2024.

Smith, R., Jones, P., Briegleb, B., Bryan, F., Danabasoglu, G., Dennis, J., Dukowicz, J., Eden, C., Fox-Kemper, B., Gent, P., Hecht, M., Jayne, S., Jochum, M., Large, W., Lindsay, K., Maltrud, M., Norton, N., Peacock, S., Vertenstein, M., and Yeager, S.: The Parallel Ocean Program (POP) Reference Manual, LAUR-10-01853, Los Alamos National Laboratory, 2010.

Smith, R. D., Dukowicz, J. K., and Malone, R. C.: PARALLEL OCEAN GENERAL-CIRCULATION MODELING, Phys. D Nonlinear phenom., 60, 38–61, https://doi.org/10.1016/0167-2789(92)90225-C, 1992.

Soufflet, Y., Marchesiello, P., Lemarié, F., Jouanno, J., Capet, X., Debreu, L., and Benshila, R.: On effective resolution in ocean models, Ocean Modelling, 98, 36–50, https://doi.org/10.1016/j.ocemod.2015.12.004, 2016.

https://doi.org/10.5194/egusphere-2025-2231
Preprint. Discussion started: 27 August 2025

Su, Z., Wang, J., Klein, P., Thompson, A. F., and Menemenlis, D.: Ocean submesoscales as a key component of the global heat budget, Nature communications, 9, 775, https://doi.org/10.1038/s41467-018-02983-w, 2018.

Thaler, F., Moosbrugger, S., Osuna, C., Bianco, M., Vogt, H., Afanasyev, A., Mosimann, L., Fuhrer, O., Schulthess, T. C., and Hoefler, T.: Porting the COSMO Weather Model to Manycore CPUs, in: Proceedings of the Platform for Advanced Scientific Computing Conference, PASC '19, Association for Computing Machinery, New York, NY, USA, ISBN 9781450367707, https://doi.org/10.1145/3324989.3325723, 2019.

Tintó Prims, O., Acosta, M. C., Moore, A. M., Castrillo, M., Serradell, K., Cortés, A., and Doblas-Reyes, F. J.: How to use mixed precision in ocean models: exploring a potential reduction of numerical precision in NEMO 4.0 and ROMS 3.6, Geoscientific Model Development, 12, 3135–3148, https://doi.org/10.5194/gmd-12-3135-2019, 2019.

Tsujino, H., Urakawa, S., Nakano, H., Small, R. J., Kim, W. M., Yeager, S. G., Danabasoglu, G., Suzuki, T., Bamber, J. L., Bentsen, M., Böning, C. W., Bozec, A., Chassignet, E. P., Curchitser, E., Boeira Dias, F., Durack, P. J., Griffies, S. M., Harada, Y., Ilicak, M., Josey, S. A., Kobayashi, C., Kobayashi, S., Komuro, Y., Large, W. G., Le Sommer, J., Marsland, S. J., Masina, S., Scheinert, M., Tomita, H., Valdivieso, M., and Yamazaki, D.: JRA-55 based surface dataset for driving ocean–sea-ice models (JRA55-do), Ocean Modelling, 130, 79–139, https://doi.org/https://doi.org/10.1016/j.ocemod.2018.07.002, 2018.

Tsujino, H., Urakawa, L. S., Griffies, S. M., Danabasoglu, G., Adcroft, A. J., Amaral, A. E., Arsouze, T., Bentsen, M., Bernardello, R., Boning, C. W., Bozec, A., Chassignet, E. P., Danilov, S., Dussin, R., Exarchou, E., Fogli, P. G., Fox-Kemper, B., Guo, C. C., Ilicak, M., Iovino, D., Kim, W. M., Koldunov, N., Lapin, V., Li, Y. W., Lin, P. F., Lindsay, K., Liu, H. L., Long, M. C., Komuro, Y., Marsland, S. J., Masina, S., Nummelin, A., Rieck, J. K., Ruprich-Robert, Y., Scheinert, M., Sicardi, V., Sidorenko, D., Suzuki, T., Tatebe, H., Wang, Q., Yeager, S. G., and Yu, Z. P.: Evaluation of global ocean-sea-ice model simulations based on the experimental protocols of the Ocean Model Intercomparison Project phase 2 (OMIP-2), Geosci. Model Dev., 13, 3643–3708, https://doi.org/10.5194/gmd-13-3643-2020, 2020.

Wang, P., Jiang, J., Lin, P., Ding, M., Wei, J., Zhang, F., Zhao, L., Li, Y., Yu, Z., Zheng, W., Yu, Y., Chi, X., and Liu, H.: The GPU version of LASG/IAP Climate System Ocean Model version 3 (LICOM3) under the heterogeneous-compute interface for portability (HIP) framework and its large-scale application, Geoscientific Model Development, 14, 2781–2799, https://doi.org/10.5194/gmd-14-2781-2021, 2021.

Wei, J., Jiang, J., Liu, H., zhang, F., Lin, P., Wang, P., Yu, Y., Chi, X., Zhao, L., Ding, M., Li, Y., Yu, Z., Zheng, W., and Wang, Y.: LICOM3-CUDA: a GPU version of LASG/IAP climate system ocean model version 3 based on CUDA, The Journal of Supercomputing, 79, 2781–2799, https://doi.org/10.1007/s11227-022-05020-2, 2023.

Wei, J., Han, X., Yu, J., Jiang, J., Liu, H., Lin, P., Yu, M., Xu, K., Zhao, L., Wang, P., Zheng, W., Xie, J., Zhou, Y., Zhang, T., Zhang, F., Zhang, Y., Yu, Y., Wang, Y., Bai, Y., Li, C., Yu, Z., Deng, H., Li, Y., and Chi, X.: A Performance-Portable Kilometer-Scale Global Ocean Model on ORISE and New Sunway Heterogeneous Supercomputers, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, SC '24, IEEE Press, ISBN 9798350352917, https://doi.org/10.1109/SC41406.2024.00009, 2024.

Xie, J., Wang, X., Liu, H., Lin, P., Yu, J., Yu, Z., Wei, J., and Han, X.: ISOM 1.0: a fully mesoscale-resolving idealized Southern Ocean model and the diversity of multiscale eddy interactions, Geoscientific Model Development, 17, 8469–8493, https://doi.org/10.5194/gmd-17-8469-2024, 2024.

Xie, J., Yu, J., Zhou, Y., Liu, H., Wei, J., Han, X., Xu, K., Yu, M., Yu, Z., Lin, P., Jiang, J., Zheng, W., Zhang, T., Wang, R., Jing, Z., and Wu, L.: A 1-km resolution global ocean simulation promises to unveil oceanic multi-scale dynamics and climate impacts, The Innovation, 6, 100 843, https://doi.org/10.1016/j.xinn.2025.100843, 2025.

Xu, K.: swLICOM, https://doi.org/10.5281/zenodo.15494635, 2025.

Xu, K., Song, Z., Chan, Y., Wang, S., Meng, X., Liu, W., and Xue, W.: Refactoring and Optimizing WRF Model on Sunway TaihuLight, in: Proceedings of the 48th International Conference on Parallel Processing, ICPP '19, Association for Computing Machinery, New York, NY, USA, ISBN 9781450362955, https://doi.org/10.1145/3337821.3337923, 2019.

Xu, S., Huang, X., Oey, L.-Y., Xu, F., Fu, H., Zhang, Y., and Yang, G.: POM.gpu-v1.0: a GPU-based Princeton Ocean Model, Geoscientific Model Development, 8, 2815–2827, https://doi.org/10.5194/gmd-8-2815-2015, 2015.

Yamagishi, T. and Matsumura, Y.: GPU Acceleration of a Non-hydrostatic Ocean Model with a Multigrid Poisson/Helmholtz solver, Procedia Computer Science, 80, 1658–1669, https://doi.org/https://doi.org/10.1016/j.procs.2016.05.502, international Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA, 2016.

Ye, Y., Song, Z., Zhou, S., Liu, Y., Shu, Q., Wang, B., Liu, W., Qiao, F., and Wang, L.: swNEMO_v4.0: an ocean model based on NEMO4 for the new-generation Sunway supercomputer, Geoscientific Model Development, 15, 5739–5756, https://doi.org/10.5194/gmd-15-5739-2022, 2022.

Y.Q., Y., S.L., T., H.L., L., P.F., L., and X.L., L.: Development and Evaluation of the Dynamic Framework of an Ocean General Circulation Model with Arbitrary Orthogonal Curvilinear Coordinate, Chinese Journal of Atmospheric Sciences (in Chinese), 42, 877–889, https://doi.org/10.3878/j.issn.1006-9895.1805.17284, 2018.

Yu, M., Ma, G., Wang, Z., Tang, S., Chen, Y., Wang, Y., Liu, Y., Jia, D., and Wei, Z.: swCUDA: Auto parallel code translation framework from CUDA to ATHREAD for new generation sunway supercomputer, CCF Transactions on High Performance Computing, 10, 2524–4930, https://doi.org/10.1007/s42514-023-00159-7, 2024.

Yu, Y., Tang, S., Liu, H., Lin, P., and Li, X.: Development and evaluation of the dynamic framework of an ocean general circulation model with arbitrary orthogonal curvilinear coordinate, Chinese Journal of Atmospheric Sciences, 42, 877–889, https://doi.org/10.3878/j.issn.1006-9895.1805.17284, 2018.

Zhang, H., Zhang, M., Jin, J., Fei, K., Ji, D., Wu, C., Zhu, J., He, J., Chai, Z., Xie, J., Dong, X., Zhang, D., Bi, X., Cao, H., Chen, H., Chen, K., Chen, X., Gao, X., Hao, H., Jiang, J., Kong, X., Li, S., Li, Y., Lin, P., Lin, Z., Liu, H., Liu, X., Shi, Y., Song, M., Wang, H., Wang, T., Wang, X., Wang, Z., Wei, Y., Wu, B., Xie, Z., Xu, Y., Yu, Y., Yuan, L., Zeng, Q., Zeng, X., Zhao, S., Zhou, G., and Zhu, J.: Description and Climate Simulation Performance of CAS-ESM Version 2, J. Adv. Model. Earth syst., 12, e2020MS002 210, https://doi.org/10.1029/2020MS002210, 2020a.

Zhang, S., Fu, H., Wu, L., Li, Y., Wang, H., Zeng, Y., Duan, X., Wan, W., Wang, L., Zhuang, Y., Meng, H., Xu, K., Xu, P., Gan, L., Liu, Z., Wu, S., Chen, Y., Yu, H., Shi, S., Wang, L., Xu, S., Xue, W., Liu, W., Guo, Q., Zhang, J., Zhu, G., Tu, Y., Edwards, J., Baker, A., Yong, J., Yuan, M., Yu, Y., Zhang, Q., Liu, Z., Li, M., Jia, D., Yang, G., Wei, Z., Pan, J., Chang, P., Danabasoglu, G., Yeager, S., Rosenbloom, N., and Guo, Y.: Optimizing high-resolution Community Earth System Model on a heterogeneous many-core supercomputing platform, Geoscientific Model Development, 13, 4809–4829, https://doi.org/10.5194/gmd-13-4809-2020, 2020b.

Zhang, Z., Liu, Y., Qiu, B., Luo, Y., Cai, W., Yuan, Q., Liu, Y., Zhang, H., Liu, H., Miao, M., et al.: Submesoscale inverse energy cascade enhances Southern Ocean eddy heat transport, Nature Communications, 14, 1335, https://doi.org/10.1038/s41467-023-36991-2, 2023.