Manuscript prepared for Geosci. Model Dev.
with version 2015/11/06 7.99 Copernicus papers of the LaTeX class copernicus.cls.
Date: 5 October 2016

P-CSI v1.0, an accelerated barotropic solver for the high-resolution ocean model component in the Community Earth System Model v2.0

Xiaomeng Huang[1,2], Qiang Tang[1], Yuheng Tseng[3], Yong Hu[1], Allison H. Baker[3], Frank O. Bryan[3], John Dennis[3], Haohuan Fu[1], and Guangwen Yang[1]

[1]Ministry of Education Key Laboratory for Earth System Modeling, and Center for Earth System Science, Tsinghua University, Beijing, 100084, China
[2]Laboratory for Regional Oceanography and Numerical Modeling, Qingdao National Laboratory for Marine Science and Technology, Qingdao, 266237, China
[3]The National Center for Atmospheric Research, Boulder, CO, USA

Correspondence to: Xiaomeng Huang (hxm@tsinghua.edu.cn), Yuheng Tseng(ytseng@ucar.edu)

Abstract. In the Community Earth System Model (CESM), the ocean model is computationally expensive for high-resolution grids and is often the least scalable component for high-resolution production experiments. The major bottleneck is that the barotropic solver scales poorly at high core counts. We design a new barotropic solver to accelerate the high-resolution ocean simulation. The novel solver adopts a Chebyshev-type iterative method to reduce the global communication cost in conjunction with an effective block preconditioner to further reduce the iterations. The algorithm and its computational complexity are theoretically analyzed and compared with other existing methods. We confirm the significant reduction of the global communication time with a competitive convergence rate using a series of idealized tests. Experimental results obtained with the CESM 0.1° global ocean model show that the proposed approach results in a factor of 1.7 speed-up over the original method with no loss of accuracy, achieving 10.5 simulated years per wall-clock day on 16,875 cores.

1 Introduction

Recent progress in high-resolution global climate models has demonstrated that models with finer resolution can better represent important climate processes to facilitate climate prediction. Significant improvements can be achieved in the high-resolution global simulations of Tropical Instability Waves (Roberts et al., 2009), El Niño Southern Oscillation (ENSO) (Shaffrey et al., 2009), the Gulf Stream separation (Chassignet and Marshall, 2008; Kuwano-Yoshida et al., 2010), the global water cycle (Demory et al., 2014), and other aspects of the mean climate and variability. Specifically, Gent et al. (2010) and Wehner et al. (2014) showed that increasing the atmosphere models' resolution results in a better mean climate, more accurate depiction of the tropical storm formation, and more realistic events of extreme daily precipitation. Bryan et al. (2010) and Graham (2014) also suggested that increasing the ocean models' resolution to the eddy resolving level helps capture the positive correlation between sea surface temperature and surface wind stress and improve the asymmetry of the ENSO cycle in the simulation.

In the High-Resolution Model Intercomparison Project (HighResMIP) for the Coupled Model Intercomparison Project phase 6 (CMIP6), global model resolutions of 25 km or finer at mid-latitudes are proposed to implement the Tier-1 and Tier-2 experiments (Eyring et al., 2015). Because all CMIP6 climate models are required to run for hundreds of years, tremendous computing resources are needed for high-resolution production simulations. To run high-resolution climate models practically, additional algorithm optimization is required to efficiently utilize the large-scale computing resources.

This work improves the barotropic solver performance in the ocean model component (Parallel Ocean Model, POP) of the National Center for Atmospheric Research (NCAR)'s fully coupled climate model: the Community Earth System Model (CESM). The POP solves the three-dimensional primitive equations with hydrostatic and Boussinesq approximations and splits the time integration into two parts: the baroclinic and barotropic modes (Smith et al., 2010). The baroclinic mode describes the three-dimensional dynamic and thermodynamic processes, and the barotropic mode solves the vertically integrated momentum and continuity equations in two dimensions.

The barotropic solver is the major bottleneck in the POP within the high-resolution CESM because it dominates the total execution time on a large number of cores (Jones et al., 2005). This results from the implicit calculation of the free-surface height in the barotropic solver, which scales poorly at high core counts due to an evident global communication bottleneck inherent to the algorithm. The implicit solver allows a large time step to efficiently compute the fast gravity wave mode but requires a large elliptic system of equations to be solved. The conjugate gradient method (CG) and its variants are popular choices in the implicit free-surface ocean solvers, such as MITgcm (Adcroft et al., 2014), FVCOM (Lai et al., 2010), MOM3 (Pacanowsky and Griffies, 1999), and OPA (Madec et al., 1997). However, the standard CG method has heavy global communication overhead in the existing POP implementation (Worley et al., 2011). The latest Chronopoulos-Gear (ChronGear) (D'Azevedo et al., 1999) variant of the CG algorithm is currently used in the POP to reduce the number of global reductions. A good overview of reducing global communication costs for CG method can be found in the work of Ghysels and Vanroose (2014). Recent efforts to improve the performance of CG method include a variant that overlaps the global reduction with the matrix-vector computation via a pipelined approach (Ghysels and Vanroose, 2014). However, the improvement is still limited when using a very large number of cores because of the remaining global reduction operations. For example, when approximately 4,000 cores are used, the global reduction in PCG (Preconditioned Conjugate Gradient method) and ChronGear takes approximately 74% and 68% of the whole barotropic time, respectively (Hu et al., 2015). This situation will worsen with more cores.

Another way to improve the CG method is preconditioning, which has been shown to effectively reduce the number of iterations. The current ChronGear solver in the POP has benefited from using a simple diagonal preconditioner (Pini and Gambolati, 1990; Reddy and Kumar, 2013). Some parallelizable methods such as polynomial, approximate-inverse, multigrid, and block preconditioning have drawn much attention recently. High-order polynomial preconditioning can reduce iterations as effectively as incomplete LU factorization in sequential simulations (Benzi, 2002). However, the computational overhead for the polynomial preconditioner typically offsets its superiority to the simple diagonal preconditioner (Meyer et al., 1989; Smith et al., 1992). The approximate-inverse preconditioner, although highly parallelizable, requires a linear system that is several times larger than the original system to be solved (Smith et al., 1992; Bergamaschi et al., 2007), which makes it less attractive for the POP.

The multigrid method is another well-known scalable and efficient approach to solve the elliptic systems and is commonly used as a preconditioner in sequential models. Recent works indicated that the geometric multigrid is promising in atmosphere and ocean modelling (Müller and Scheichl, 2014; Matsumura and Hasumi, 2008; Kanarska et al., 2007). However, the geometric multigrid in global ocean models does not always scale ideally because of the presence of complex topography, non-uniform or anisotropic grids (Fulton et al., 1986; Stüben, 2001; Tseng and Ferziger, 2003; Matsumura and Hasumi, 2008). The current POP, which employs general orthogonal girds to avoid a pole singularity, is a typical example. This leads to an elliptic system with variable coefficients defined on an irregular domain with non-uniform grids. The algebraic multigrid (AMG) is an alternative to the geometric multigrid for handling a complex topography. However, the AMG setup in the parallel environment is more expensive than the iterative solver in climate modelling, which makes it unfavourable as a preconditioner (Müller and Scheichl, 2014).

Block preconditioning has been shown to be an effective parallel preconditioner (Concus et al., 1985; White and Borja, 2011) and is appealing for the POP because it uses the block structure of the coefficient matrix that arises from the discretization of the elliptic equations. This advantage can further improve solver parallel performance. Some other algorithmic approaches also attempt to improve the parallel performance of ocean models. For example, a load-balancing algorithm based on the space-filling curve that not only eliminates land blocks but also reduces the communication overhead due to the reduced number of processes was proposed (Dennis, 2007; Dennis and Tufo, 2008). Beare and Stevens (1997) also proposed increasing the number of extra halos and communication overlaps in the parallel ocean general circulation. Although these approaches improve the performance of ocean models, the global communication bottleneck still exists.

To improve the scalability of the POP at high core counts, we abandon the CG-type approach and design a new barotropic solver that does not include global communication in iteration steps. The new barotropic solver, named P-CSI, uses a Classical Stiefel Iteration (CSI) method (proposed originally in Hu et al., 2015) with an efficient block preconditioner based on the Error Vector Propagation

(EVP) method (Roache, 1995). The P-CSI solver is now the default ocean barotropic solver for the upcoming CESM 2.0 release.

This paper is an extension of a work (Hu et al., 2015) originally presented at the 27th International Conference for High Performance Computing, Networking, Storage and Analysis (SC). Most of the audience at the SC conference are supercomputing specialists. Therefore, we simplified the background of the ocean model and focused on the design of the algorithm, scalability tests and efficiency in the SC paper. To expand the influence of our work to more climate modelers, we made some major changes in this paper as follows:

- After our presentation at the SC conference in November 2015, much helpful advice was gathered. Some specialists suggested that we should provide more information regarding the universal applicability of our new solver in different cases/applications. Therefore, we theoretically analysed the characteristics of P-CSI through the associated eigenvalues and their connection with the convergence rate here. The careful analysis also provided the main reasons why the proposed approach can lead to a significant improvement from many aspects, including the spectral radius of the matrix, condition number and the convergence rate. We showed that the P-CSI can converge as fast as the CG and ChronGear solvers which require additional global reductions. In the SC paper, we only presented the computational complexity which is not completed.
- We provided more comprehensive reviews of barotropic mode and the associated solvers adopted in the original POP. We believe that this addition will help other climate modelers to comprehensively understand the general large-scale computing problem in the POP, MOM, MITgcm, FVCOM, OPA models etc. The completed description helps the potential users to easily incorporate this approach to their own models. In the SC paper, we only presented a brief introduction of barotropic mode and the ChronGear solver.
- To help climate modelers, instead of computer scientists, to better understand the new solver, we rewrote most of the sentences and moved all pseudo-codes and the procedure of preconditioning into the appendix for interested readers. We also avoided the use of obscure computer jargon to make it more readable. Moreover, all figures have been redrawn to emphasize the advantage of our proposed method and the overall performance of the POP.

The remainder of this paper is organized as follows. Section 2 reviews the existing barotropic solver in the POP. Section 3 details the design of the P-CSI solver, and Section 4 contains an analysis of the computational complexity and convergence rate of P-CSI. Section 5 further compares the high-resolution performance of the existing solvers and the P-CSI solvers. Finally, conclusions are given in Section 6.

2 Barotropic solver background

We briefly describe the governing equations to formally derive the new P-CSI solver in the POP. The primitive momentum and continuity equations are expressed as:

$$\frac{\partial}{\partial t}\mathbf{u}+\mathcal{L}(\mathbf{u})+f\times\mathbf{u}=-\frac{1}{\rho_0}\nabla p+F_H(\mathbf{u})+F_V(\mathbf{u}), \tag{1}$$

$$\mathcal{L}(1)=0, \tag{2}$$

where $\mathcal{L}(\alpha)=\frac{\partial}{\partial x}(u\alpha)+\frac{\partial}{\partial y}(v\alpha)+\frac{\partial}{\partial z}(w\alpha)$, which is equivalent to the divergence operator when $\alpha=1$; x,y, and z are the horizontal and vertical coordinates; $\mathbf{u}=[u,v]^T$ is the horizontal velocity; w is the vertical velocity; f is the Coriolis parameter; p and ρ_0 represent the pressure and the constant reference water density, respectively; F_H and F_V are the horizontal and vertical dissipative terms, respectively (Smith et al., 2010). In particular, we emphasize the two-dimensional barotropic mode in the time-splitting scheme, where the P-CSI is implemented.

2.1 Barotropic mode

POP uses the splitting technique to solve the barotropic and baroclinic systems (Smith et al., 2010). All terms in Eq. (1) use the explicit scheme except the implicit treatment of barotropic mode and semi-implicit treatment of Coriolis and vertical mixing terms. The implicit treatment of barotropic mode is necessary to simulate fast gravity waves with a speed of $\sqrt{g*H}\approx 200m/s$ so that we can use the same time step as the baroclinic mode, which has a velocity scale of less than $2m/s$ (Hu et al., 2015). Solving the barotropic mode via an implicit method allows for a much larger time step. For instance, in the 0.1° POP model, using the implicit method, the time step is 172.8s ; otherwise, it would be 1.73s. The governing equations for the barotropic mode can be obtained by vertically integrating Eq. (1) and Eq. (2) from the ocean bottom topography to the sea surface:

$$\frac{\partial\mathbf{U}}{\partial t}=-g\nabla\eta+F, \tag{3}$$

$$\frac{\partial\eta}{\partial t}=-\nabla\cdot H\mathbf{U}+q_w, \tag{4}$$

where $\mathbf{U}=\frac{1}{H+\eta}\int_{-H}^{\eta}dz\mathbf{u}(z)\approx\frac{1}{H}\int_{-H}^{0}dz\mathbf{u}(z)$ is the vertically integrated barotropic velocity, g is the acceleration due to gravity, η is the sea surface height (defined as $p_s/\rho_0 g$, where p_s is the surface pressure associated with undulations of the free surface), H is the depth of the ocean bottom, q_w is the freshwater flux per unit area, and F is the vertical integral of all other terms except the time-tendency and surface pressure gradient in momentum Eq. (1). When we directly integrate the continuity equation from the bottom to the surface, we obtain a form $\int_{-H}^{\eta}dz(\nabla\cdot\mathbf{u}+\frac{\partial w}{\partial z})=\frac{\partial\eta}{\partial t}+\nabla\cdot(H+\eta)\mathbf{U}-q_w=0$ under the surface boundary condition $w(\eta)=\frac{d\eta}{dt}-q_w=\frac{\partial\eta}{\partial t}+\mathbf{u}(\eta)\cdot\nabla\eta-q_w$. The term including η inside the divergence leads to a nonlinear elliptic system; thus, many mature numerical methods such as the conjugate gradient method cannot handle this problem. To avoid this, the POP linearizes the continuity equation by dropping the divergence term in the boundary condition, which becomes $w(\eta) = \frac{\partial\eta}{\partial t} - q_w$. Equation (4) is the barotropic continuity equation which has been linearized; for more details, refer to (Smith et al., 2010).

Equation (3) and Eq. (4) are then discretized in time using an implicit scheme as follows:

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\tau} = -g\nabla\eta^{n+1} + F, \tag{5}$$

$$\frac{\eta^{n+1} - \eta^n}{\tau} = -\nabla\cdot H\mathbf{U}^{n+1} + q_w, \tag{6}$$

where τ is the time step associated with the time advance scheme. By replacing the barotropic velocity in Eq. (6) with the barotropic velocity at the next time step in Eq. (5), an elliptic system of sea surface height η is obtained

$$[-\nabla\cdot H\nabla + \frac{1}{g\tau^2}]\eta^{n+1} = -\nabla\cdot H[\frac{\mathbf{U}^n}{g\tau} + \frac{F}{g}] + \frac{\eta^n}{g\tau^2} + \frac{q_w}{g\tau}. \tag{7}$$

For simplicity, we can rewrite elliptic Eq. (7) as

$$[-\nabla\cdot H\nabla + \frac{1}{g\tau^2}]\eta^{n+1} = \psi(\eta^n, \tau), \tag{8}$$

where ψ represents a function of the current state of η.

Spatially, the POP utilizes the Arakawa B-grid on the horizontal grid (Smith et al., 2010) with the following nine-point stencils to discretize Eq. (8) as follows (see Fig. 1):

$$\nabla\cdot H\nabla\eta = \frac{1}{\Delta y}\delta_x\overline{[\Delta y H\delta_x\overline{\eta}^y]}^y + \frac{1}{\Delta x}\delta_y\overline{[\Delta x H\delta_y\overline{\eta}^x]}^x, \tag{9}$$

where δ_ξ ($\xi \in \{x, y\}$) are finite differences and Δ_ξ ($\xi \in \{x, y\}$) are the associated grid lengths. The finite difference $\delta_\xi(\psi)$ and average $\overline{\psi}^\xi$ notations are defined, respectively, as follows:

$$\delta_\xi\psi = [\psi(\xi + \Delta_\xi/2) - \psi(\xi - \Delta_\xi/2)]/\Delta_\xi, \tag{10}$$

$$\overline{\psi}^\xi = [\psi(\xi + \Delta_\xi/2) + \psi(\xi - \Delta_\xi/2)]/2. \tag{11}$$

Because the POP uses general orthogonal grids, the coefficient matrix varies in space. To demonstrate the properties of the sparse matrix used in the POP, we can simplify Eq. (9) using a special case with uniform grids as follows:

$$\begin{aligned}[\nabla\cdot H\nabla\eta]_{i,j} = -\frac{1}{S_{i,j}}[&B^O\overline{H}\eta_{i,j} + B^{NW}H_{i-1,j}\eta_{i-1,j+1} + \frac{1}{2}B^N(H_{i,j} + H_{i-1,j})\eta_{i,j+1} \\ &+ B^{NE}H_{i,j}\eta_{i+1,j+1} + \frac{1}{2}B^W(H_{i-1,j} + H_{i-1,j-1})\eta_{i-1,j} + \frac{1}{2}B^E(H_{i,j} + H_{i,j-1})\eta_{i+1,j} \\ &+ B^{SW}H_{i-1,j-1}\eta_{i-1,j-1} + \frac{1}{2}B^S(H_{i,j-1} + H_{i-1,j-1})\eta_{i,j-1} + B^{SE}H_{i,j-1}\eta_{i+1,j-1}],\end{aligned} \tag{12}$$

where $S_{i,j} = \Delta x\Delta y$ and $\overline{H} = \frac{1}{4}(H_{i,j} + H_{i-1,j} + H_{i,j-1} + H_{i-1,j-1})$; the H inside this equation is the ocean bottom depth in the columns of U-points (Smith et al., 2010). B terms are determined by using Δx and Δy:

$$\begin{aligned}
&\alpha = \frac{\Delta y}{\Delta x}, \quad \beta = 1/\alpha,\\
&B^{NW} = B^{NE} = B^{SW} = B^{SE} = -(\alpha+\beta)/4,\\
&B^{W} = B^{E} = (\beta-\alpha)/2,\\
&B^{N} = B^{S} = (\alpha-\beta)/2,\\
&B^{O} = \alpha+\beta.
\end{aligned} \tag{13}$$

To make the discretization of Eq. (8) more succinct, notations are introduced as follows:

$$\begin{aligned}
&A^{O}_{i,j} = B^{O}\overline{H},\\
&A^{N}_{i,j} = \frac{1}{2}B^{N}(H_{i,j}+H_{i-1,j}), A^{W}_{i,j} = \frac{1}{2}B^{W}(H_{i-1,j}+H_{i-1,j-1}),\\
&A^{E}_{i,j} = \frac{1}{2}B^{E}(H_{i,j}+H_{i,j-1}), A^{S}_{i,j} = \frac{1}{2}B^{S}(H_{i,j-1}+H_{i-1,j-1}),\\
&A^{NW}_{i,j} = B^{NW}H_{i-1,j}, A^{NE}_{i,j} = B^{NE}H_{i,j},\\
&A^{SW}_{i,j} = B^{SW}H_{i-1,j-1}, A^{SE}_{i,j} = B^{SE}H_{i,j-1},
\end{aligned} \tag{14}$$

These $A^{\chi}_{i,j}(\chi \in \mathcal{Q} = \{O, NW, NE, SW, SE, W, E, N, S\})$ are coefficients between a grid point (i,j) and its neighbours using the nine-point stencil discretization (9). The full discretization of Eq. (8) for any given grid point (i,j) can then be written as

$$\begin{aligned}
&(A^{O}_{i,j}+\phi)\eta_{i,j} + A^{NW}_{i,j}\eta_{i-1,j+1} + A^{N}_{i,j}\eta_{i,j+1} + A^{NE}_{i,j}\eta_{i+1,j+1} + A^{W}_{i,j}\eta_{i-1,j}\\
&+ A^{E}_{i,j}\eta_{i+1,j} + A^{SW}_{i,j}\eta_{i-1,j-1} + A^{S}_{i,j}\eta_{i,j-1} + A^{SE}_{i,j}\eta_{i+1,j-1} = S_{i,j}\psi_{i,j},
\end{aligned} \tag{15}$$

where $\phi = \frac{S_{i,j}}{g\tau^2}$ is a factor of the time step.

Therefore, elliptic Eq. (7) leads to a linear system of η, i.e., $Ax = b$, where A is a block tridiagonal matrix composed of coefficients $A^{\chi}_{i,j}(\chi \in \mathcal{Q})$. The simplified equation set of (13), (14) and (15) shows that A is mainly determined based on the horizontal grid sizes, ocean depth and time step. These impacts will be further discussed in Section 4.1. Note that Eq. (15) also indicates that the sparsity pattern of A comes directly from the nine nonzero elements in each row (Fig. 2).

POP evenly divides the horizontal domain into small blocks and distributes them to processes. We assume that there are N and M grids along the longitude and latitude respectively, and the global domain is divided into $n*m$ small blocks with a size of $\frac{N}{n} * \frac{M}{m}$. These blocks are distributed to processors using the simple Cartesian strategy or space-filling curve method (Smith et al., 2010).

2.2 Barotropic solvers

The barotropic solver in the original POP uses the PCG method with a diagonal preconditioner $M = \Lambda(A)$ because of its efficiency in small-scale parallelism (Dukowicz and Smith, 1994) (see

Appendix B1 for the details). To mitigate the global communication bottleneck, ChronGear, a variant of the CG method proposed by D'Azevedo et al. (1999), was later introduced as the default solver in the POP. It combines the two separated global communications of a single scalar into a single global communication (see Appendix B2). By this strategic rearrangement, the ChronGear method achieves a one-third latency reduction in the POP. However, the scaling bottleneck still exists in the high-resolution POP using this solver, particularly with a large number of cores (Fig. 3).

To accurately profile the parallel cost of the barotropic solvers, we clearly separate the timing for computation, halo exchange and global reduction. Operations such as scalar computations and vector scaling are categorized as pure computations, which are relatively cheap due to the independent operations on each process. The extra halo exchange is required for each process to update the boundary values from its neighbours (Fig. 1) after the matrix-vector multiplication. This halo exchange usually costs more than the computation when a large number of cores is used (due to a decreasing problem size per core). The global reduction, which is needed by the inner products of the vectors, is even more costly (Hu et al., 2013). Worley et al. (2011) and Dennis et al. (2012) specifically indicated that the global reduction in the POP's barotropic solver is the main scaling bottleneck for high-resolution ocean simulations.

Figure 3 confirms that the percentage of execution time for the barotropic mode in the 0.1° POP indeed increases with an increasing number of processor cores on Yellowstone. When 470 cores are used, the execution time of the barotropic (baroclinic) solver is approximately 5% (90%) of the total execution time (excludes initialization and I/O). However, when several thousand cores are used, the percentage of time spent in the baroclinic mode decreases, which is associated with the increasing percentage of time in the barotropic solver. With more than sixteen thousand cores, the percentage of the total execution time due to the barotropic solver is nearly 50%.

3 Design of the P-CSI solver

The CG-type solver converges rapidly in the sequential computation (Golub and Van Loan, 2012). However, the bottleneck of global communication embedded in ChronGear still limits the large-scale parallel performance. Here, we design a new solver targeted for reducing global communication so that the speed-up can be as close to unity as possible when a significant number of cores is used.

3.1 Classical Stiefel Iteration method

The CSI is a special type of Chebyshev iterative method (Stiefel, 1958). Saad et al. (1985) proposed a generalization of CSI on linearly connected processors and claimed that this approach outperforms the CG method when the eigenvalues are known. This method was revisited by Gutknecht and Röllin (2002) and shown to be ideal for massively parallel computers. In the procedure of preconditioned CSI (P-CSI; details are provided in Appendix B3), the iteration parameters, which control the searching directions in the iteration step, are derived from a stretched Chebyshev function of two extreme eigenvalues (Stiefel, 1958). We demonstrate in Section 4.2 that the stretched Chebyshev function in P-CSI provides a series of preset parameters for iteration directions. As a result, P-CSI requires no inner product operation, thus potentially avoiding the bottleneck of global reduction. This makes the P-CSI more scalable than ChronGear on massively parallel architectures. However, it requires *a priori* knowledge about the spectrum of coefficient matrix A (Gutknecht and Röllin, 2002). It is well known that obtaining the eigenvalues of a linear system of equations is equivalent to solving it. Fortunately, the coefficient matrix A and its preconditioned form in the POP are both positive definite real symmetric matrices. Approximate estimation of the largest and smallest eigenvalues, μ and ν, respectively, of the preconditioned coefficient matrix is sufficient to ensure the convergence of P-CSI.

To efficiently estimate the extreme eigenvalues of the preconditioned matrix $M^{-1}A$ (where M is the preconditioner), we adopt the Lanczos method (Paige, 1980) (see the algorithm in Appendix C). Initial tests indicate that only a small number of Lanczos steps is necessary to reasonably estimate the extreme eigenvalues of $M^{-1}A$ that result in near-optimal P-CSI convergence (Hu et al., 2015). Therefore, the computational overhead of the eigenvalue estimation is very small in our algorithm.

3.2 A block EVP preconditioner

Block preconditioning is quite promising in the POP because the parallel domain-decomposition is ideal for the block structure. A block preconditioning based on the EVP method is proposed and detailed in Hu et al. (2015); it improves improve the parallel performance of the barotropic solver in the POP. To the best of our knowledge, the EVP and its variants are among the least costly algorithms for solving elliptic equations in serial computation (Roache, 1995) and have also been used in several different Ocean models (Dietrich et al., 1987; Sheng et al., 1998; Young et al., 2012). The parallel EVP solver was also implemented by Tseng and Chien (2011). The standard EVP is actually a direct solver, which requires two solution steps: preprocessing and solving. In the preprocessing stage, the influence coefficient matrix and its inverse are computed, involving a computational complexity of $\mathcal{C}_{pre} = (2n-5)*9n^2 + (2n-5)^3 = \mathcal{O}(26n^3)$, which is intensive but computed only once at the beginning. The solving stage is computed at every time step and requires only $\mathcal{C}_{evp} = 2*9n^2 + (2n-5)^2 = \mathcal{O}(22n^2)$ (Hu et al., 2015), which is a much lower computational cost than those of other direct solvers, such as LU.

The EVP method is efficient for solving elliptic equations. Although EVP preconditioning may increase the required computation for each iteration, the barotropic solver can greatly benefit from the resulting reduction in the iteration number, particularly at very large numbers of cores when communication costs dominate (Hu et al., 2015). For large-scale parallel computing, a larger number of processors typically results in smaller domains, which in fact favours the application of the EVP method (Dietrich, 1975; Roache, 1995). If the domain size is too large without using domain decomposition, the computation will be very slow (see the complexity analysis in Section 4.3 when $p = 1$). Using parallel domain decomposition can actually help and speed up the EVP solver.

4 Algorithm analysis and comparison

The extreme eigenvalues of the coefficient matrix are critical to determine the convergence of the iterative solvers (such as P-CSI, PCG and ChronGear). Here, the characteristics of P-CSI are investigated in terms of the associated eigenvalues and their connection with the convergence rate. The computational complexity is also addressed.

4.1 Spectrum and condition number

Because the coefficient matrix A in the POP is symmetric and positive-definite (Smith et al., 2010), its eigenvalues are positive real numbers (Stewart, 1976). We assume that the spectrum (Golub and Van Loan, 2012) of A is $\mathcal{S} = \{\lambda_1, \lambda_2, \cdots, \lambda_N\}$, where $\lambda_{min} = \lambda_1 \le \lambda_i \le \lambda_{\mathcal{N}} = \lambda_{max}(1 < i < \mathcal{N}$, $\mathcal{N}$ is the size of A) are the eigenvalues of A. The condition number, defined as $\kappa = \lambda_{max}/\lambda_{min}$, is determined based on the spectral radius. Using the Gershgorin circle theorem (Bell, 1965), we know that for any $\lambda \in \mathcal{S}$, there exists a pair of (i, j) satisfying

$$|\lambda - (A^{O}_{i,j} + \phi)| \le \sum_{\chi \in \mathcal{Q}-\{O\}} |A^{\chi}_{i,j}|, \tag{16}$$

where $\phi = \frac{S}{g\tau^2}$ is defined in Section 2.1. With the definition of the coefficients in (13) and (14), we obtain

$$\begin{aligned} &\lambda_{max} \le (4\max(\alpha, \frac{1}{\alpha}) + \Phi)\max(H), \\ &\lambda_{min} \ge (2\min(\alpha - \frac{1}{\alpha}, \frac{1}{\alpha} - \alpha) + \Phi)\max(H). \end{aligned} \tag{17}$$

where $\Phi = \frac{\phi}{\max(H)}$, where $\max(H)$ is the maximal depth of the ocean bottom; for more details, refer to Appendix A,

To quantitatively evaluate the impacts of the condition number, we set up a series of idealized test cases to solve Eq. (8) in which the coefficient matrices are derived from Eq. (13), (14) and (15) on an idealized cylinder with an earth-sized perimeter, which is $2\pi R$ (radius R is 6,372 km), and a height of πR. A uniform grid with a size of $N \times M$ is used, where the grid size along the perimeter and height are $\Delta x = 2\pi R/N$ and $\Delta y = \pi R/M$, respectively. The depth H is set as a constant $4km$ to simplify the analysis.

The inequalities (17) suggest that the lower bound of the eigenvalues is mostly determined by Φ. If we assume that the grid aspect ratio is unity, we can rewrite $\Phi = \frac{S}{g\tau^2 H}$ as $\Phi = \frac{v^2}{gH(CFL)^2}$ in terms of the CFL number, where $CFL = \frac{v \cdot \Delta t}{\Delta x}$. This indicates that for a given ocean configuration and grid size, the lower bound of the eigenvalues will decrease with increasing CFL number, resulting in a larger condition number. Figure 4 shows the condition number versus the CFL number for three different velocities ($v = 2m/s$, $v = 20m/s$ and $v = 200m/s$). In the "$v = 200m/s$" case, Φ becomes very large and dominates both λ_{max} and λ_{min} when the CFL number is sufficiently small (smaller than 10^{-1} s) . As a result, the condition number is close to 1. When the CFL number is large enough (i.e., close to 5), the condition number is highly determined by the grid aspect ratio α because of the reduced impact of Φ.

When the aspect ratio of the horizontal grid cell is close to unity, the upper (lower) bound of the largest (smallest) eigenvalue decreases (increases), leading to a reduced spectral radius ($[\lambda_{min}, \lambda_{max}]$). This implies that the condition number is also reduced simultaneously. Figure 5 shows the condition number versus the aspect ratio, which is consistent with the theoretical bounds of the extreme eigenvalues in Eq. (17). As expected, the smallest condition number is found in Fig. 5 when the grid aspect ratio is close to unity regardless of the CFL number. When the aspect ratio equals unity (i.e., $\alpha = \frac{\Delta y}{\Delta x} = 1$), we obtain $\lambda_{max} \leq (4+\Phi)H$ and $\lambda_{min} \geq \Phi H$.

Our analysis suggests that the spectrum radius is confined in $(\Phi H, (4+\Phi)H)$ if the aspect ratio is unity regardless of grid sizes. However, the condition number may vary greatly because of the dependency on the grid size $\mathcal{N}$ and the aspect ratio. When the grid size $\mathcal{N}$ increases, the largest eigenvalue remains close to $4H$, whereas the smallest eigenvalue becomes closer to ΦH. Therefore, the condition number is significantly affected when the aspect ratio is far from unity. To focus on the impact of the number of grid points, we choose a constant aspect ratio $\alpha = 1$. Figure 6 shows that the condition number increases monotonically with increasing grid size for the five given different CFL conditions. It also shows that the CFL number has a large impact on the condition number.

In 0.1° realistic run, the CFL number is approximately $c \cdot \Delta t/\Delta x \approx 3.46$ (where $c = 200m/s$, $\Delta t = 172.8s$, and $\Delta x = 10000m$ are the typical gravity wave speed, time step and spatial resolution, respectively) and the condition number is approximately 250. Though the grid size of 0.1° POP is much larger than that of 1° POP, the condition number of 0.1° POP is smaller than the condition number of 1° POP (approximately 1200) owing to a smaller CFL number because of the small time step.

4.2 Convergence rate

The convergence rate of any elliptic solver relies heavily on the condition number of the preconditioned coefficient matrix A'. Both PCG and ChronGear have the same theoretical convergence rate because of the same numerical algorithm but different implementations (D'Azevedo et al., 1999). Their relative residual in the k-th iteration has an upper bound as follows (Liesen and Tichý, 2004):

$$\frac{||\mathbf{x}_k - \mathbf{x}^*||_{A'}}{||\mathbf{x}_0 - \mathbf{x}^*||_{A'}} \leq \min_{p \in \mathcal{P}_k, p(0)=1} \max_{\lambda \in \mathcal{S}} |p(\lambda)|, \tag{18}$$

where $\mathbf{x}_k$ is the solution vector after the k-th iteration, $\mathbf{x}^*$ is the solution of the linear equation (i.e., $\mathbf{x}^* = A^{-1}b$), λ represents an eigenvalue of A', and $\mathcal{P}_k$ is the vector space of polynomials with real coefficients and a degree less than or equal to k. Applying the Chebyshev polynomials of the first type to estimate this min-max approximation, we obtain

$$||\mathbf{x}_k - \mathbf{x}^*||_{A'} \leq 2(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^k ||\mathbf{x}_0 - \mathbf{x}^*||_{A'}, \tag{19}$$

where $\kappa = \kappa_2(A') = \frac{\lambda'_{max}}{\lambda'_{min}}$ is the condition number of matrix A' with respect to the l_2-norm. Equation (19) indicates that the theoretical bound of the convergence rate of PCG decreases with increasing condition number. PCG converges faster for a well-conditioned matrix (e.g., a matrix with a small condition number) than an ill-conditioned matrix.

We now show that the P-CSI has the same order of convergence rate as PCG and ChronGear with the additional advantage of fewer global reductions in parallel computing. With the estimated smallest and largest extreme eigenvalues of coefficient matrix ν and μ, the residual for the P-CSI algorithm satisfies

$$\mathbf{r}_k = P_k(A')\mathbf{r}_0, \tag{20}$$

where $P_k(\zeta) = \frac{\tau_k(\beta - \alpha\zeta)}{\tau_k(\beta)}$ for $\zeta \in [\nu, \mu]$ (Stiefel, 1958), $\alpha = \frac{2}{\mu - \nu}$ and $\beta = \frac{\mu+\nu}{\mu-\nu}$. $\tau_k(\xi)$ is a Chebyshev polynomial expressed as

$$\tau_k(\xi) = \frac{1}{2}[(\xi + \sqrt{\xi^2 - 1})^k + (\xi + \sqrt{\xi^2 - 1})^{-k}]. \tag{21}$$

when $\xi \in [-1, 1]$, the Chebyshev polynomial has an equivalent form

$$\tau_k(\xi) = cos(k \cos^{-1} \xi), \tag{22}$$

which clearly shows that $|\tau_k(\xi)| \leq 1$ when $|\xi| \leq 1$. $P_k(\zeta)$ is the polynomial satisfying

$$P_k = \min_{p \in \mathcal{P}_k, p(0)=1} \max_{\zeta \in [\nu, \mu]} |p(\zeta)|. \tag{23}$$

Assume that $A' = Q^T \Lambda Q$, where Λ is a diagonal matrix having the eigenvalues of A' on the diagonal and Q is a real orthogonal matrix with columns that are eigenvectors of A'. We then have

$$P_k(A') = Q^T P_k(\Lambda) Q = Q^T \begin{bmatrix} P_k(\lambda_1) & & & \\ & P_k(\lambda_2) & & \\ & & \ddots & \\ & & & P_k(\lambda_N) \end{bmatrix} Q. \tag{24}$$

Assuming that ν and μ satisfy $0 < \nu \leq \lambda_i \leq \mu$ $(i = 1, 2, \cdots, \mathcal{N})$, Eq. (22) indicates that $|\beta - \alpha\lambda_i| \leq 1$ and $|P_k(\lambda_i)| = \frac{\tau_k(\beta - \alpha\lambda_i)}{\tau_k(\beta)} \leq \tau_k^{-1}(\beta)$. Equations (20) and (24) indicate that

$$\frac{||\mathbf{r}_k||_2}{||\mathbf{r}_0||_2} \leq \tau_k^{-1}(\beta) = \frac{2(\beta + \sqrt{\beta^2 - 1})^k}{1 + (\beta + \sqrt{\beta^2 - 1})^{2k}} \leq 2(\frac{\sqrt{\kappa'}-1}{\sqrt{\kappa'}+1})^k, \tag{25}$$

where $\kappa' = \frac{\mu}{\nu}$. Equation (25) shows that P-CSI has the same theoretical upper bound of the convergence rate as PCG and ChronGear when the estimation of eigenvalues is appropriate (e.g., $\kappa' = \kappa$).

The foregoing analysis applies to cases in which a nontrivial preconditioning is used. Assume that the preconditioned coefficient matrix $A' = M^{-1}A$. Note that the preconditioned matrix in the PCG, ChronGear and P-CSI algorithms is actually $M^{-1/2}A(M^{-1/2})^T$, which is symmetric and has the same set of eigenvalues as $M^{-1}A$ (Shewchuk, 1994). Thus, the condition number of the preconditioned matrix is $\kappa = \kappa_2(M^{-1/2}A(M^{-1/2})^T)$, which is usually smaller than the condition number of A. The closer M is to A, the smaller the condition number of $M^{-1}A$ is. When M is the same as A, then $\kappa_2(M^{-1}A) = 1$.

Because the convergence rate of P-CSI is on the same order as that of PCG and ChronGear, the performance between P-CSI and the CG-type solvers should be comparable when a small number of cores is used. When a large number of cores is used for the high-resolution ocean model, P-CSI should be significantly faster than PCG or ChronGear per iteration due to the bottleneck in the CG-type method. This is shown in the following analysis of computational complexity.

4.3 Computational complexity

To analyse the computational complexity of P-CSI and compare it with ChronGear, we assume that p is the number of processes and $\mathcal{N}$ is the number of grid points following the same definition as in Hu et al. (2015). Both the ChronGear and P-CSI solver time can then be divided into three major components: computation $\mathcal{T}_c$, halo exchanging $\mathcal{T}_b$, and global communication $\mathcal{T}_g$. The complexity of computation varies among different solvers and preconditioners. The halo exchange complexity is $\mathcal{T}_b = \mathcal{O}(4\varpi + 8\sqrt{\frac{\mathcal{N}}{p}}\vartheta)$, where ϖ is the ratio of point-to-point communication latency per message to the time of one floating-point operation and ϑ is the ratio of the transfer time per byte (inverse of bandwidth) to the time of one floating-point operation. All halo exchange times show a similarly decreasing trend with increasing number of processes but have a lower bound of 4ϖ. The global communication exists only in the ChronGear solver and contains one global reduction per iteration, resulting from the MPI_Allreduce and a masking operation that excludes land points. The cost of the masking operation decreases with increasing processes p, whereas the cost of MPI_Allreduce monotonically increases; thus, the global reduction complexity satisfies $\mathcal{T}_g = \mathcal{O}(2\frac{\mathcal{N}}{p} + \log p\varpi)$.

The execution time of one diagonal preconditioned ChronGear solver step can then be expressed as:

$$\mathcal{T}_{cg} = \mathcal{K}_{cg}(\mathcal{T}_c + \mathcal{T}_b + \mathcal{T}_g) = \mathcal{O}(\mathcal{K}_{cg}(18\frac{\mathcal{N}}{p} + 8\sqrt{\frac{\mathcal{N}}{p}}\vartheta + (4 + \log p)\varpi), \tag{26}$$

where $\mathcal{K}_{cg}$ is the number of iterations, which does not change with the number of processes (Hu et al., 2015). The complexity of P-CSI with a diagonal preconditioner is

$$\mathcal{T}_{pcsi} = \mathcal{O}(\mathcal{K}_{pcsi}(12\frac{\mathcal{N}}{p} + 8\sqrt{\frac{\mathcal{N}}{p}}\vartheta + 4\varpi)), \tag{27}$$

where $\mathcal{K}_{pcsi}$ is the number of iterations.

Equation (26) indicates that the computation and halo exchange time decrease with increasing numbers of processes. However, the time required for the global reduction increases with increasing numbers of processes. Therefore, we can expect the execution time of the ChronGear solver to increase when the number of processors exceeds a certain threshold. Our analysis shows that P-CSI has a lower computational complexity than that of ChronGear due to the lack of a $\log p$ term associated with global communications.

We further consider the computational complexity of preconditioning. The EVP preconditioning has $\mathcal{O}(22\frac{\mathcal{N}}{p})$. Thus, with the EVP preconditioning, the computational complexity of ChronGear and P-CSI becomes $\mathcal{O}(39\frac{\mathcal{N}}{p})$ and $\mathcal{O}(33\frac{\mathcal{N}}{p})$, respectively. As a result, the total complexities of ChronGear and P-CSI with EVP preconditioning are

$$\mathcal{T}_{cg-evp} = \mathcal{O}(\mathcal{K}_{cg-evp}(39\frac{\mathcal{N}}{p} + 8\sqrt{\frac{\mathcal{N}}{p}}\vartheta + (4 + \log p)\varpi), \tag{28}$$

$$\mathcal{T}_{pcsi-evp} = \mathcal{O}(\mathcal{K}_{pcsi-evp}(33\frac{\mathcal{N}}{p} + 8\sqrt{\frac{\mathcal{N}}{p}}\vartheta + 4\varpi)). \tag{29}$$

Although the computation time in each iteration doubles with the EVP preconditioning, the total time may still decrease if the number of iterations is reduced. Indeed, with EVP preconditioning, the iteration number $\mathcal{K}_{pcsi-evp}$ decreases by almost one-half (see Fig. 8). As a result, the total number of communications, which is the most time-consuming part for a large number of cores, decreases by approximately one-half.

5 Numerical experiments

To evaluate the new P-CSI solver, we first demonstrate its characteristics and compare it with PCG (and thus ChronGear) using an idealized test case. The actual performance of P-CSI in the CESM POP is then evaluated and compared with that of the existing solvers using the 0.1° high-resolution simulation.

5.1 Condition number and convergence rate

To confirm the theoretical analysis of the convergence in Section 4.2, we created a series of matrices with the idealized setting illustrated in Section 4.1. Instead of a cylindrical grid, we choose a spherical grid with two polar continents (ocean latitude varies from 80°S to 80°N). A uniform latitude-longitude grid is used in which the grid size along the longitude varies with latitude coordinate θ, that is, $\Delta x = \pi R \cos\theta$. The time step size is set to $\tau = \frac{\Delta x}{v}$, where $v = 2m/s$ is the barotropic velocity of the ocean water, as used in Section 4.1. These cases differ with respect to the number of grid points; thus, the condition numbers vary. We compare the results using PCG and P-CSI solvers with no preconditioning, diagonal preconditioning or EVP preconditioning. Here, the block size in

EVP preconditioning is set as 5×5 and the convergence tolerance is $tol = 10^{-6}$. We note that the theoretical convergence rates of ChronGear and PCG are identical; thus, the results here can apply to the ChronGear simultaneously.

As shown in Fig. 7, when the problem size increases, the coefficient matrix becomes more poorly conditioned until it reaches the upper bound at the order of gH/v^2. All solvers must iterate more to obtain the same level of relative residual. For both PCG and P-CSI, the convergence rate varies with different preconditioners. Given the same problem size, the solvers without preconditioning need the largest number of iterations, while those using the EVP preconditioning require the fewest. This confirms that with the EVP preconditioning, the matrix becomes better conditioned than the matrix without preconditioning or with diagonal preconditioning. As shown in the previous section, the P-CSI has the same theoretical lower bound of the convergence rate as PCG and ChronGear when the estimation of extreme eigenvalues is appropriate ($k^{'} = k$). However, the P-CSI commonly has a slower convergence rate than that of PCG if the same preconditioning is applied (Fig. 7). Because the P-CSI requires that $0 < \nu < \lambda_i < \mu (i = 1, ..., N)$, which means that $k^{'} = \mu/\nu \geq \lambda_{max}/\lambda_{min} = k$, Eq. (19) and Eq. (25) suggest that the P-CSI will converge more slowly than the PCG unless the estimation of extreme eigenvalues is optimal. Furthermore, the theoretical bound is often too conservative for PCG as the problem size increases in application, which is not completely linear (known as superlinear convergence of the PCG method (Beckermann and Kuijlaars, 2001)). Note that the diagonal preconditioner only slightly improves the convergence in our idealized cases because of the uniform grid and the constant ocean depth configuration.

If the condition numbers are very large, any advanced preconditioner that can quickly reduce the iteration count will be very useful for improving performance. In fact, the EVP solver is a direct fast solver; thus, it is well suitable as the preconditioner within each block. It is also simple enough to effectively reduce the condition number of the coefficient matrix by approximately 5 times in both 1 and 0.1 degree cases, leading to an overall reduction of 2/3 iterations. Even so, further studies regarding the preconditioner in practical climate models will be very useful and will be our future work.

5.2 A practical application using the high-resolution CESM POP

5.2.1 Experiment platform and configuration

We evaluate the performance of P-CSI in CESM1.2.0 on the Yellowstone supercomputer, located at NCAR-Wyoming Supercomputing Center (NWSC) (Loft et al., 2015). Yellowstone uses Intel Xeon E5-2670 (Sandy Bridge: 16 cores @ 2.6 GHz, hyperthreading enabled, 20 MB shared L3 cache) and provides a total of 72,576 cores connected by a 13.6 GBps InfiniBand network. More than 50% of Yellowstone's cycles are consumed by CESM. Therefore, the ability to accelerate the parallel performance on Yellowstone is critical to support the CESM production simulations.

To emphasize the advantage of P-CSI, we use the finest 0.1° grid and a POP with 60 vertical levels with the "G_NORMAL_YEAR" configuration, which uses active ocean and sea ice components (i.e., the atmosphere and land components are replaced by pre-determined forcing data sets). The I/O optimization is another important issue for the high-resolution POP (Huang et al., 2014) but is not addressed here.

The choice of ocean block size and layout has a large impact on performance for the high-resolution POP because it directly affects the distribution of the workload among processors. To remove the influence of different block distributions on our results, we carefully specify block decompositions for each core with the same ratio. The time step is set to the default of 172.8 seconds. For a fair comparison among solvers, the convergence is checked every 10 iterations for all tests. The impacts of CSI and the EVP preconditioner are evaluated separately using several different numerical experiments.

5.2.2 Overall performance of P-CSI

This experiment is designed to illustrate the overall performance of P-CSI which is important in high-resolution production simulations. Figure 8 compares the convergence rate (relative residual versus the number of iterations) among different barotropic solvers with different preconditioners. The P-CSI converges slightly more slowly than PCG and ChronGear with the same diagonal preconditioner at the beginning and final iteration steps, which is related to the unstable distribution of the coefficient matrix's eigenvalues. However, the slopes are similar for all of these solvers, thus supporting the same upper bound of the convergence rate discussed in Section 4.2.

Figure 9 further evaluates the solver time for the different phases. P-CSI outperforms ChronGear primarily because it only requires a few global reductions in the convergence check. No significant difference can be found for the halo exchange and the computation phases when using large core counts except the evident reduced execution time of halo exchange with the EVP preconditioner. The reduction in global communications will also significantly reduce the sensitivity of the ocean model component to operating system noise (Ferreira et al., 2008) by increasing the time interval between global synchronizations.

According to Fig. 8, the P-CSI solver can reach the same relative residual using many fewer iterations with the EVP preconditioner. As a result, it reduces not only the execution time of global reduction but also the execution time of halo exchange owing to the reduced iterations which is illustrated in Fig. 9. All of these results are consistent with the theoretical analysis in Section 4.3. Note that the extra computation operations required by the EVP preconditioner have only a small impact on the overall performance of the barotropic solver.

The overall performance of P-CSI in a realistic 0.1° POP run is illustrated in Fig. 10. Using the EVP preconditioner, P-CSI can accelerate the barotropic calculation from 6.2 SYPD (Simulated Years Per wall-clock Day) to 10.5 SYPD on 16,875 cores. Dennis et al. (2012) indicated that 5 simulated years per wall-clock day is the minimum requirement to run long-term climate simulations. For the completed POP simulation, Fig. 10 indicates that the simulated timing of P-CSI achieves 10.5 simulated years per wall-clock day on 16,875 cores, whereas the timing of ChronGear with a diagonal preconditioner achieves only 6.2 simulated years per wall-clock day using the same number of cores. In Section 2, we demonstrated that the percentage of the POP execution time required by the barotropic solver increases with increasing number of cores using the original ChronGear solver. In particular, ChronGear with diagonal preconditioning accounts for approximately 50% of the total execution time on 16,875 cores (see Fig. 3). In contrast, Fig. 10 also shows that by using the scalable P-CSI solver, the barotropic calculation time constitutes only approximately 16% of the total execution time on 16,875 cores. Finally, based on an ensemble-based statistical method for the 1° POP, Hu et al. (2015) verified that the climate is not changed using our new solver.

6 Conclusions

We accelerated the high-resolution POP in the CESM framework by implementing a new P-CSI ocean barotropic solver. This new solver adopts a Chebyshev-type iterative method to avoid the global communication operations in conjunction with an effective EVP preconditioner to improve the parallel performance further. The superior performance of the P-CSI is carefully investigated using the theoretical analysis of the algorithm and computational complexity. Compared with the existing ChronGear solver, it significantly reduces the global reductions and realizes a competitive convergence rate. The proposed alternative has become the default barotropic solver in the POP within CESM and may greatly benefit other climate models.

7 Code availability

The present P-CSI solver v1.0 is available on https://zenodo.org/record/56705 and https://github.com/hxmhuang/PCSI. This solver is also included in the upcoming CESM public release v2.0. For the older CESM versions 1.2.x, the user should follow these steps indicated in the Readme.md file:
(1) Create a complete case or an ocean component case.
(2) Copy our files into the corresponding case path and build this case.
(3) Add two lines at the end of user_nl_pop2 file to use our new solver.
(4) Execute the preview_namelists file to activate the solver.
(5) Run the case.
The user are welcome to see the website mentioned above for more details and use the configuration files to repeat our experiments.

Appendix A: Estimation of extrem eigenvalues with variable ocean depth H

Rewrite the full discretization of Eq. (8) for any given grid point (i,j):

$$\begin{aligned}&(A^O_{i,j}+\phi)\eta_{i,j}+A^{NW}_{i,j}\eta_{i-1,j+1}+A^N_{i,j}\eta_{i,j+1}+A^{NE}_{i,j}\eta_{i+1,j+1}+A^W_{i,j}\eta_{i-1,j}\\&+A^E_{i,j}\eta_{i+1,j}+A^{SW}_{i,j}\eta_{i-1,j-1}+A^S_{i,j}\eta_{i,j-1}+A^{SE}_{i,j}\eta_{i+1,j-1}=S_{i,j}\psi_{i,j},\end{aligned}\tag{A1}$$

According to the Gershgorin circle theorem (Bell, 1965), we know that for any $\lambda\in\mathcal{S}$, there exists a pair of (i,j) satisfying

$$|\lambda-(A^O_{i,j}+\phi)|\leq\sum_{\chi\in\mathcal{Q}-\{O\}}|A^\chi_{i,j}|.\tag{A2}$$

The upper bound of eigenvalues can be deduced as follows

$$\begin{aligned}\lambda&\leq A^O_{i.j}+\phi+\sum_{\chi\in\mathcal{Q}-\{O\}}|A^\chi_{i,j}|\\&=2(\alpha+\beta)\overline{H}+2|\alpha-\beta|\overline{H}+\phi\\&=4\max(\alpha,\frac{1}{\alpha})\overline{H}+\phi\\&\leq(4\max(\alpha,\frac{1}{\alpha})+\Phi)\max(H)\end{aligned}\tag{A3}$$

The lower bound of eigenvalues can be deduced as follows

$$\begin{aligned}\lambda&\geq A^O_{i.j}+\phi-\sum_{\chi\in\mathcal{Q}-\{O\}}|A^\chi_{i,j}|\\&=-2|\alpha-\beta|\overline{H}+\phi\\&=2\min(\alpha-\beta,\beta-\alpha)\overline{H}+\phi\\&\geq(2\min(\alpha-\frac{1}{\alpha},\frac{1}{\alpha}-\alpha)+\Phi)\max(H)\end{aligned}\tag{A4}$$

Appendix B: Algorithms

B1 PCG algorithm

The procedure of PCG is shown as follows (Smith et al., 2010):

```
Initial guess: $\mathbf{x}_0$
Compute residual $\mathbf{r}_0=\mathbf{b}-\mathbf{A}\mathbf{x}_0$
Set $\mathbf{s}_0=0$, $\beta_0=1$
For $k=1,2,\cdots,k_{max}$ do
    1. $\mathbf{r}'_{k-1}=\mathbf{M}^{-1}\mathbf{r}_{k-1}$
    2. $\boxed{\beta_k=\mathbf{r}^T_{k-1}\mathbf{r}'_{k-1}}$
```

```
    3. $\mathbf{s}_k = \mathbf{r}'_{k-1} + (\beta_k/\beta_{k-1})\mathbf{s}_{k-1}$
    4. $\mathbf{s}'_k = \mathbf{A}\mathbf{s}_k$
    5. $\boxed{\alpha_k = \beta_k/(\mathbf{s}_k^T\mathbf{s}'_k)}$
    6. $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k\mathbf{s}_k$
    7. $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k\mathbf{s}'_k$
    8. convergence_check($\mathbf{r}_k$)
End Do
```

Operations such as β_k/β_{k-1} in line (3) are scalar computations, whereas $\alpha_k\mathbf{s}_k$ in line (6) are vector scalings. $\mathbf{A}\mathbf{s}_k$ in line (4) is a matrix-vector multiplication. Inner products of vectors are $\mathbf{r}_{k-1}^T\mathbf{r}'_{k-1}$ in line (2) and $\mathbf{s}_k^T\mathbf{s}'_k$ in line (5), these inner products use two global reduction operations.

B2 ChronGear algorithm

The procedure of ChronGear is shown as follows:

```
Initial guess: $\mathbf{x}_0$
Compute residual $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
Set $\mathbf{s}_0 = 0$, $\mathbf{p}_0 = 0$, $\rho_0 = 1$, $\sigma_0 = 0$
For $k = 1, 2, \cdots, k_{max}$ do
    1. $\mathbf{r}'_k = \mathbf{M}^{-1}\mathbf{r}_{k-1}$
    2. $\mathbf{z}_k = \mathbf{A}\mathbf{r}'_k$
    3. $\boxed{\rho_k = \mathbf{r}_{k-1}^T\mathbf{r}'_k}$
    4. $\boxed{\sigma_k = \mathbf{z}_k^T\mathbf{r}'_k - \beta_k^2\sigma_{k-1}}$
    5. $\beta_k = \rho_k/\rho_{k-1}$
    6. $\alpha_k = \rho_k/\sigma_k$
    7. $\mathbf{s}_k = \mathbf{r}'_k + \beta_k\mathbf{s}_{k-1}$
    8. $\mathbf{p}_k = \mathbf{z}_k + \beta_k\mathbf{p}_{k-1}$
    9. $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k\mathbf{s}_k$
    10. $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k\mathbf{p}_k$
```

```
    11. convergence_check($\mathbf{r}_k$)
End Do
```

The inner products in ρ_k and σ_k use two global reduction operations. However, these two global reductions can be combined into one operation thus halving the latency.

B3 P-CSI algorithm

The pseudocode of the P-CSI algorithm is shown as follows:

```
Initial guess: $\mathbf{x}_0$, estimated eigenvalue boundaries $[\nu,\mu]$
Set $\alpha = \frac{2}{\mu-\nu}$, $\beta = \frac{\mu+\nu}{\mu-\nu}$, $\gamma = \frac{\beta}{\alpha}$, $\omega_0 = \frac{2}{\gamma}$
Compute residual $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\Delta\mathbf{x}_0 = \gamma^{-1}\mathbf{M}^{-1}\mathbf{r}_0$, $\mathbf{x}_1 = \mathbf{x}_0 + \Delta\mathbf{x}_0$, $\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_1$
For $k = 1,2,\cdots,k_{max}$ do
    1. $\omega_k = 1/(\gamma - \frac{1}{4\alpha^2}\omega_{k-1})$
    2. $\mathbf{r}'_k = \mathbf{M}^{-1}\mathbf{r}_k$
    3. $\Delta\mathbf{x}_k = \omega_k\mathbf{r}'_k + (\gamma\omega_k - 1)\Delta\mathbf{x}_{k-1}$
    4. $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$
    5. $\mathbf{r}_{k+1} = \mathbf{b} - \mathbf{A}\mathbf{x}_{k+1}$
    6. convergence_check($\mathbf{r}_k$)
End Do
```

Appendix C: Eigenvalue Estimation

The procedure of the Lanczos method to estimate the extreme eigenvalues of the matrix $M^{-1}A$ is shown as follows:

```
Initial guess: $\mathbf{r}_0$
Set $\mathbf{s}_0 = \mathbf{M}^{-1}\mathbf{r}_0$; $\mathbf{q}_1 = \mathbf{r}_0/(\mathbf{r}_0^T\mathbf{s}_0)$; $\mathbf{q}_0 = \mathbf{0}$; $\beta_0 = 0$;   $\mu_0 = 0$; $T_0 = \emptyset$
For $j = 1,2,\cdots,m$ do
    1. $\mathbf{p}_j = \mathbf{M}^{-1}\mathbf{q}_j$
    2. $\mathbf{r}_j = \mathbf{A}\mathbf{p}_j - \beta_{j-1}\mathbf{q}_{j-1}$
    3. $\alpha_j = \mathbf{p}_j^T\mathbf{r}_j$
```

```
    4. $\mathbf{r}_j = \mathbf{r}_j - \alpha_j \mathbf{q}_j$
    5. $\mathbf{s}_j = \mathbf{M}^{-1}\mathbf{r}_j$
    6. $\beta_j = \mathbf{r}_j^T \mathbf{s}_j$
    7. if $\beta_j == 0$ then return
    8. $\mu_j = max(\mu_{j-1}, \alpha_j + \beta_j + \beta_{j-1})$
    9. $T_j = tri\_diag(T_{j-1}, \alpha_j, \beta_j)$
   10. $\nu_j = eigs(T_j, 'smallest')$
   11. if $|\frac{\mu_j}{\mu_{j-1}} - 1| < \epsilon$ and $|1 - \frac{\nu_j}{\nu_{j-1}}| < \epsilon$ then return
   12. $\mathbf{q}_{j+1} = \mathbf{r}_j / \beta_j$
End Do
```

In step (9), T is a tridiagonal matrix which contains $\alpha_j (j = 1, 2, ..., m)$ as the diagonal entries and $\beta_j (j = 1, 2, ..., m-1)$ as the off-diagonal entries.

$$T_m = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{m-1} \\ & & & \beta_{m-1} & \alpha_m \end{bmatrix}$$

Let ξ_{min} and ξ_{max} be the smallest and largest eigenvalues of T_m, respectively. Paige (1980) demonstrated that $\nu \le \xi_{min} \le \nu + \delta_1(m)$ and $\mu - \delta_2(m) \le \xi_{max} \le \mu$. As m increases, $\delta_1(m)$ and $\delta_2(m)$ will gradually converge to zero. Thus, the eigenvalue estimation of $M^{-1}A$ is transformed to solve the eigenvalues of T_m. Step (8) in eigenvalue estimation employs the Gershgorin circle theorem to estimate the largest eigenvalue of T_m, that is, $\mu = \max_{1 \le i \le m} \sum_{j=1}^{m} |T_{ij}| = \max_{1 \le i \le m}(\alpha_i + \beta_i + \beta_{i-1})$. The efficient QR algorithm (Ortega and Kaiser, 1963) with a complexity of $\Theta(m)$ is used to estimate the smallest eigenvalue ν in step (9).

Acknowledgements. This work is supported in part by a grant from the National Natural Science Foundation of China (41375102), and the National Grand Fundamental Research 973 Program of China (No. 2014CB347800). Computing resources were provided by the Climate Simulation Laboratory at NCAR's Computational and Information Systems Laboratory (sponsored by the NSF and other agencies).

References

Adcroft, A., Campin, J., Dutkiewicz, S., Evangelinos, C., Ferreira, D., Forget, G., Fox-Kemper, B., Heimbach, P., Hill, C., Hill, E., et al.: MITgcm user manual, 2014.

Beare, M. and Stevens, D.: Optimisation of a parallel ocean general circulation model, in: Annales Geophysicae, vol. 15, pp. 1369–1377, Springer, 1997.

Beckermann, B. and Kuijlaars, A. B. J.: Superlinear convergence of conjugate gradients, SIAM Journal on Numerical Analysis, 39, 300–329, 2001.

Bell, H. E.: Gershgorin's theorem and the zeros of polynomials, The American Mathematical Monthly, 72, 292–295, 1965.

Benzi, M.: Preconditioning techniques for large linear systems: a survey, Journal of Computational Physics, 182, 418–477, 2002.

Bergamaschi, L., Gambolati, G., and Pini, G.: A numerical experimental study of inverse preconditioning for the parallel iterative solution to 3D finite element flow equations, Journal of Computational and Applied Mathematics, 210, 64–70, 2007.

Bryan, F. O., Tomas, R., Dennis, J. M., Chelton, D. B., Loeb, N. G., and McClean, J. L.: Frontal scale air-sea interaction in high-resolution coupled climate models, Journal of Climate, 23, 6277–6291, 2010.

Chassignet, E. P. and Marshall, D. P.: Gulf Stream separation in numerical ocean models, Ocean Modeling in an Eddying Regime, pp. 39–61, 2008.

Concus, P., Golub, G., and Meurant, G.: Block preconditioning for the conjugate gradient method, SIAM Journal on Scientific and Statistical Computing, 6, 220–252, 1985.

D'Azevedo, E., Eijkhout, V., and Romine, C.: Conjugate gradient algorithms with reduced synchronization overhead on distributed memory multiprocessors, 1999.

Demory, M.-E., Vidale, P. L., Roberts, M. J., Berrisford, P., Strachan, J., Schiemann, R., and Mizielinski, M. S.: The role of horizontal resolution in simulating drivers of the global hydrological cycle, Climate Dynamics, 42, 2201–2225, 2014.

Dennis, J.: Inverse space-filling curve partitioning of a global ocean model, in: Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International, pp. 1–10, IEEE, 2007.

Dennis, J., Vertenstein, M., Worley, P., Mirin, A., Craig, A., Jacob, R., and Mickelson, S.: Computational performance of ultra-high-resolution capability in the Community Earth System Model, International Journal of High Performance Computing Applications, 26, 5–16, 2012.

Dennis, J. M. and Tufo, H. M.: Scaling climate simulation applications on the IBM Blue Gene/L system, IBM Journal of Research and Development, 52, 117 –126, doi:10.1147/rd.521.0117, 2008.

Dietrich, D.: Optimized Block-Implicit Relaxation, Journal of Computational Physics, 18, 421–439, 1975.

Dietrich, D. E., Marietta, M., and Roache, P. J.: An ocean modelling system with turbulent boundary layers and topography: Numerical description, International journal for numerical methods in fluids, 7, 833–855, 1987.

Dukowicz, J. K. and Smith, R. D.: Implicit free-surface method for the Bryan-Cox-Semtner ocean model, Journal of Geophysical Research: Oceans, 99, 7991–8014, doi:10.1029/93JC03455, http://dx.doi.org/10.1029/93JC03455, 1994.

Eyring, V., Bony, S., Meehl, G. A., Senior, C., Stevens, B., Stouffer, R. J., and Taylor, K. E.: Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organisation, Geoscientific Model Development Discussions, 8, 10 539–10 583, doi:10.5194/gmdd-8-10539-2015, http://www.geosci-model-dev-discuss.net/8/10539/2015/, 2015.

Ferreira, K. B., Bridges, P., and Brightwell, R.: Characterizing application sensitivity to OS interference using kernel-level noise injection, in: Proceedings of SC Conference, pp. 1–12, SC Conference, doi:10.1145/1413370.1413390, 2008.

Fulton, S. R., Ciesielski, P. E., and Schubert, W. H.: Multigrid methods for elliptic problems: A review, Monthly Weather Review, 114, 943–959, 1986.

Gent, P. R., Yeager, S. G., Neale, R. B., Levis, S., and Bailey, D. A.: Improvements in a half degree atmosphere/land version of the CCSM, Climate Dynamics, 34, 819–833, 2010.

Ghysels, P. and Vanroose, W.: Hiding Global Synchronization Latency in the Preconditioned Conjugate Gradient Algorithm, Parallel Computing, 40, 224–238, doi:10.1016/j.parco.2013.06.001, 2014.

Golub, G. H. and Van Loan, C. F.: Matrix computations, vol. 3, JHU Press, 2012.

Graham, T.: The importance of eddy permitting model resolution for simulation of the heat budget of tropical instability waves, Ocean Modelling, 79, 21–32, 2014.

Gutknecht, M. and Röllin, S.: The Chebyshev iteration revisited, Parallel Computing, 28, 263–283, 2002.

Hu, Y., Huang, X., Wang, X., Fu, H., Xu, S., Ruan, H., Xue, W., and Yang, G.: A scalable barotropic mode solver for the parallel ocean program, in: Euro-Par 2013 Parallel Processing, pp. 739–750, Springer, 2013.

Hu, Y., Huang, X., Baker, A. H., Tseng, Y.-h., Bryan, F. O., Dennis, J. M., and Yang, G.: Improving the scalability of the ocean barotropic solver in the community earth system model, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, p. 42, ACM, 2015.

Huang, X., Wang, W., Fu, H., Yang, G., Wang, B., and Zhang, C.: A fast input/output library for high-resolution climate models, Geoscientific Model Development, 7, 93–103, 2014.

Jones, P. W., Worley, P., Yoshida, Y., and White III, J. B.: Practical performance portability in the Parallel Ocean Program (POP), Concurrency and Computation: Practice and Experience, 17, 1317–1327, 2005.

Kanarska, Y., Shchepetkin, A., and McWilliams, J.: Algorithm for non-hydrostatic dynamics in the regional oceanic modeling system, Ocean Modelling, 18, 143–174, 2007.

Kuwano-Yoshida, A., Minobe, S., and Xie, S.-P.: Precipitation response to the Gulf Stream in an atmospheric GCM*, Journal of Climate, 23, 3676–3698, 2010.

Lai, Z., Chen, C., Cowles, G. W., and Beardsley, R. C.: A nonhydrostatic version of FVCOM: 1. Validation experiments, Journal of Geophysical Research: Oceans, 115, 2010.

Liesen, J. and Tichý, P.: Convergence analysis of Krylov subspace methods, GAMM-Mitteilungen, 27, 153–173, doi:10.1002/gamm.201490008, 2004.

Loft, R., Andersen, A., Bryan, F., Dennis, J. M., Engel, T., Gillman, P., Hart, D., Elahi, I., Ghosh, S., Kelly, R., Kamrath, A., Pfister, G., Rempel, M., Small, J., Skamarock, W., Wiltberger, M., Shader, B., Chen, P., and Cash, B.: Yellowstone: A Dedicated Resource for Earth System Science, in: Contemporary High Performance Computing: From Petascale Toward Exascale, Volume Two, edited by Vetter, J. S., vol. 2 of *CRC Computational Science Series*, p. 262, Chapman and Hall/CRC, Boca Raton, 1 edn., 2015.

Madec, G., Delecluse, P., Imbard, M., and Levy, C.: Ocean general circulation model reference manual, Note du Pôle de modélisation, 1997.

Matsumura, Y. and Hasumi, H.: A non-hydrostatic ocean model with a scalable multigrid Poisson solver, Ocean Modelling, 24, 15–28, 2008.

Meyer, P. D., Valocchi, A. J., Ashby, S. F., and Saylor, P. E.: A numerical investigation of the conjugate gradient method as applied to three-dimensional groundwater flow problems in randomly heterogeneous porous media, Water Resources Research, 25, 1440–1446, 1989.

Müller, E. H. and Scheichl, R.: Massively parallel solvers for elliptic partial differential equations in numerical weather and climate prediction, Quarterly Journal of the Royal Meteorological Society, 140, 2608–2624, 2014.

Ortega, J. and Kaiser, H.: The LLT and QR methods for symmetric tridiagonal matrices, The Computer Journal, 6, 99–101, 1963.

Pacanowsky, R. and Griffies, S.: The MOM3 manual, Geophysical Fluid Dynamics Laboratory/NOAA, Princeton, USA, p. 680, 1999.

Paige, C.: Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem, Linear Algebra and its Applications, 34, 235 – 258, doi:10.1016/0024-3795(80)90167-6, 1980.

Pini, G. and Gambolati, G.: Is a simple diagonal scaling the best preconditioner for conjugate gradients on supercomputers?, Advances in Water Resources, 13, 147–153, 1990.

Reddy, R. S. and Kumar, M. M.: Comparison of conjugate gradient methods and strongly implicit procedure for groundwater flow simulation, Journal of the Indian Institute of Science, 75, 667, 2013.

Roache, P. J.: Elliptic marching methods and domain decomposition, vol. 5, CRC press, 1995.

Roberts, M. J., Clayton, A., Demory, M.-E., Donners, J., Vidale, P. L., Norton, W., Shaffrey, L., Stevens, D., Stevens, I., Wood, R., et al.: Impact of resolution on the tropical Pacific circulation in a matrix of coupled models, Journal of Climate, 22, 2541–2556, 2009.

Saad, Y., Sameh, A., and Saylor, P.: Solving elliptic difference equations on a linear array of processors, SIAM Journal on Scientific and Statistical Computing, 6, 1049–1063, 1985.

Shaffrey, L. C., Stevens, I., Norton, W., Roberts, M., Vidale, P.-L., Harle, J., Jrrar, A., Stevens, D., Woodage, M. J., Demory, M.-E., et al.: UK HiGEM: the new UK high-resolution global environment model-model description and basic evaluation, Journal of Climate, 22, 1861–1896, 2009.

Sheng, J., Wright, D. G., Greatbatch, R. J., and Dietrich, D. E.: CANDIE: A new version of the DieCAST ocean circulation model, Journal of Atmospheric and Oceanic Technology, 15, 1414–1432, 1998.

Shewchuk, J. R.: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, Science (80-.)., 49, 64, doi:10.1.1.110.418, 1994.

Smith, R., Dukowicz, J., and Malone, R.: Parallel ocean general circulation modeling, Physica D: Nonlinear Phenomena, 60, 38–61, 1992.

Smith, R., Jones, P., Briegleb, B., Bryan, F., Danabasoglu, G., Dennis, J., Dukowicz, J., Fox-Kemper, C. E. B., Gent, P., Hecht, M., et al.: The Parallel Ocean Program (POP) Reference Manual Ocean Component of the Community Climate System Model (CCSM), 2010.

Stewart, J.: Positive definite functions and generalizations, an historical survey, Rocky Mountain J. Math, 6, 1976.

Stiefel, E. L.: Kernel polynomial in linear algebra and their numerical applications, in: Further contributions to the determination of eigenvalues, NBS Applied Math. Ser., 49, 1–22, 1958.

Stüben, K.: A review of algebraic multigrid, Journal of Computational and Applied Mathematics, 128, 281–309, 2001.

Tseng, Y.-h. and Chien, M.-h.: Parallel Domain-decomposed Taiwan Multi-scale Community Ocean Model (PD-TIMCOM), Computers & Fluids, 45, 77–83, 2011.

Tseng, Y.-h. and Ferziger, J. H.: A ghost-cell immersed boundary method for flow in complex geometry, Journal of computational physics, 192, 593–623, 2003.

Wehner, M. F., Reed, K. A., Li, F., Bacmeister, J., Chen, C.-T., Paciorek, C., Gleckler, P. J., Sperber, K. R., Collins, W. D., Gettelman, A., et al.: The effect of horizontal resolution on simulation quality in the Community Atmospheric Model, CAM5. 1, Journal of Advances in Modeling Earth Systems, 2014.

White, J. A. and Borja, R. I.: Block-preconditioned Newton–Krylov solvers for fully coupled flow and geomechanics, Computational Geosciences, 15, 647–659, 2011.

Worley, P. H., Mirin, A. A., Craig, A. P., Taylor, M. A., Dennis, J. M., and Vertenstein, M.: Performance of the community earth system model, in: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11, pp. 54:1–54:11, ACM, New York, NY, USA, doi:10.1145/2063384.2063457, 2011.

Young, C.-C., Tseng, Y.-H., Shen, M.-L., Liang, Y.-C., Chen, M.-H., and Chien, C.-H.: Software development of the TaIwan Multi-scale Community Ocean Model (TIMCOM), Environmental Modelling and Software, 38, 214–219, 2012.

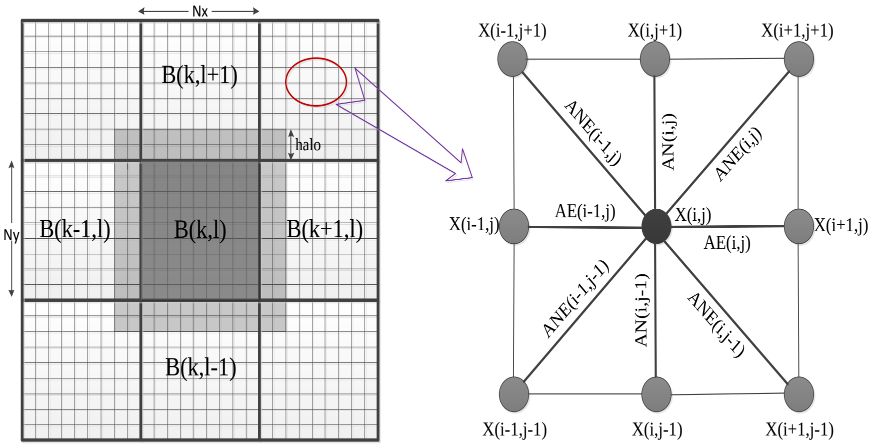

Figure 1. Grid domain decomposition of the ocean model component in CESM.

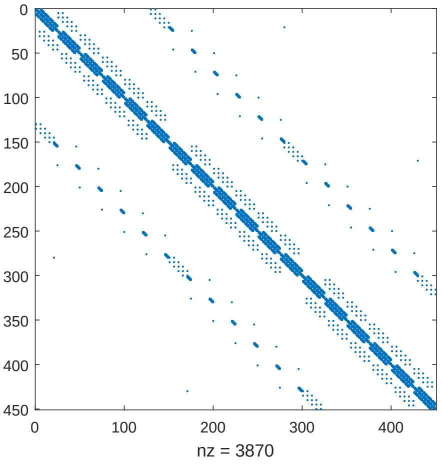

Figure 2. Sparsity pattern of the coefficient matrix in the case with 30×15 grids using nine-point stencils.

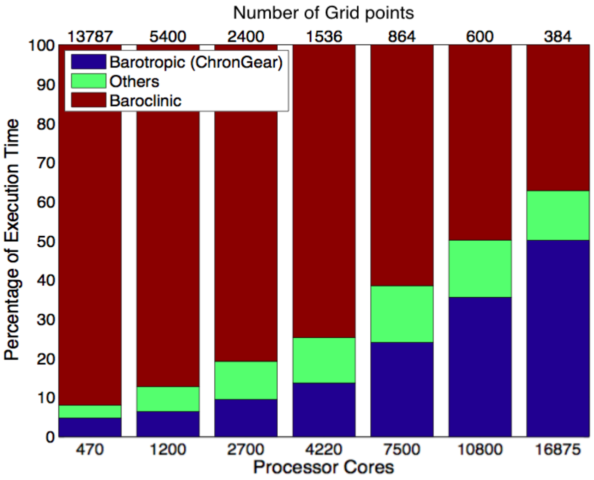

Figure 3. Number of grid points per processor and percentage of execution time in 0.1° POP using the default diagonal-preconditioned ChronGear solver on Yellowstone.

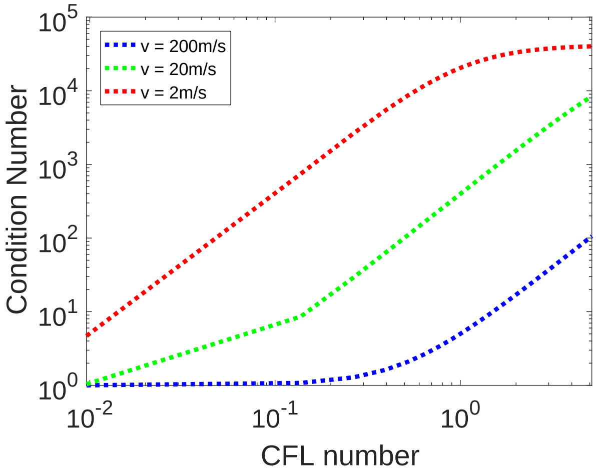

Figure 4. Relationship between the CFL number and the condition number of the coefficient matrix under the condition of different velocities, where the CFL number varies from 10^{-2} to 5.

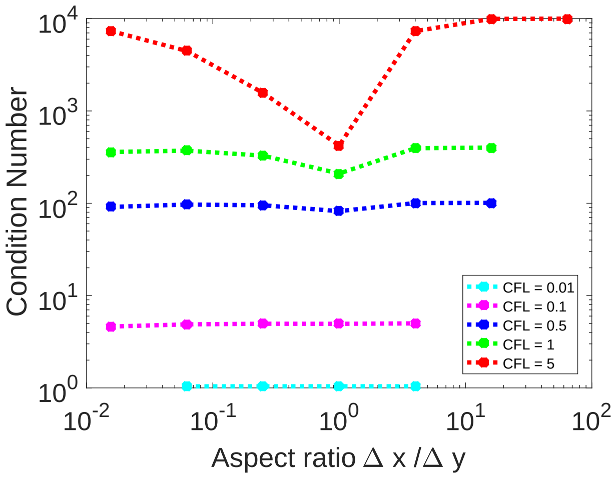

Figure 5. Relationship between aspect ratio and the condition number of the coefficient matrix under the condition of different typical CFL numbers.

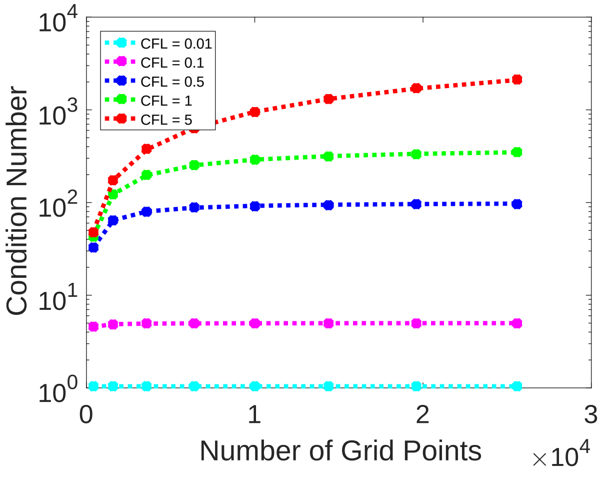

Figure 6. When the aspect ratio is constant $\alpha = 1$, relationship between the number of grid points and the condition number of the coefficient matrix under the condition of different typical CFL numbers.

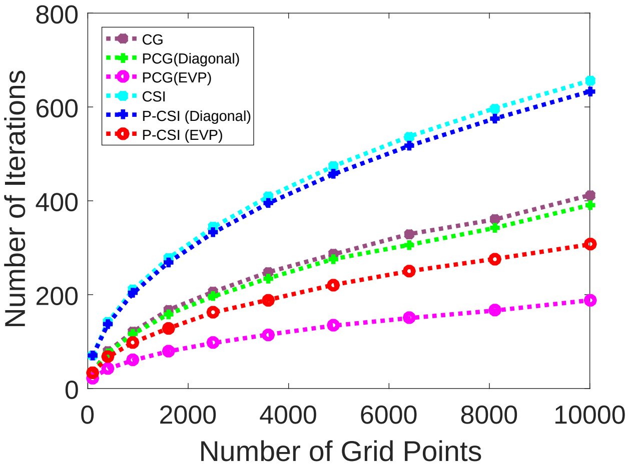

Figure 7. Relationship between grid sizes and number of iterations of different solvers in test cases with the idealized configuration.

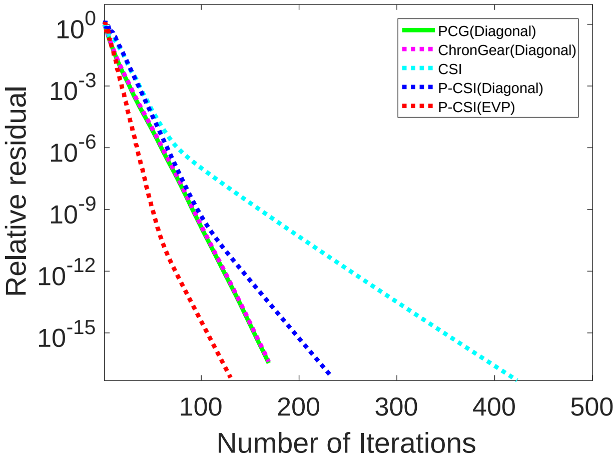

Figure 8. The convergence rate of different barotropic solvers with diagonal preconditioner and the convergence rate of CSI solver with different preconditioners in the 0.1° POP on Yellowstone.

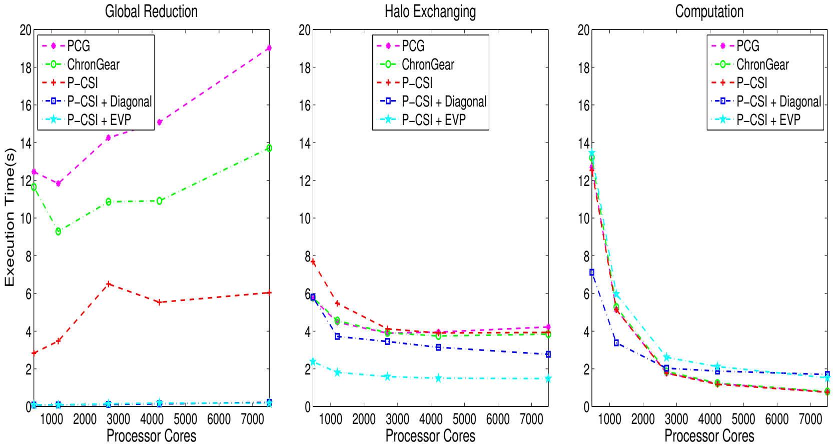

Figure 9. The execution time for different phases using different barotropic solvers and the execution time for different phases with different preconditioners in the P-CSI solver in 0.1° POP.

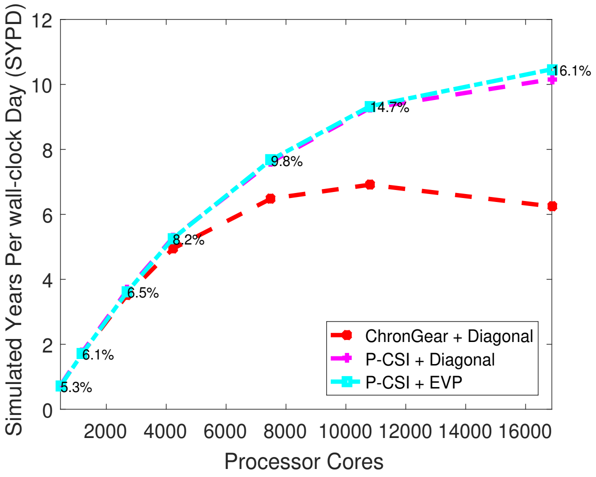

Figure 10. The simulated speed of the 0.1° ocean model component using different barotropic solver. The numbers on the dotted line represent the percentage of execution time spent in barotropic mode with P-CSI(EVP) using different number of processor cores. Information about the number of grid points per processor can refer to Fig. 3.