Computers & Fluids 110 (2015) 88–95

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Mixed order discretization based two-level Schwarz preconditioners for a tracer transport problem on the cubed-sphere

Haijian Yang [a], Chao Yang [b,c,*], Xiao-Chuan Cai [d]

[a] College of Mathematics and Econometrics, Hunan University, Changsha, Hunan 410082, China
[b] Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[c] State Key Laboratory of Computer Science, Chinese Academy of Sciences, Beijing 100190, China
[d] Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA

ARTICLE INFO

Article history:
Received 25 October 2013
Accepted 1 May 2014
Available online 14 May 2014

Keywords:
Tracer transport problem
Cubed-sphere mesh
Finite volume scheme
Fully implicit method
Multilevel domain decomposition
Strong and weak scalabilities

ABSTRACT

Solving tracer transport problems on the sphere at high resolution is important for atmospheric science. In this paper, we study a fully implicit method that allows larger time step size than explicit methods. The key to the success of the implicit approach is how to solve the discretized linear system at each time step. We investigate a parallel Krylov–Schwarz approach based on two discretizations, and test the proposed algorithm with two benchmark test cases. We show numerically that the solver scales well in terms of both strong and weak scalabilities on a supercomputer with thousands of processors.

1. Introduction

The tracer transport problem plays an important role in global atmospheric modeling [5]. Its numerical simulation at high resolution is very demanding in terms of computational resources. In recent years, there have been numerous efforts to develop advanced numerical methods for solving global atmospheric problems [7,9–11,14,20,21]. Among these, the fully implicit method has become an interesting research focus because the time step size is no longer constrained by the stability condition. The main challenge to implicit methods is how to solve the large system of equations.

In this paper, we study a Krylov–Schwarz method [2,12] to efficiently solve the linear system arising from the implicit finite volume discretization of a tracer transport problem. We use a class of two-level restricted Schwarz methods based on mixed order discretizations, in which the coarse preconditioner is based on a first-order spatial discretization while a second-order scheme is used to construct the preconditioner on the fine mesh. Our experiments show that this strategy leads to better parallel performance than using the second-order discretization on both levels.

* Corresponding author at: Institute of Software, Chinese Academy of Sciences, Beijing 100190, China. Tel./fax: +86 10 62661632.
E-mail addresses: haijianyang@gmail.com (H. Yang), yangchao@iscas.ac.cn (C. Yang), cai@cs.colorado.edu (X.-C. Cai).

http://dx.doi.org/10.1016/j.compfluid.2014.05.002
0045-7930/

The rest of the paper is organized as follows. In Section 2, we present a fully implicit discretization scheme for the tracer transport problem on the cubed-sphere. Section 3 focuses on the details of the two-level Schwarz preconditioning method. Numerical results for two benchmark test cases are provided in Section 4 to illustrate the accuracy and the parallel performance of the proposed methods. We end the paper with some concluding remarks in Section 5.

2. Finite volume discretization of the tracer transport problem

Let $\mathbb{S}$ be the surface of the sphere that is a closed domain without boundary and T denote the final time of the simulation. In the absence of source or sink, we write the tracer transport equation [5] defined on $\mathbb{S} \times (0, T]$ as

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{V}\phi) = 0, \tag{1}$$

where ϕ is the transported scalar, $\mathbf{V}$ is a given two-dimensional velocity vector, and ∇ is the gradient operator. We discretize the surface of the sphere using the cubed-sphere mesh, which is mapped to the sphere from the six faces of an inscribed cube Ω based on a gnomonic projection [9,10,14]. Let $(\xi, \eta) \in [-\pi/4, \pi/4] \times [-\pi/4, \pi/4]$ be the curvilinear coordinates on the cubed-sphere. Then (1) defined on each of the six faces of Ω can be written as

$$\frac{\partial \phi}{\partial t}+\frac{1}{\Lambda}\left(\frac{\partial}{\partial \xi}(\Lambda v^1\phi)\right)+\frac{1}{\Lambda}\left(\frac{\partial}{\partial \eta}(\Lambda v^2\phi)\right)=0, \tag{2}$$

where $\Lambda=(\sec\xi\sec\eta)^2\Big/\sqrt{(1+\tan^2\xi+\tan^2\eta)^3}$ and (v^1,v^2) is the contravariant coordinates of $\mathbf{V}$; more details can be found in [10,13].

In the study, a cell-centered finite volume method is used to discretize (2) in the spatial domain Ω, which consists of the six faces Ω^k $(k=1,2,\ldots,6)$. Let each Ω^k be covered with $N\times N$ mesh cells. Then each cell $\Omega^k_{i,j}=[\xi_{i-1/2},\xi_{i+1/2}]\times[\eta_{j-1/2},\eta_{j+1/2}]$ is centered at the position $\xi_i=-\pi/4+ih$ and $\eta_j=-\pi/4+jh$ with $i,j=1,\ldots,N$, and $h=\pi/(2N)$. The approximate solution $Q^k_{i,j}$ in the cell $\Omega^k_{i,j}$ is defined as

$$Q^k_{i,j}=\frac{1}{h^2\Lambda_{i,j}}\int_{\eta_{i+1/2}}^{\eta_{i-1/2}}\int_{\xi_{i+1/2}}^{\xi_{i-1/2}}\Lambda(\xi,\eta)\phi(\xi,\eta,t)d\Omega^k_{i,j},$$

where $\Lambda_{i,j}$ is computed at the cell center on each face. We integrate (2) over $\Omega^k_{i,j}$ and obtain the following semi-discrete system:

$$\frac{\partial Q^k_{i,j}}{\partial t}+\frac{F_{i+1/2,j}-F_{i-1/2,j}}{\Lambda_{i,j}h}+\frac{G_{i,j+1/2}-G_{i,j-1/2}}{\Lambda_{i,j}h}=0,$$

where the numerical fluxes $F_{i\pm1/2,j}$ and $G_{i,j\pm1/2}$ defined on the four cell boundaries are approximated as

$$F_{i\pm1/2,j}=\frac{\Lambda_{i\pm1/2,j}}{h}\int_{\eta_{j+1/2}}^{\eta_{j-1/2}}v^1(\xi_{i\pm1/2},y)\phi(\xi_{i\pm1/2},y)dy,$$

$$G_{i,j\pm1/2}=\frac{\Lambda_{i,j\pm1/2}}{h}\int_{\xi_{i+1/2}}^{\xi_{i-1/2}}v^2(x,\eta_{j\pm1/2})\phi(x,\eta_{j\pm1/2})d\xi,$$

respectively. A local Lax–Friedrichs flux formula is employed for approximating $F_{i\pm1/2,j}$ and $G_{i,j\pm1/2}$:

$$\tilde{f}(Q^-,Q^+)=[(f(Q^-)+f(Q^+))-\alpha(Q^+-Q^-)]/2,$$

where α is the maximum absolute value of the normal velocity along each cell boundary, and Q^- and Q^+ are the reconstructed states of Q on the cell boundary.

Two approaches are implemented in our methods for the reconstruction of the left and right states. The first one uses a piecewise constant method by

$$\begin{cases} Q^-_{i-1/2,j}=Q_{i-1,j},\\ Q^+_{i-1/2,j}=Q_{i,j},\\ Q^-_{i+1/2,j}=Q_{i,j},\\ Q^+_{i+1/2,j}=Q_{i+1,j}, \end{cases} \tag{3}$$

which leads to a first-order method [21]. To achieve a higher order of accuracy, we use a second-order reconstruction by

$$\begin{cases} Q^-_{i-1/2,j}=Q_{i-1,j}+(Q_{i,j}-Q_{i-2,j})/4,\\ Q^+_{i-1/2,j}=Q_{i,j}-(Q_{i+1,j}-Q_{i-1,j})/4,\\ Q^-_{i+1/2,j}=Q_{i,j}+(Q_{i+1,j}-Q_{i-1,j})/4,\\ Q^+_{i+1/2,j}=Q_{i+1,j}-(Q_{i+2,j}-Q_{i,j})/4. \end{cases} \tag{4}$$

We can similarly obtain Q^- and Q^+ in the η-direction. On the cubed-sphere mesh, values near face boundaries should be correctly passed to couple the patches together. One layer of ghost cells is used and the numerical fluxes are calculated symmetrically across the interface [10,11,19,23].

3. Fully implicit discretization and domain decomposition preconditioning

A semi-discrete system is obtained after applying the spatial discretization to (2),

$$\frac{\partial Q^k_{i,j}}{\partial t}=L\left(Q^k_{i,j}\right). \tag{5}$$

To integrate (5) in time, we employ a second-order backward differentiation formula with a fixed time step size $\triangle t$ and obtain

$$\frac{3Q^{k,m}_{i,j}-4Q^{k,m-1}_{i,j}+Q^{k,m-2}_{i,j}}{2\triangle t}=L\left(Q^{k,m}_{i,j}\right), \tag{6}$$

where $Q^{k,m}_{i,j}$ is the numerical solution at the m^{th} time step. A first-order backward Euler method is applied at the first time step.

A major advantage of fully implicit schemes is that the time step size $\triangle t$ is no longer constrained by the Courant–Friedrichs–Lewy (CFL) condition, which is often required by explicit or semi-implicit methods. But the price to pay in using implicit methods for the tracer transport problem is to solve a linear system (6) at each time step. In this study we use a Krylov–Schwarz method to fix this issue. More precisely speaking, we use a right-preconditioned Krylov subspace method (GMRES) [12] as the linear solver, and an overlapping one-level or two-level Schwarz method [2,4,18,22] as the preconditioner. The rest of the paper will focus on the use of one-level and two-level overlapping Schwarz preconditioners.

To define an overlapping restricted additive Schwarz preconditioner, we first partition the computational domain Ω into N_p non-overlapping subdomains Ω_l for $l=1,2,\ldots,N_p$, such that

$$\overline{\Omega}=\bigcup_{l=1}^{N_p}\overline{\Omega_l},$$

where N_p is the number of subdomains and also the number of processor cores. Then within Ω we extend each subdomain Ω_l with δ layers of mesh cells to a larger subdomain Ω_l^δ that overlaps with its neighbors. In each overlapping subdomain, we define a local subdomain matrix A_l that is the restriction of the global matrix A to Ω_l^δ with the restriction operator R_l^δ, i.e.,

$$A_l=R_l^\delta A(R_l^\delta)^T.$$

Here, R_l^δ and $(R_l^\delta)^T$ are the restriction and prolongation operators, respectively. Given a global vector defined on all mesh cells in Ω, R_l^δ restricts the global vector to a local vector that is defined only on the mesh cells within the overlapping subdomain Ω_l^δ, while $(R_l^\delta)^T$ prolongates the restricted vector back to a vector defined on the whole domain Ω with zeros filled to the unknowns corresponding to mesh cells outside Ω_l^δ. The one-level restricted additive Schwarz (RAS) preconditioner is defined as

$$M_h^{-1}=\sum_{l=1}^{N_p}(R_l^0)^T(A_l)^{-1}R_l^\delta, \tag{7}$$

where the restriction operator R_l^0 is defined as the restriction to the non-overlapping subdomain Ω_l, and the matrix–vector multiplication with $(A_l)^{-1}$ is obtained approximately by an incomplete LU (ILU) factorization.

Next we define a two-level Schwarz preconditioner by combining the one-level additive Schwarz preconditioner M_h^{-1} on the fine mesh with a coarse mesh preconditioner M_H^{-1} in a multiplicative manner. Let A_h (A_H) be the global matrix on the fine (coarse) mesh and $\mathcal{I}_H^h$ an interpolation operator from the coarse mesh to the fine mesh. Similarly, $\mathcal{I}_h^H$ denotes a restriction operator from the fine mesh to the coarse mesh. Then a multiplicative type two-level Schwarz preconditioner [15,16,18] is defined as

$$M_{two}^{-1}=B_H^{-1}+M_h^{-1}-M_h^{-1}A_hB_H^{-1}, \tag{8}$$

where $B_H^{-1}=\mathcal{I}_H^hA_H^{-1}\mathcal{I}_h^H$. More precisely speaking, the application of the preconditioner $y=M_{two}^{-1}r$ is obtained in the following steps:

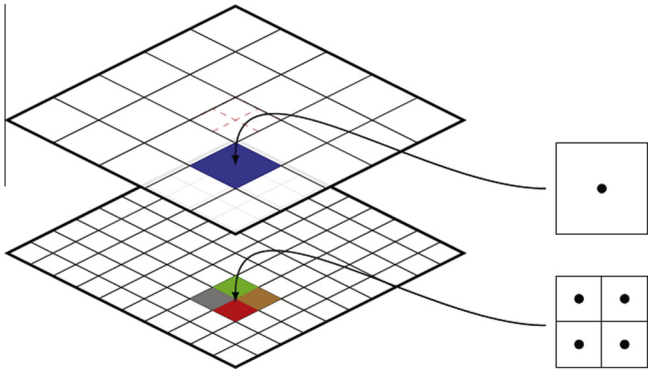


Fig. 1. Partitions for one face of the cubed-sphere with the coarse-to-fine mesh ratio 1:2. The bottom (upper) panel is the partition of the fine (coarse) mesh.

$$r_H = \mathcal{I}_h^H r, \tag{9}$$

$$A_H u_H = r_H, \tag{10}$$

$$u_h = \mathcal{I}_H^h u_H, \tag{11}$$

$$y = u_h + M_h^{-1}(r - A_h u_h). \tag{12}$$

In (10), a smaller linear system associated with A_H is then solved. Here, we solve the coarse problem iteratively by using GMRES with the one-level RAS preconditioner (7), i.e., $u_H = A_H^{-1} r_H$ is computed by approximately solving the following problem

$$\|r_H - A_H M_H^{-1} M_H u_H\| \leqslant \eta_r^c \|r_H\|,$$

where η_r^c is the solver tolerance for the iteration on the coarse mesh and M_H^{-1} is the one-level RAS preconditioner (7) on the coarse mesh.

On the coarse level, (10) is expensive to solve when the mesh size is fine. The situation becomes worse when a high order discretization is used, because the resulting matrix has considerably more non-zero elements. On the other hand, the coarse solve is used to improve the convergence of the iterative solver, it may not need to be solved as accurately as on the fine mesh. Based on these considerations, we propose a mixed order discretization based two-level Schwarz preconditioner. The approach consists of two components: we solve the coarse problem based on the subdomain matrix with the first-order spatial discretization (3) for the construction of the preconditioner M_H^{-1}, while the second-order scheme (4) is used to build the matrix A_H for the residual calculation.

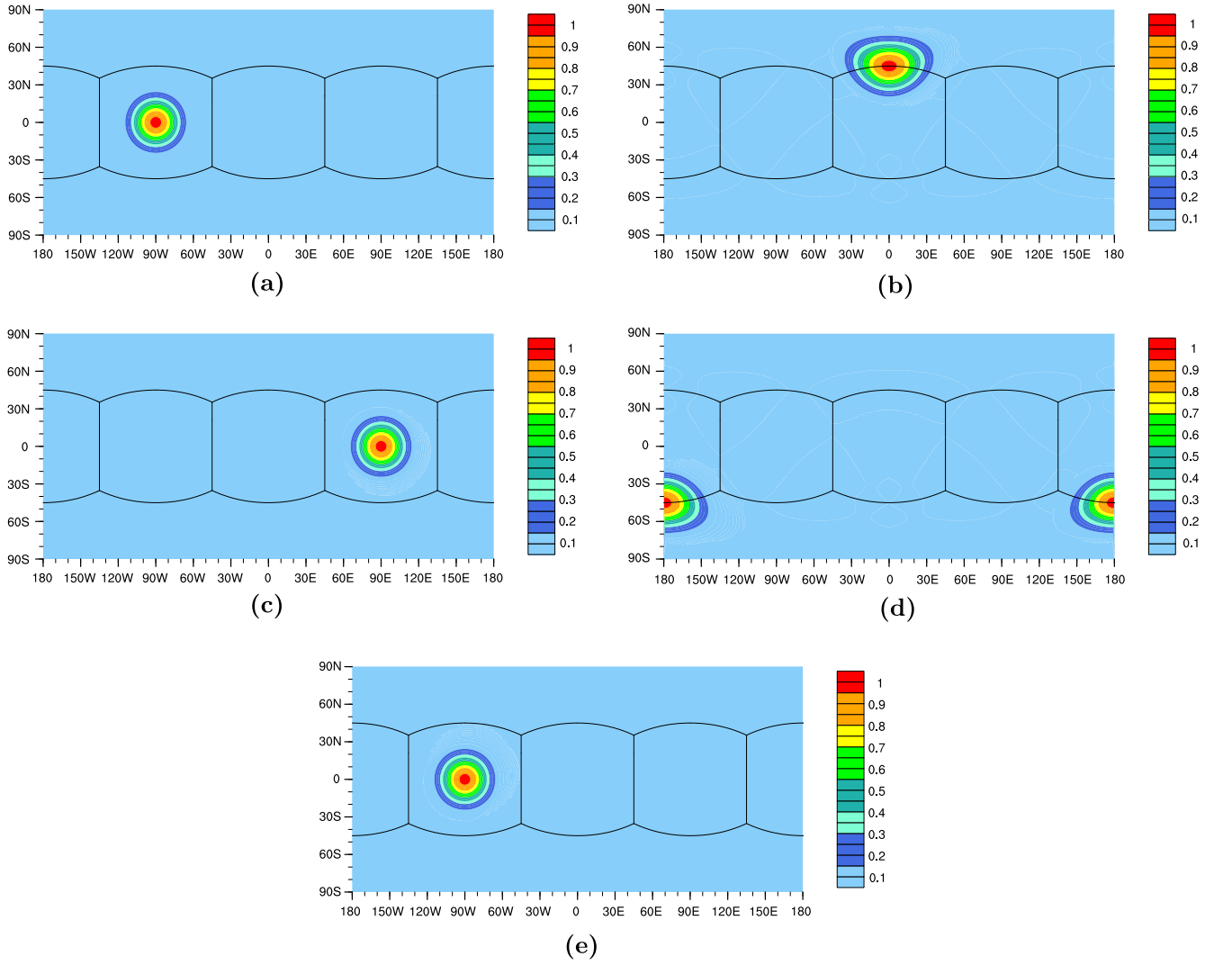


Fig. 2. Contour plots of **Case-1**. The problem is solved on a $512 \times 512 \times 6$ mesh. (a) the initial scalar field ϕ_0; (b) the numerical solution at day three; (c) the numerical solution at day six; (d) the numerical solution at day nine; and (e) the numerical solution at day twelve.

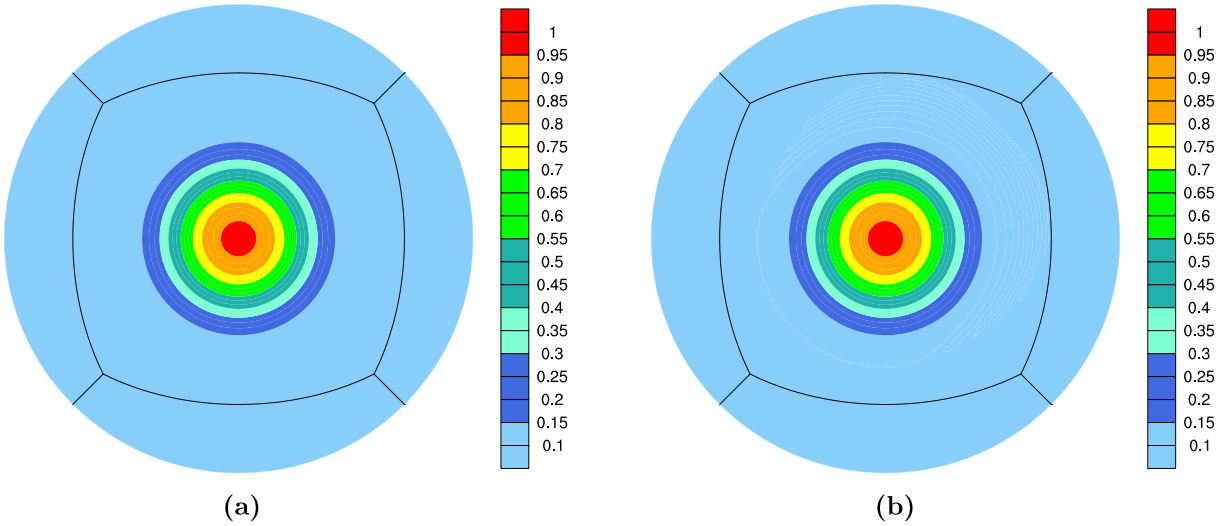


Fig. 3. Contour plots of **Case-1** on the sphere. The problem is solved on a $512 \times 512 \times 6$ mesh. (a) the initial scalar field ϕ_0 and (b) the numerical solution at the final time.

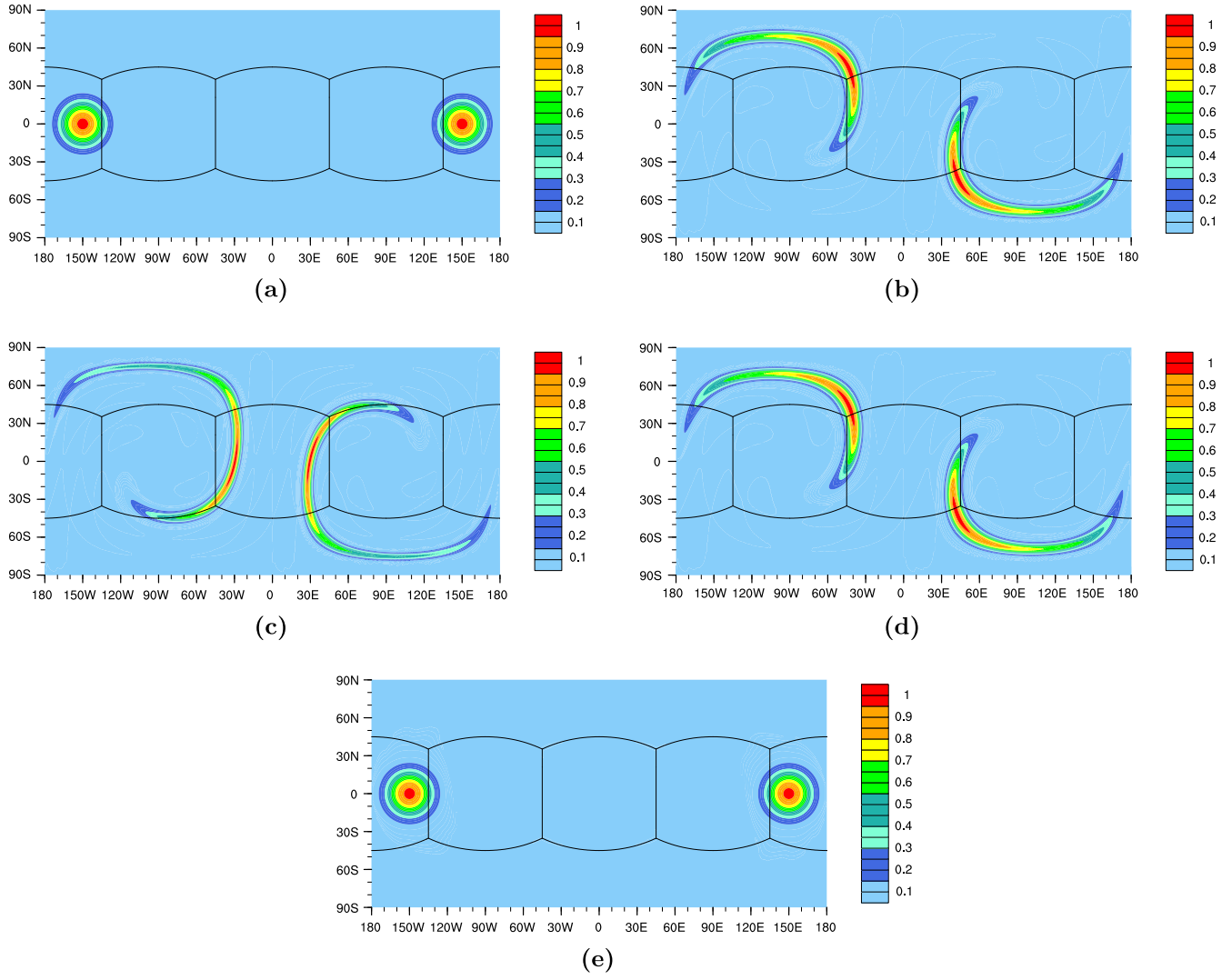


Fig. 4. Contour plots of **Case-2**. The problem is solved on a $512 \times 512 \times 6$ mesh. (a) the initial scalar field ϕ_0; (b) the numerical solution at day three; (c) the numerical solution at day six; (d) the numerical solution at day nine; and (e) the numerical solution at day twelve.

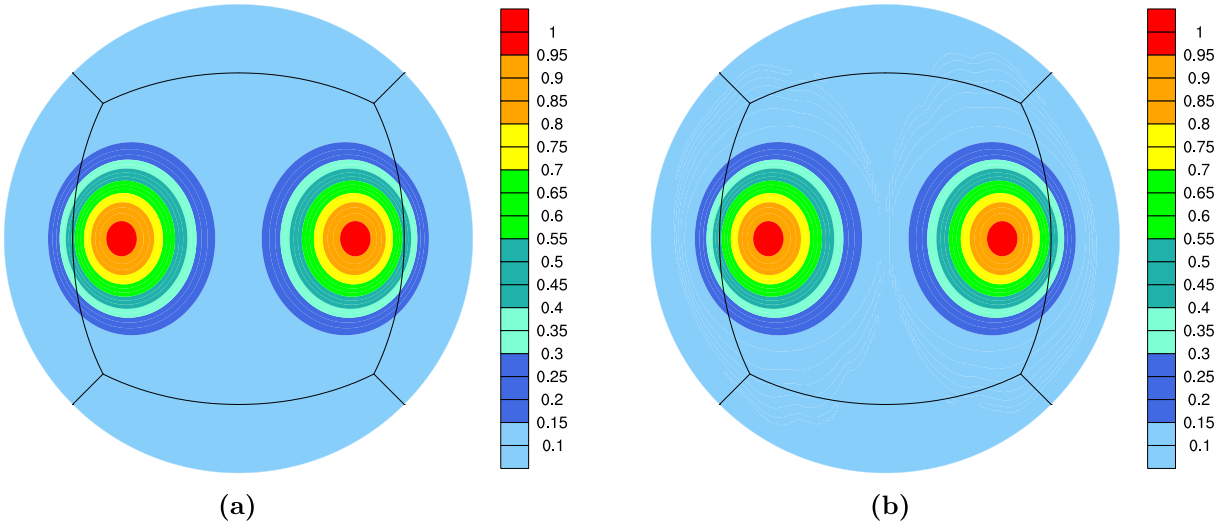
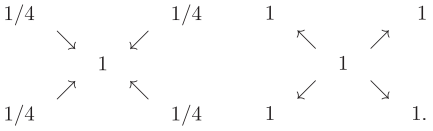


Fig. 5. Contour plots of **Case-2** on the sphere. The problem is solved on a $512 \times 512 \times 6$ mesh. (a) the initial scalar field ϕ_0 and (b) the numerical solution at the final time.

Table 1
Effect of overlapping size δ for **Case-2** with a fixed $4096 \times 4096 \times 6$ mesh. The time step size is fixed to $\triangle t = 0.2$ days, the simulation is stopped at day four. The calculation is carried out with 768 processors for the one-level and two-level preconditioners.

δ	One-level		$H = 2h$		$H = 4h$		$H = 8h$	
	GMRES	Time	GMRES	Time	GMRES	Time	GMRES	Time
0	86.3	226.1	29.7	178.1	48.3	172.4	64.0	191.1
1	73.0	220.8	25.6	191.7	42.3	175.3	57.1	205.0
2	72.4	215.8	24.9	192.2	42.5	171.5	57.4	197.3
4	71.5	213.5	24.8	194.6	41.7	180.0	56.2	202.3
6	70.3	219.2	24.6	198.8	41.6	184.1	55.4	213.0
8	70.0	229.2	24.4	205.7	41.4	193.1	54.8	222.9

Another consideration in designing the two-level method is how to formulate the restriction matrix $\mathcal{I}_h^H$ in (9) and the interpolation matrix $\mathcal{I}_H^h$ in (11). With the cell-centered finite volume scheme, it happens that a coarse mesh point does not coincide with any fine mesh point; see Fig. 1. We find that the overall method may not converge, if the operators are not chosen properly. In this paper, we build $\mathcal{I}_h^H$ and $\mathcal{I}_H^h$ for the case of the coarse-to-fine mesh ratio 1:2, using the following weights



The weights for other coarse-to-fine mesh ratios are defined in a similar way.

4. Numerical results

The algorithms described in the previous sections are implemented based on the Portable Extensible Toolkit for Scientific computation (PETSc) [1]. All computations are performed on a Dell PowerEdge C6100 supercomputer located at the University of Colorado Boulder. Each node contains 24 GB local memory and two hex-core 2.8 Ghz Intel Westmere processors. The nodes are interconnected via a non-blocking QDR Infiniband high performance network. We focus on the parallel performance of the proposed methods, including the effect on using different parameters in the preconditioner, the robustness of the method when the time step size is large, and the strong and weak scalabilities of the algorithm.

4.1. Test cases

In this subsection, we introduce two published benchmark problems that are used later for numerical tests.

4.1.1. ***Case-1***: *Solid-body advection of a cosine bell*

The first test case describes a steady and non-divergent wind field that transports a tracer field around a great circle without distortion [6,3,7,17]. The velocity field is set as

$$\begin{cases} u(\lambda,\theta,t) = u_0(\cos\alpha\cos\theta + \sin\alpha\cos\lambda\sin\theta), \\ v(\lambda,\theta,t) = -u_0\sin\alpha\sin\lambda, \end{cases}$$

where λ is the longitude, θ is the latitude, u_0 is the reference wind velocity, and α is the angle between the polar axis and the axis of the solid body rotation ($\alpha = 0$ is for pure zonal flow and $\alpha = \pi/2$ is for over-the-pole flow). Here, we use $\alpha = \pi/4$, which is

Table 2
Effect of time step size $\triangle t$ for **Case-2** with a fixed $4096 \times 4096 \times 6$ mesh. The simulation is stopped at day four. The calculation is carried out with 768 processors for the one-level and two-level preconditioners.

$\triangle t$	Steps	One-level		$H = 2$ h		$H = 4$ h		$H = 8$ h	
		GMRES	Time	GMRES	Time	GMRES	Time	GMRES	Time
0.025	160	19.4	879.9	10.2	856.1	12.5	745.5	14.5	854.5
0.05	80	34.0	569.6	14.8	544.3	19.5	453.4	24.3	532.2
0.1	40	53.8	392.8	20.6	334.3	31.1	293.6	41.8	343.4
0.5	8	110.3	116.8	38.0	104.6	57.8	87.7	86.5	109.2
1.0	4	199.8	102.9	84.5	89.3	105.0	67.2	153.2	81.6

Table 3
Effect of different number of processors N_p for **Case-1** and **Case-2** with a fixed $4096 \times 4096 \times 6$ mesh. The time step size is fixed to $\triangle t = 0.2$ days, the simulation is stopped at day four.

N_p	One-level		$H = 2$ h		$H = 4$ h		$H = 8$ h	
	GMRES	Time	GMRES	Time	GMRES	Time	GMRES	Time
Case-1								
384	134.4	803.9	31.6	585.6	66.0	603.8	102.7	778.3
768	133.6	408.0	32.3	242.8	67.3	272.3	103.5	358.8
1536	134.5	234.5	31.9	144.6	67.6	159.8	106.7	223.4
3072	137.1	144.3	32.4	101.3	67.7	110.0	107.3	129.0
Case-2								
384	75.4	409.7	25.5	344.3	39.8	308.9	54.6	353.2
768	73.0	228.9	25.6	191.7	42.3	175.3	57.1	205.0
1536	79.4	143.7	24.6	109.2	43.4	109.0	58.1	128.7
3072	79.4	93.9	25.7	82.6	45.5	79.5	59.7	88.3

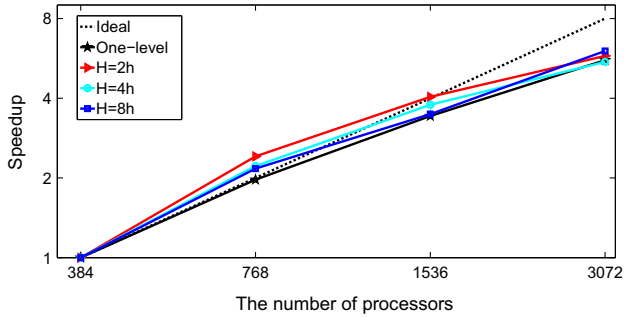


Fig. 6. Strong scalability results for **Case-1** with a fixed $4096 \times 4096 \times 6$ mesh. The time step size is fixed to $\triangle t = 0.2$ days, the simulation is stopped at day four.

challenging for the test on the cubed-sphere, and $u_0 = 2\pi/T$ with the finial time $T = 12$ days means that a full rotation around the earth finishes every twelve days. A cosine bell is defined as:

$$\phi_0(\lambda,\theta) = \begin{cases} 0.1 + 0.9h_0(\lambda,\theta) & r_0 < r, \\ 0.1 & \text{otherwise}, \end{cases} \tag{13}$$

where $r = 1/2$ is the base radius of the bells and

$$h_0(\lambda,\theta) = \frac{1}{2}(1 + \cos(\pi r_0/r)), \quad \text{if } r_0 < r,$$

with r_0 being the great-circle distance between (λ,θ) and $(\lambda_0,\theta_0) = (-\pi/2, 0)$:

$$r_0(\lambda,\theta) = \arccos(\sin\theta_0 \sin\theta + \cos\theta_0 \cos\theta(\lambda - \lambda_0)).$$

The solid-body rotation of a cosine bell along a great-circle trajectory is a widely used benchmark test, which has a final solution that is identical to the initial condition. Numerical solutions are obtained by using the implicit method, as shown in Figs. 2 and 3. We find that the final solution is in good agreement with the initial condition obtained by our method.

4.1.2. ***Case-2***: *Advection of the double cosine bells*

The second test case describes cosine bells with a strongly deformational non-divergent flow [6,8]. The two cosine bells (13) are centered at $(\lambda_0,\theta_0) = (5\pi/6, 0)$ and $(-5\pi/6, 0)$, respectively. The zonal and meridional velocity components in the flow are

$$\begin{cases} u(\lambda,\theta,t) = u_0 \sin^2(\lambda)\sin(2\theta)\cos(\pi t/T), \\ v(\lambda,\theta,t) = u_0 \sin(2\lambda)\cos(\theta)\cos(\pi t/T), \end{cases}$$

respectively, where the final time $T = 12$ days and $u_0 = 10/T$. In the simulation, the two cosine bells follow the complex trajectories of the flow and undergo severe deformation. Moreover, the flow reverses its course at half-time and the scalar fields return to their initial positions and shapes at the final time, which makes the exact solution available at the end of the simulation, as shown in Figs. 4 and 5.

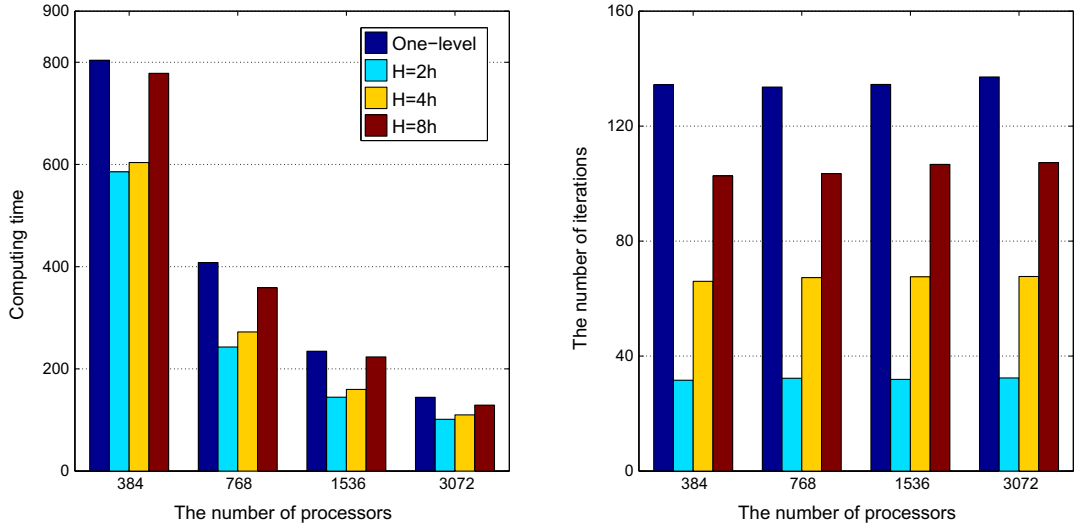


Fig. 7. Strong scalability results for **Case-1** with different number of processors N_p. The mesh is $4096 \times 4096 \times 6$.

4.2. Performance and parallel scalability

The parallel performance of the domain decomposition preconditioner is investigated. In the one-level method, the second-order discretization is used for the residual calculation and the construction of the RAS preconditioner. In the two-level method, both the second-order and the first-order methods are evaluated using the algorithm introduced in Section 3, the former is used for the residual calculation on both the fine and coarse levels, and the latter is used to build the RAS preconditioner on the coarse level. Moreover, in the two-level preconditioner, the size of the coarse mesh has a strong impact on the efficiency and parallel scaling of the method, using a relatively fine coarse mesh helps reduce the total number of linear iterations, but at the same time the total compute time may increase. An important implementation detail to consider in designing the two-level method is to balance the effects of preconditioning and the computing time of the coarse solve. Let h be the size of the fine mesh and H the size of the coarse mesh. In the experiments, we compare the following algorithms using different coarse-to-fine mesh ratios:

- *One-level*: one-level additive Schwarz method (7) is used as the preconditioner.
- $H = 2h$: two-level multiplicative Schwarz method (8) is used as the preconditioner with the coarse-to-fine mesh ratio 1:2.
- $H = 4h$: two-level multiplicative Schwarz method (8) is used as the preconditioner with the coarse-to-fine mesh ratio 1:4.
- $H = 8h$: two-level multiplicative Schwarz method (8) is used as the preconditioner with the coarse-to-fine mesh ratio 1:8.

In the tests, GMRES(30) and FGMRES(30) are used to solve the linear systems on the coarse and the fine meshes, respectively. We use 10^{-5} as the absolute stopping condition for all linear solves, except for the coarse solve of the two-level preconditioner, for which we use 10^{-1} as the relative stopping tolerance. The subdomain problems are solved with a sparse ILU factorization with four levels of fill-ins.

Throughout this subsection, "GMRES" denotes the average number of iterations per time step, and "Time" denotes the total computing time in seconds.

4.2.1. Impact of the overlapping parameter

For the Schwarz preconditioner, the overlapping size δ is an important parameter that influences the effect of the preconditioner. From the theory of the overlapping Schwarz method, larger overlap often implies a faster convergence in terms of GMRES iterations. However, larger overlap also means larger subdomain problems and more information transfers between subdomains; as a result, the overall compute time may increase. Table 1 shows the effect of the overlapping parameter for **Case-2**. We observe from the table that the methods converge better as the overlap increases and on the other hand a larger overlap may also increases interprocess communications.

4.2.2. Impact of time step sizes

Next, to show the unconditional stability of the fully implicit scheme, we solve the same problem with gradually increased time step size $\triangle t$. Table 2 shows the number of iterations and the computing time of the fully implicit method with respect to different time step sizes for **Case-2**. As Δt increases, the total number of GMRES iterations increases, while the computing time deceases due to the reduced number of time steps. Since we use a fully implicit method in which the time step size is no longer constrained by the CFL condition, the results clearly indicate that the one-level and two-level methods are convergent for even very large value of Δt.

4.2.3. Impact of the number of processors

Besides the robustness with respect to the overlapping parameter δ and time step sizes $\triangle t$, scalability is another important concern in the design of solution algorithms for the tracer transport problem. As mentioned in [15,16], the domain decomposition method is still not well understood for hyperbolic problems such as the tracer transport problem, due to the lack of ellipticity. Therefore we numerically investigate the strong and weak scalabilities of the proposed algorithms for **Case-1** and **Case-2** in terms of the computing time and GMRES iterations. We first study the strong scalability of the methods by using a fixed mesh $4096 \times 4096 \times 6$ and also a fixed time step size $\triangle t = 0.2$ days in Table 3. We see that the algorithms are scalable with respect to the number of processors, the number of GMRES iterations varies mildly with different N_p. In Fig. 6, we report the average computing time and the average number of GMRES iterations per time step with respect to the number of processors. It is important to note that the timing results obtained by using the two-level preconditioners are always better than the results obtained with the one-level preconditioner (see Fig. 7).

We further test our algorithms in terms of the weak scalability with a fixed 192×192 mesh per processor. We refine the mesh and increase the number of processors simultaneously to keep

Table 4
Weak scalability results for **Case-1** and **Case-2** with a fixed 192×192 mesh per processor. The time step size is fixed to $\triangle t = 0.2$ days, the simulation is stopped at day four. The total degree of freedom for the largest simulation is $4224 \times 4224 \times 6 = 107,053,056$.

N_p	Mesh	One-level		$H = 2$ h		$H = 4$ h		$H = 8$ h	
		GMRES	Time	GMRES	Time	GMRES	Time	GMRES	Time
Case-1									
24	$384 \times 384 \times 6$	10.9	16.9	6.8	15.3	8.7	14.9	9.7	15.3
96	$768 \times 768 \times 6$	18.1	20.6	9.7	18.4	13.4	21.2	16.0	22.9
384	$1536 \times 1536 \times 6$	37.3	45.0	16.8	36.5	24.2	38.0	29.8	38.9
864	$2304 \times 2304 \times 6$	58.8	68.9	22.9	56.1	35.7	56.2	45.3	62.6
1536	$3072 \times 3072 \times 6$	89.1	95.4	27.2	73.5	47.9	75.9	67.7	92.3
1944	$3456 \times 3456 \times 6$	109.4	118.5	29.2	85.3	54.3	87.6	79.8	110.7
2904	$4224 \times 4224 \times 6$	146.7	163.5	32.8	106.1	68.6	115.0	115.1	135.0
Case-2									
24	$384 \times 384 \times 6$	9.9	13.4	7.5	16.2	8.4	14.6	8.6	14.4
96	$768 \times 768 \times 6$	16.0	16.7	10.2	19.1	12.5	17.8	14.0	18.5
384	$1536 \times 1536 \times 6$	25.6	26.5	14.5	26.9	18.6	24.9	22.1	26.8
864	$2304 \times 2304 \times 6$	37.6	41.5	17.8	40.1	25.7	37.6	31.3	40.5
1536	$3072 \times 3072 \times 6$	53.5	65.9	20.9	53.3	33.5	50.6	44.0	60.6
1944	$3456 \times 3456 \times 6$	64.7	89.5	22.6	58.3	38.0	53.1	50.8	62.8
2904	$4224 \times 4224 \times 6$	89.3	111.4	25.7	86.8	46.9	80.8	64.9	90.1

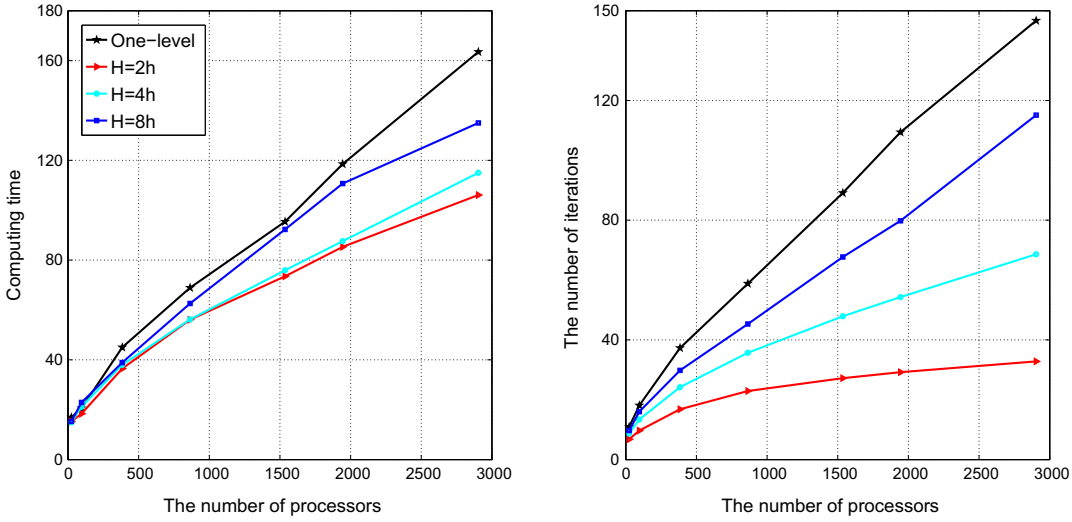


Fig. 8. Weak scalability results for **Case-1** with a fixed 192×192 mesh per processor. The time step size is fixed to $\triangle t = 0.2$ days, the simulation is stopped at day four.

the number of unknowns per processor as a constant. The test starts with a $384 \times 384 \times 6$ mesh for $N_p = 24$ and ends up with a $4224 \times 4224 \times 6$ mesh for $N_p = 2904$, as shown in Table 4. For the fully implicit methods, the condition number of the preconditioned matrix becomes larger when we refine the mesh and increase N_p, leading to the growth of the number of GMRES iterations, especially for the one-level method. If we focus on **Case-1**, observing from Fig. 8, we remark that the number of linear iterations for the case of $H = 2h$ is much smaller than for other cases, and the total solution time behaves similarly. Meanwhile, for **Case-2**, we find that the two-level method with $H = 4h$ shows the best performance in terms of the computing time.

5. Conclusion

In this paper, we developed a parallel two-level domain decomposition method with mixed order discretizations for the tracer transport problem on the cubed-sphere mesh, based on a fully implicit finite volume discretization. Since the scalability of the proposed method depends heavily on the design of the preconditioner, we used the idea of mixed order discretizations to build a class of two-level restricted Schwarz methods. In the approach, we built the Schwarz preconditioner in a multiplicative manner, in which the preconditioner on the coarse mesh is based on the first-order spatial discretization while the second-order scheme is used to construct the preconditioner on the fine mesh. Finally, we tested the algorithm for two benchmark problems to show the robustness and accuracy on high resolution meshes with more than one hundred million unknowns. The numerical experiments indicate that the method performs well in terms of both strong and weak scalabilities with up to 3072 processors.

Acknowledgements

This work was supported in part by NSF Grant CCF-1216314 and DOE Grant DE-SC0001774. H. Yang was also supported in part by NSFC Grants 91330111, 11201137 and 11272352. C. Yang was also supported in part by NSFC Grants 61170075 and 91130023 and 973 Grant 2011CB309701.

References

[1] Balay S, Brown J, Buschelman K, Eijkhout V, Gropp WD, Kaushik D, et al. PETSc users manual. Argonne National Laboratory; 2013.
[2] Cai X-C, Sarkis M. A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J Sci Comput 1999;21:792–7.
[3] Harris LM, Lauritzen PH, Mittal R. A flux-form version of the conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid. J Comput Phys 2011;230:1215–37.
[4] Hsu H-W, Hwang F-N, Wei Z-H, Lai S-H, Lin C-A. A parallel multilevel preconditioned iterative pressure Poisson solver for the large-eddy simulation of turbulent flow inside a duct. Comput Fluids 2011;45:138–46.
[5] Lauritzen PH, Jablonowski C, Taylor M, Nair RD. Numerical techniques for global atmospheric models. Berlin: Springer; 2011.
[6] Lauritzen PH, Nair RD, Ullrich PA. A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid. J Comput Phys 2010;229:1401–24.
[7] Nair RD, Jablonowski C. Moving vortices on the sphere: a test-case for horizontal advection problems. Mon Weather Rev 2008;136:699–711.
[8] Nair RD, Lauritzen PH. A class of deformational flow test cases for linear transport problems on the sphere. J Comput Phys 2010;229:8868–87.
[9] Rancic MR, Purser J, Mesinger F. A global-shallow water model using an expanded spherical cube: gnomonic versus conformal coordinates. Quart J Roy Meteor Soc 1996;122:959–82.
[10] Ronchi C, Iacono R, Paolucci P. The cubed sphere: a new method for the solution of partial differential equations in spherical geometry. J Comput Phys 1996;124:93–114.
[11] Rossmanith JA. A wave propagation method for hyperbolic systems on the sphere. J Comput Phys 2006;213:629–58.
[12] Saad Y. Iterative methods for sparse linear systems. SIAM; 2003.
[13] Sadourny R, Arakawa A, Mintz Y. Integration of the nondivergent barotropic vorticity equation with an icosahedralhexagonal grid for the sphere. Mon Weather Rev 1968;96:351–6.
[14] Sadourny R. Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. Mon Weather Rev 1972;100:211–24.
[15] Smith B, Bjørstad P, Gropp W. Domain decomposition: parallel multilevel methods for elliptic partial differential equations. Cambridge University Press; 1996.
[16] Toselli A, Widlund O. Domain decomposition methods: algorithms and theory. Springer; 2005.
[17] Williamson DL, Drake JB, Hack JJ, Jakob R, Swarztrauber PN. A standard test set for numerical approximations to the shallow water equations in spherical geometry. J Comput Phys 1992;102:2110–224.
[18] Wu Y, Cai X-C. A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition. Comput Fluids 2011;45:92–102.
[19] Yang C, Cai X-C. Parallel multilevel methods for implicit solution of shallow water equations with nonsmooth topography on cubed-sphere. J Comput Phys 2011;230:2523–39.
[20] Yang C, Cai X-C. A scalable fully implicit compressible Euler solver for mesoscale nonhydrostatic simulation of atmospheric flows. SIAM J Sci Comput 2014. To appear.
[21] Yang C, Cao J, Cai X-C. A fully implicit domain decomposition algorithm for shallow water equations on the cubed-sphere. SIAM J Sci Comput 2010;32:418–38.
[22] Yang H, Prudencio E, Cai X-C. Fully implicit Lagrange–Newton–Krylov–Schwarz algorithms for boundary control of un-steady incompressible flows. Int J Numer Meth Eng 2012;91:644–65.
[23] Yang H, Yang C, Cai X-C. Parallel domain decomposition methods with mixed order discretization for fully implicit solution of tracer transport problems on the cubed-sphere. J Sci Comput 2014. To appear.