Multigrid Solver for a Bulk Lightning Model Coupled with a Meteorological Model with Terrain-Following Coordinates

Soma Asai*[1,2][0009-0007-3230-5908], Yousuke Sato[1,3][0000-0002-6857-3783], Takeshi Iwashita[4][0000-0003-1938-1723], Yuta Kawai[3][0000-0002-9964-955X], and Seiya Nishizawa[3][0000-0001-9457-7457]

[1]Graduate School of Engineering, The University of Osaka, Suita, Osaka, Japan
[2] Graduate School of Science, Hokkaido University, Sapporo, Hokkaido, Japan
[3]RIKEN Center for Computational Science, Kobe, Hyogo, Japan
[4]Academic Center for Computing and Media Studies, Kyoto University, Kyoto, Japan
*Corresponding author
soma.asai@civil.eng.osaka-u.ac.jp,
yousuke.sato@civil.eng.osaka-u.ac.jp,iwashita@i.kyoto-u.ac.jp,
yuta.kawai@riken.jp, s-nishizawa@riken.jp

Abstract. In this study, we investigated the computational performance of a multigrid (MG) solver for the Poisson equation in a bulk lightning model (BLM) that explicitly calculates lightning frequency and is coupled to a meteorological model. The meteorological model employs highly anisotropic grids due to the strong disparity between typical horizontal and vertical scales of atmospheric phenomena, and it also exhibits terrain-induced horizontal grid non-uniformity. To improve the performance of the MG solver on such anisotropic grids, we implemented a vertical line Gauss-Seidel (*z*-line GS) smoother. Furthermore, we investigated the impact of terrain-induced grid non-uniformity on the convergence rate of the MG solver at horizontal grid spacings of $O(100$ m) by comparing it with the Bi-Conjugate Gradient Stabilized method preconditioned with symmetric Gauss-Seidel (SGS-BiCGSTAB) and incomplete lower-upper factorization (ILU-BiCGSTAB), respectively. For the performance evaluations, we computed the electric potential using a snapshot of the charge distribution from a realistic simulation with problem sizes (nz, nx, ny) = (32, 128, 128), (32, 256, 256), (32, 512, 512), and (32, 1024, 1024), corresponding to horizontal grid spacings of 2000, 1000, 500, and 250 m, respectively, using 16, 64, and 256 processors. For the (nz, nx, ny) = (32, 1024, 1024) case with 256-processor, the MG solver with the z-line GS smoother reduced the elapsed time by 93% (94%) (no terrain) and 89% (89%) (with terrain) relative to SGS-BiCGSTAB (ILU-BiCGSTAB). Without terrain, the iteration count for the MG solver was nearly independent of the problem size, whereas the iteration count for both preconditioned BiCGSTAB increased with the problem size. With terrain, the convergence rate of the MG solver deteriorated at high resolution due to terrain-induced grid non-uniformity. However, it remained superior to the preconditioned BiCGSTAB solvers, whose iteration count increased significantly with the problem size. These results suggest that the MG solver offers an advantage even for

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

simulations with the grid non-uniformity induced by small-scale topography on terrain-following coordinates at a high resolution of O(100 m).

Keywords: Terrain-following coordinates, multigrid solver, Poisson equation, bulk lightning model

1 Introduction

Meteorological models (e.g., the Consortium for Small-scale Modeling (COSMO) model [3], the Japan Meteorological Agency Nonhydrostatic Mesoscale (JMA-NHM) model [25], the Scalable Computing for Advanced Library and Environment (SCALE) model [21,27], and the Weather Research and Forecasting (WRF) model [31]) are computational codes designed to predict the temporal evolution of meteorological fields such as wind speed, temperature, humidity, and precipitation. These models are widely used by research communities, operational numerical weather prediction (NWP) centers, and private companies. Their applications cover a wide range of needs in modern society, including operational weather forecasting, the study of meteorological phenomena [13,26], the generation of landslide hazard warning maps [12,23], and demand forecasting for weather-sensitive products [2]. Consequently, meteorological models have become essential tools in modern society.

Lightning is a critical meteorological hazard in today's highly electrified society, and its accurate prediction is essential for mitigating lightning-related damage. In current NWP models, lightning is not represented explicitly; instead, the lightning threat or lightning frequency is often diagnosed empirically from model output variables [17]. Bulk lightning models (BLMs) [10,28], which predict lightning frequency based on the predicted charge density of hydrometeors, are powerful tools for accurate lightning prediction [32]. However, their computational costs are too high for operational NWP. Therefore, more computationally efficient BLMs are required for operational lightning prediction.

The primary computational bottleneck in BLMs is solving the Poisson equation:

$$\nabla^2\phi(x,y,z) = -\frac{\rho_e(x,y,z)}{\epsilon} \quad (1),$$

which calculates the electric potential (ϕ) from the charge density (ρ_e) and permittivity (ϵ). Discretizing Eq. 1 using a grid-point method yields a sparse linear system:

$$A\boldsymbol{u} = \boldsymbol{b} \quad (2),$$

where the total number of unknowns equals the total number of grid points in meteorological models. Solving Eq. 2 is computationally expensive; therefore, an efficient linear solver is essential for reducing the computational cost of BLMs. Furthermore, for future operational use, the following three characteristics of meteorological models must be taken into consideration. (i) Highly anisotropic grids are commonly used for real-case simulations due to the strong scale discrepancy between the horizontal and vertical directions. (ii) The problem size is expected to increase in the future. To improve forecast accuracy, the horizontal grid spacing used in operational forecasting has been reduced to $O(10^3$ m). Further grid refinement is anticipated as computational power continues to increase [22]. (iii) Terrain-induced horizontal grid non-uniformity

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

becomes more important as grid spacing decreases because small-scale topographic structures can be resolved. Based on these characteristics, BLMs require linear solvers for Eq. 1 that remain efficient for large problem sizes with small-scale terrain on anisotropic grids. For example, the Conjugate Residual method has been used in Meso-NH [15-16], and a multigrid (MG) solver [9] has been implemented in WRF and has been shown to maintain high computational efficiency even for large problem sizes [10]. Our research group also applied both an MG solver [1], the Bi-Conjugate Gradient Stabilized method [35] preconditioned with symmetric Gauss-Seidel (SGS-BiCGSTAB) and incomplete lower-upper factorization (ILU-BiCGSTAB) [24] to Eq. 1 in our BLM [28] in SCALE. It is known that the Krylov subspace methods including Conjugate Residual method and BiCGSTAB with an appropriate preconditioner are effective on anisotropic grids [19]. Thus, such solvers are expected to be effective for meteorological models. However, in some cases, the iteration count of these Krylov solvers tends to increase with the problem size [6]. In contrast, MG solvers can offer computational advantages for large-scale problems. Moreover, MG solvers can maintain the computational efficiency even on anisotropic grids by adopting an appropriate smoother [8]. However, it is unclear to what extent terrain-induced grid non-uniformity deteriorates the convergence rates of both solvers in simulations with small-scale terrain.

Against this background, the aim of this study is to investigate how terrain-induced horizontal non-uniformity affects the convergence of the MG solver and SGS-BiCGSTAB for the Poisson problem in high-resolution BLM experiments with small-scale topography. We then evaluate which solver is more advantageous for BLM simulations at fine grid resolutions. The MG solver was implemented by Asai and Sato [1] and was shown to achieve fast performance for large problems. However, the original implementation has two limitations: the smoother cannot maintain high computational efficiency on anisotropic grids, and the interpolation method has low computational cost but insufficient accuracy. To achieve our purpose, we address these two issues and compare the performance of the improved MG solver with that of SGS-BiCGSTAB and ILU-BiCGSTAB, respectively.

2 Method

2.1 Discretization of the Poisson Equation in a BLM

In this study, we focus on the Poisson equation in a BLM coupled to a meteorological model, i.e., SCALE. The Poisson equation must be solved at each time step to obtain the electric potential ϕ from the three-dimensional distribution of the predicted charge density ρ_e. On the Arakawa-C grid adopted by SCALE, scalar quantities (e.g., temperature, density) are defined at cell centers, whereas vector quantities (e.g., momentum) are defined at cell faces. By applying a second-order finite-difference method, the discretized form of Eq. 1 is derived as:

$$\frac{1}{\Delta x_j^f}\left(\frac{\phi_{j+1,k,l}-\phi_{j,k,l}}{\Delta x_j^c}-\frac{\phi_{j,k,l}-\phi_{j-1,k,l}}{\Delta x_{j-1}^c}\right)$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

$$+\frac{1}{\Delta y_k^f}\left(\frac{\phi_{j,k+1,l}-\phi_{j,k,l}}{\Delta y_k^c}-\frac{\phi_{j,k,l}-\phi_{j,k-1,l}}{\Delta y_{k-1}^c}\right)$$
$$+\frac{1}{\Delta z_l^f}\left(\frac{\phi_{j,k,l+1}-\phi_{j,k,l}}{\Delta z_l^c}-\frac{\phi_{j,k,l}-\phi_{j,k,l-1}}{\Delta z_{l-1}^c}\right)=-\frac{(\rho_e)_{j,k,l}}{\epsilon} \quad (3),$$

where Δx, Δy, and Δz are the grid spacings in the x, y, and z directions, respectively; the superscript f denotes face-to-face spacing, whereas those with the superscript c denotes center-to-center spacing. Subscripts j, k, and l denote the grid indices in the x, y, and z directions, respectively. Rearranging and assembling Eq. 3 over all grid points yields the linear equation system:

$$A'\boldsymbol{u}' = \boldsymbol{b} \quad (4).$$

Here, the coefficient matrix A' is sparse, with only seven nonzero elements in each row, determined by the grid spacing. The solution vector $\boldsymbol{u}'$ and the right-hand side vector $\boldsymbol{b}$ correspond to the electric potential ϕ and the charge density ρ_e divided by the permittivity ϵ, respectively.

For real-case simulations such as NWP, it is necessary to account for Earth's curvature and terrain. Cartesian coordinates (x, y, z) are transformed into computational coordinates (ξ, η, ζ), which consist of locally orthogonal curvilinear coordinates in the horizontal directions and a terrain-following coordinate in the vertical direction. The coordinate transformation is defined by:

$$\xi = x \quad (5a),$$
$$\eta = y \quad (5b),$$
$$\zeta = \frac{z_{\mathrm{toa}}(z - z_{\mathrm{sfc}})}{z_{\mathrm{toa}} - z_{\mathrm{sfc}}} \quad (5c).$$

Here, z_{toa} is the altitude of the model top and z_{sfc} is the terrain height. In (ξ, η, ζ), $\nabla\phi$ can be written as:

$$\nabla\phi = m\left(\frac{\partial\phi}{\partial\xi}+J_{x\zeta}\frac{\partial\phi}{\partial\zeta}\right)\boldsymbol{e_x}+n\left(\frac{\partial\phi}{\partial\eta}+J_{y\zeta}\frac{\partial\phi}{\partial\zeta}\right)\boldsymbol{e_y}+J_{z\zeta}\left(\frac{\partial\phi}{\partial\zeta}\right)\boldsymbol{e_z} \quad (6).$$

Here, $\boldsymbol{e_x}$, $\boldsymbol{e_y}$, and $\boldsymbol{e_z}$ are the unit vectors for the x, y, and z directions, respectively. $J_{x\zeta}$, $J_{y\zeta}$, and $J_{z\zeta}$ are the Jacobian elements representing terrain effects. m and n are the map factors arising from the coordinate transformation due to Earth's curvature. Substituting Eq. 6 into Eq. 1, the left-hand side becomes:

$$\nabla^2\phi = m\left(\frac{\partial(\nabla\phi)_x}{\partial\xi}+J_{x\zeta}\frac{\partial(\nabla\phi)_x}{\partial\zeta}\right)+n\left(\frac{\partial(\nabla\phi)_y}{\partial\eta}+J_{y\zeta}\frac{\partial(\nabla\phi)_y}{\partial\zeta}\right)+J_{z\zeta}\left(\frac{\partial(\nabla\phi)_z}{\partial\zeta}\right)$$
$$= m\frac{\partial}{\partial\xi}\left(m\frac{\partial\phi}{\partial\xi}\right)+m\frac{\partial}{\partial\xi}\left(mJ_{x\zeta}\frac{\partial\phi}{\partial\zeta}\right)+mJ_{x\zeta}\frac{\partial}{\partial\zeta}\left(m\frac{\partial\phi}{\partial\xi}\right)+mJ_{x\zeta}\frac{\partial}{\partial\zeta}\left(mJ_{x\zeta}\frac{\partial\phi}{\partial\zeta}\right)$$
$$+n\frac{\partial}{\partial\eta}\left(n\frac{\partial\phi}{\partial\eta}\right)+n\frac{\partial}{\partial\eta}\left(nJ_{y\zeta}\frac{\partial\phi}{\partial\zeta}\right)+nJ_{y\zeta}\frac{\partial}{\partial\zeta}\left(n\frac{\partial\phi}{\partial\eta}\right)+nJ_{y\zeta}\frac{\partial}{\partial\zeta}\left(nJ_{y\zeta}\frac{\partial\phi}{\partial\zeta}\right)$$
$$+J_{z\zeta}\frac{\partial}{\partial\zeta}\left(J_{z\zeta}\frac{\partial\phi}{\partial\zeta}\right) \quad (7).$$

Discretizing Eq. 7 in a manner similar to Eq. 3 yields Eq. 2. In this case, A becomes a coefficient matrix with 15 nonzero elements per row, consisting of the nonzero components from Eq. 3 and $\phi_{j\pm1,k,l-1}$, $\phi_{j,k\pm1,l-1}$, $\phi_{j\pm1,k,l+1}$, $\phi_{j,k\pm1,l+1}$. The solution vector $\boldsymbol{u}$ corresponds to the electric potential defined on terrain-following coordinates.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

2.2 Multigrid Solver

A multigrid (MG) solver [34] is an iterative method that efficiently reduces errors at each wavenumber by solving the problem on grids with different resolutions (called grid levels). Due to this property, the number of iterations remains nearly constant regardless of problem size, enabling high computational efficiency even for large-scale problems. We adopted an MG solver with a V-cycle in which an approximate solution is obtained by repeating restriction and interpolation. Algorithms 1 and 2 outline the MG solver and the V-cycle procedures, respectively.

Algorithm 1 Multigrid solver

`function` `Multigrid` $(A, \boldsymbol{b}, \boldsymbol{w}_{\text{in}})$:
  (1) Set an arbitrary vector as initial approximate solution vector: $\boldsymbol{w}^{(0)} = \boldsymbol{w}_{\text{in}}$
  `for` $i_t = 0,\ 1,\ 2, \cdots$ `until` $\|\boldsymbol{r}^{(i_t)}\|_2 \leq \varepsilon \|\boldsymbol{b}\|_2$ `do`
    (2) Compute the approximate solution vector: $\boldsymbol{w}^{(i_t+1)} =$`V-cycle` $(0, \boldsymbol{w}^{(i_t)})$
    (3) Compute the residual vector: $\boldsymbol{r}^{(i_t+1)} = \boldsymbol{b} - A\boldsymbol{w}^{(i_t+1)}$
  `end for`
  (4) Set $\boldsymbol{w}^{(i_t+1)}$ as approximate vector: $\boldsymbol{w}_{\text{out}} = \boldsymbol{w}^{(i_t+1)}$
  `return` $\boldsymbol{w}_{\text{out}}$

Algorithm 2 V-cycle

`function` `V-cycle` $(lv, \boldsymbol{w}^{(lv)})$:
  (1) Presmoothing: $\bar{\boldsymbol{w}}^{(lv)} =$`S`$_1$ $(A^{(lv)}, \boldsymbol{w}^{(lv)}, \boldsymbol{b}^{(lv)})$
  (2) Compute the residual vector: $\boldsymbol{r}^{(lv)} = \boldsymbol{b}^{(lv)} - A^{(lv)}\bar{\boldsymbol{w}}^{(lv)}$
  (3) Restrict the residual vector: $\boldsymbol{b}^{(lv+1)} = R^{(lv)}\boldsymbol{r}^{(lv)}$
  (4) Compute an approximate vector: $\bar{\boldsymbol{w}}^{(lv+1)}$ by
    `if` $(lv + 1 = lv_{\max})$:
      fast solver: $\bar{\boldsymbol{w}}^{(lv_{\max})} =$`S`$_2$ $(A^{(lv_{\max})}, \boldsymbol{w}^{(lv_{\max})}, \boldsymbol{b}^{(lv_{\max})})$
    `else`:
      V-cycle algorithm: $\bar{\boldsymbol{w}}^{(lv+1)} =$`V-cycle` $(lv + 1, \boldsymbol{w}^{(lv+1)})$
  (5) Interpolate the correction: $d\boldsymbol{w}^{(lv)} = P^{(lv)}\bar{\boldsymbol{w}}^{(lv+1)}$
  (6) Compute the corrected approximate vector: $\hat{\boldsymbol{w}}^{(lv)} = \bar{\boldsymbol{w}}^{(lv)} + d\boldsymbol{w}^{(lv)}$
  (7) Postsmoothing: $\boldsymbol{w}'^{(lv)} =$`S`$_3$ $(A^{(lv)}, \hat{\boldsymbol{w}}^{(lv)}, \boldsymbol{b}^{(lv)})$
  `return` $\boldsymbol{w}'^{(lv)}$

Here, $R^{(lv)}$ and $P^{(lv)}$ are restriction and interpolation (prolongation) matrices, respectively, while `S`$_1$, `S`$_2$, and `S`$_3$ are the presmoother, coarsest-grid solver, and postsmoother functions.

Due to the non-uniform vertical grids used in meteorological models, it is difficult to perform restriction in the vertical direction. Therefore, this study performs restriction only in the horizontal directions. The coefficient matrix $A^{(lv)}$ at each level must be precomputed before executing the MG algorithm. To account for Earth's curvature and terrain, the coefficient matrix includes the map factors and the Jacobian elements. These factors must be calculated consistently with the grid system at each level. The former

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

can be determined based on the corresponding grid spacing. For the latter, terrain height z_{sfc} must be generated for each grid level.

The convergence rate of the Gauss-Seidel smoother [24] applied in Asai and Sato [1] deteriorates on grids with vertically non-uniform spacing and large aspect ratios. Thus, we adopted a forward and backward z-line Gauss-Seidel (GS) smoother for $\mathtt{S}_1$ and $\mathtt{S}_3$ because it maintains a high convergence rate even on such anisotropic grids [8]. For $\mathtt{S}_2$, we applied an unpreconditioned BiCGSTAB. Additionally, we employed bilinear interpolation [34] to improve the accuracy of the correction.

2.3 Parallelization Methods

SCALE uses the Message Passing Interface (MPI) for process parallelism. To enable parallel execution of the MG solver, the restriction, interpolation, and smoothing operations need to be parallelized using MPI.

Before describing the details of the parallel implementation, we outline the domain decomposition strategy at each level. The computational domains are constructed by reducing the number of grid points and MPI processes in two steps: (i) reducing the grid size on each process until the number of grid points in each horizontal direction equals twice the number of ghost cells (i.e., halo cells); and (ii) further reducing the grid size across the entire computational domain by decreasing the number of active processes until only a single processor remains. Since the number of active processes varies across levels, we create a distinct MPI communicator for each level. Fig. 1 illustrates the parallelization method for a case with (nx, ny) = (32, 32), 16 MPI processes, and a ghost-cell count of 2. In this instance, the minimum grid size per process is 4 × 4, and the maximum level is 3.

First, we explain the halo-exchange procedure required for restriction, interpolation, and smoothing. In the case of Process 5 in Fig. 1b, for example, the components of the approximate solution in the halo regions across the X and Y boundaries (represented as $w^{(lv=1)}_{j=3:4(9:10),k=5:8}$ and $w^{(lv=1)}_{j=5:8,k=3:4(9:10)}$) are updated via MPI communication with Processes 4 (or 6) and 1 (or 9), respectively. Additionally, when performing bilinear interpolation, the corner halo values (represented as $w^{(lv=1)}_{j=3:4,k=3:4}$, $w^{(lv=1)}_{j=3:4,k=9:10}$, $w^{(lv=1)}_{j=9:10,k=3:4}$ and $w^{(lv=1)}_{j=9:10,k=9:10}$) must also be updated via MPI communication with Processes 0, 8, 2, and 10, respectively.

Next, we describe how restriction and interpolation are implemented using MPI communication. For example, when performing the restriction from level 2 to level 3 in Figs. 1c-d, we send the residual components $r^{(lv=2)}_{j=5:8,k=1:4}$, $r^{(lv=2)}_{j=1:4,k=5:8}$, and $r^{(lv=2)}_{j=5:8,k=5:8}$ from Processes 1, 2, and 3 to Process 0. These components are then combined to form $r^{(lv=2)}_{j=1:8,k=1:8}$ on Process 0. Then, Process 0 executes the restriction at level 3, corresponding to Step 3 of Algorithm 2. Conversely, the interpolation is conducted in the reverse order of the restriction. When interpolating from level 3 to level 2 in Figs. 1c-d, we first perform the halo exchange, including corner values, and the bilinear interpolation at level 3 corresponding to Step 5 of Algorithm 2, thereby generating $dw^{(lv=2)}_{j=1:8,k=1:8}$. Then,

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

we divide $dw_{j=1:8,k=1:8}^{(lv=2)}$ into four segments. Process 0 distributes $dw_{j=1:4,k=5:8}^{(lv=2)}$, $dw_{j=5:8,k=1:4}^{(lv=2)}$, and $dw_{j=5:8,k=5:8}^{(lv=2)}$ to Processes 1, 2, and 3, respectively, while retaining $dw_{j=1:4,k=1:4}^{(lv=2)}$.

Finally, we explain the smoothing implementation using MPI communication. The forward (backward) z-line GS method requires the components of the approximate solution at indices $(x_{j-1(j+1)}, y_k, z_l)$ and $(x_j, y_{k-1(k+1)}, z_l)$ to update the component at index (x_j, y_k, z_l). Thus, before smoothing, we update the halo cells excluding corner values, which are not required by the smoother.

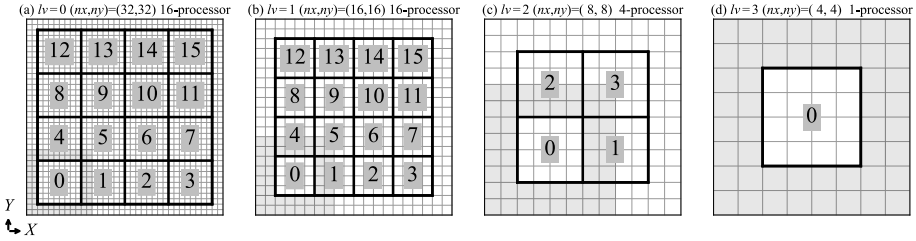

Fig. 1. Process parallelization method at grid levels (a) 0, (b) 1, (c) 2, and (d) 3 for the case of $(nx, ny) = (32,32)$ and a ghost-cell count of 2. Black and gray lines show the boundaries of process subdomains model grids, respectively. Gray-shaded area indicates the halo area of Process 0. The numbers at the center of black cells show the process ranks.

2.4 Preparing the Charge Distribution for Performance Evaluation

To evaluate the performance of each solver, we prepared a problem with a charge distribution corresponding to the right-hand side vector obtained from a long-run simulation.

Model Configuration. The simulation was performed using SCALE version 5.5.4 [21,27]. The temporal discretization in the dynamical core employed an explicit scheme in the horizontal directions and an implicit scheme in the vertical direction. Cloud microphysics was represented by the one-moment bulk microphysics scheme of Tomita [33]. Subgrid-scale turbulence was parameterized using the Smagorinsky–Lilly scheme [7,29] in the horizontal directions and the MYNN scheme [18,20] in the vertical direction. Radiative transfer was calculated using the MSTRN-X scheme [30]. For the lightning scheme, we incorporated the explicit BLM of Sato et al. [28]. The single-layer urban canopy model [14] was used as the urban scheme. Surface temperature was determined by balancing the surface energy budget, and surface fluxes were computed using the Beljaars-type bulk scheme [4]. Sea surface temperature (SST) was held constant at its initial value during the simulation.

Experimental Setup. The model employed a horizontal grid size of 1024×1024 with a 250-m resolution, and 32 vertical layers from the surface to the model top at a height of 19,980 m. The vertical grid spacing was stretched from 81 m near the surface to 1,594 m at the top. The computational domain is shown in Fig. 2. We carried out a simulation for a lightning event on July 5, 2017. This simulation was conducted from 00:00 to 03:00 UTC without the lightning discharge process. Topography was included

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

in this run. Initial and boundary conditions for this simulation were derived from the JMA Mesoscale Analysis (MANL) [11], which provides a 5-km horizontal resolution, 50 vertical layers, and a 3-h temporal resolution. To maintain physical consistency between the parent and child models, nudging was applied within 40 km of the lateral boundaries. The Neumann condition was imposed at the lateral boundaries and the Dirichlet condition was applied to the upper and bottom boundaries.

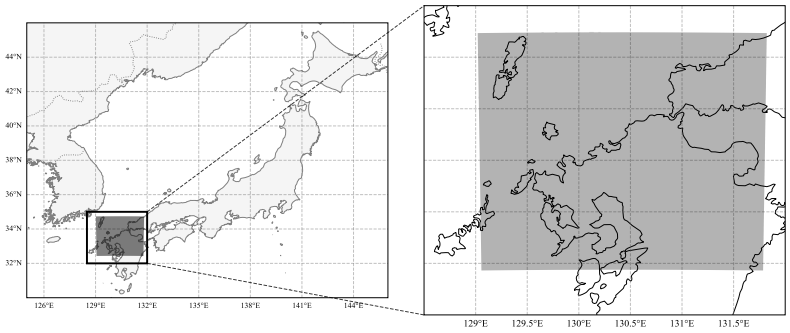

Fig. 2. Computational domain represented as the shaded area.

2.5 Performance Evaluation Method

To evaluate the performance of the MG solver, SGS-BiCGSTAB, and ILU-BiCGSTAB, we performed experiments with and without terrain and measured the elapsed time and the number of iterations required for each solver to converge. The total number of grid points and the horizontal grid spacings are listed in Table 1. The performance evaluations were conducted using 16, 64, and 256 processors. The right-hand side vectors for each problem size were created by restriction from a snapshot of the charge distribution obtained at 03:00 UTC on July 5, 2017, from the simulation described in Sect. 2.4. Convergence was judged using the criterion:

$$\left\|\boldsymbol{r}^{(m)}\right\|_2 / \|\boldsymbol{b}\|_2 \leq 10^{-10} \tag{8},$$

where $\boldsymbol{r}^{(m)} = \boldsymbol{b} - A\boldsymbol{w}^{(m)}$ is the residual vector, and $\boldsymbol{w}^{(m)}$ is the approximate solution vector after m iterations. The number of presmoothing and postsmoothing, as well as the number of iterations of the coarsest-grid solver, were all set to one, which yielded the shortest elapsed time based on preliminary tuning experiments. The iteration count for the MG solver corresponds to the number of V-cycle iterations required to satisfy the convergence criterion. The initial approximate solution vector for all solvers was the zero vector. Each experiment was repeated five times, and the elapsed time was defined as the average over the five runs.

Table 1. Total number of grid points and horizontal grid spacing in the performance evaluations

(*nz*, *nx*, *ny*)	(32, 128, 128)	(32, 256, 256)	(32, 512, 512)	(32, 1024, 1024)
Total number of grid points	524288	2097152	8388608	33554432
Horizontal grid spacing (m)	2000	1000	500	250

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

2.6 Computational Environment

The simulation to create the right-hand side vector was performed on the SQUID supercomputer operated by the D3 Center at The University of Osaka, and the performance evaluations were conducted on the Miyabi-C supercomputer operated by the Information Technology Center at The University of Tokyo and the Center for Computational Sciences at University of Tsukuba. The hardware specifications of Miyabi-C are summarized in Table 2.

Table 2. Configuration of Miyabi-C supercomputer (Retrieved from https://www.cc.u-tokyo.ac.jp/supercomputer/miyabi/system.php, accessed 10 December 2025 UTC)

System	Model name	Fujitsu Server PRIMERGY CX2550 M7
	Number of nodes	190
	Theoretical performance (double precision)	1.3 PFLOPS
	Total memory capacity	24.3 TiB
	Network topology	InfiniBand NDR200 interconnect
Node	Processor count (cores)	2 (2×56=112)
	Theoretical performance (double precision)	6.8096 TFLOPS (1.9 GHz×56 cores×32 instructions×2 CPUs)
	Memory capacity	128 GiB
	Theoretical memory bandwidth	3.2 TB/s
	Interconnect	InfiniBand NDR×1 port (200 Gbps×1)
	OS	Rocky Linux 9
Processor	Processor name	Intel Xeon MAX 9480
	Architecture	Sapphire Rapids HBM
	Clock speed	1.9 GHz
	Core count	56
	Theoretical performance	3.405 TFLOPS

3 Results

3.1 Number of Iterations

We examined the dependence of the iteration count on the problem size, terrain, and the number of processors. Table 3 shows the dependence of the iteration count on the problem size for cases (a) without terrain and (b) with terrain.

We first describe the dependence of the iteration count on the problem size without terrain (Table 3a). For the MG solver, as expected, the iteration count remains nearly constant regardless of the problem size. However, a slight increase is observed as the problem size increases for the 64-processor case. This is because the aspect ratio—defined as the vertical grid spacing divided by the horizontal grid spacing—increases with higher resolutions, leading to a decrease in the convergence rate of the z-line GS smoother at fine grid levels. However, at the coarser grid levels, the aspect ratio remains small, and the smoother maintains a high convergence rate. Thus, the impact of the increased aspect ratio on the MG solver is limited. In contrast, the number of iterations for SGS-BiCGSTAB and ILU-BiCGSTAB increased at an average rate of 1.539 (= $(237/65)^{1/3}$) and 1.587 (= $(292/73)^{1/3}$) using 256 processors, respectively. This is because the Krylov subspace method creates an approximate solution vector by expanding

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

the Krylov subspace, and the number of basis vectors required to approximate the solution tends to increase with increasing the problem size when standard preconditioning methods are used.

Next, we focus on the effects of terrain (Table 3b). In the experiments with terrain, the number of iterations increased with resolution for the MG solver. In particular, we confirmed an 87% increase for the (32, 1024, 1024) case using 16 processors. This behavior can be attributed to terrain-induced degradation of the convergence rate of the smoother at each level. In other words, grid non-uniformity arising from resolved terrain at the finest horizontal spacing (250 m) significantly affects the convergence rate of the smoother. For the horizontal resolutions coarser than 500 m, corresponding to the problem size (32, 512, 512), the effect of grid non-uniformity is minor. By contrast, similar degradation was negligible for both SGS-BiCGSTAB and ILU-BiCGSTAB.

We next discuss the dependence of the iteration count on the number of processors. Increasing the number of parallel processes led to a slight increase in the iteration count for the MG solver. This is likely because, at MPI process boundaries, the forward (backward) z-line GS smoother cannot access the updated solution components defined at the preceding (following) indices, which reduces the convergence performance of the smoother. Such degradation was not confirmed for either SGS-BiCGSTAB or ILU-BiCGSTAB.

3.2 Elapsed Time

Next, we focus on the elapsed time to investigate the practical computational efficiency. Fig. 3 shows the dependence of the elapsed time on the problem size for cases (a-c) without terrain and (d-f) with terrain. In the experiments without terrain, the elapsed time of the MG solver increased almost proportionally to the problem size (Fig. 3a). In contrast, in the experiments with terrain (Fig. 3d), the increase in the number of iterations caused a slight deviation from ideal linear scaling, especially for the (32, 1024, 1024) case. However, significant departures from linear scaling were observed in both experiments for small problem sizes, particularly as the number of processors increased. This behavior was caused by the communication costs for the restriction and interpolation, which dominate the total elapsed time. For both SGS-BiCGSTAB (Figs. 3b, c) and ILU-BiCGSTAB (Figs. 3e, f), the elapsed time deviated substantially from linear scaling for all problem sizes. The performance degradation became notable for large-scale problems because more iterations were required. As a result, the elapsed times of both preconditioned BiCGSTAB solvers were much longer than that of the MG solver. For instance, in the (32, 1024, 1024) case using 256 processors, the MG solver reduced the elapsed time by 93% (94%) without terrain and by 89% (89%) with terrain relative to SGS-BiCGSTAB (ILU-BiCGSTAB). Overall, these results indicate that terrain effects were secondary to the impact of problem size, and the MG solver was more efficient than the preconditioned BiCGSTAB solvers for the cases considered in this study.

In addition, we conducted weak-scaling evaluation without terrain (Fig. 4a) and with terrain (Fig. 4b). In the evaluation, the total number of grid points per processor is 131,072 ($=32\times64\times64$). Without terrain, the MG solver exhibited near-ideal weak

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

scaling, indicating high parallel efficiency. However, with terrain, the MG solver did not achieve weak scaling. In contrast, neither preconditioned BiCGSTAB solver exhibited for weak scaling with or without terrain.

3.3 Comparison of Solutions

Finally, we confirmed whether the approximate solution calculated using the MG solver was equivalent to that using SGS-BiCGSTAB. The relative difference between these solutions is defined as:

$$RE(x,y,z) = \left| \frac{\phi_{\mathrm{MG}}(x,y,z) - \phi_{\mathrm{SGS-BiCGSTAB}}(x,y,z)}{\phi_{\mathrm{SGS-BiCGSTAB}}(x,y,z)} \right| \times 100\ (\%) \qquad (9).$$

where $\phi_{\mathrm{MG}}(x,y,z)$ and $\phi_{\mathrm{SGS-BiCGSTAB}}(x,y,z)$ are the electric potentials calculated by the MG solver and SGS-BiCGSTAB, respectively, at position (x,y,z). The maximum relative difference was $\max(RE)$ = 0.565%. Thus, we conclude that the MG solver achieves numerical accuracy comparable to that of SGS-BiCGSTAB.

Table 3. Number of iterations to satisfy the convergence criterion (Eq. 8), computed using 16, 64, and 256 processors with the MG solver, SGS-BiCGSTAB, and ILU-BiCGSTAB in the experiments (a) without terrain and (b) with terrain

(a)

Solver name (number of processors)	(32,128,128)	(32,256,256)	(32,512,512)	(32,1024,1024)
MG solver (16)	15	15	16	28
MG solver (64)	15	16	16	28
MG solver (256)	16	16	17	28
SGS-BiCGSTAB (16)	66	87	145	228
SGS-BiCGSTAB (64)	62	75	146	240
SGS-BiCGSTAB (256)	72	88	144	255
ILU-BiCGSTAB (16)	65	89	149	262
ILU-BiCGSTAB (64)	69	89	155	283
ILU-BiCGSTAB (256)	70	105	149	256

(b)

Solver name (number of processors)	(32,128,128)	(32,256,256)	(32,512,512)	(32,1024,1024)
MG solver (16)	15	15	15	15
MG solver (64)	15	16	16	16
MG solver (256)	16	16	16	16
SGS-BiCGSTAB (16)	62	78	132	223
SGS-BiCGSTAB (64)	61	82	126	231
SGS-BiCGSTAB (256)	65	82	123	237
ILU-BiCGSTAB (16)	70	94	134	264
ILU-BiCGSTAB (64)	74	87	147	247
ILU-BiCGSTAB (256)	73	97	143	292

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

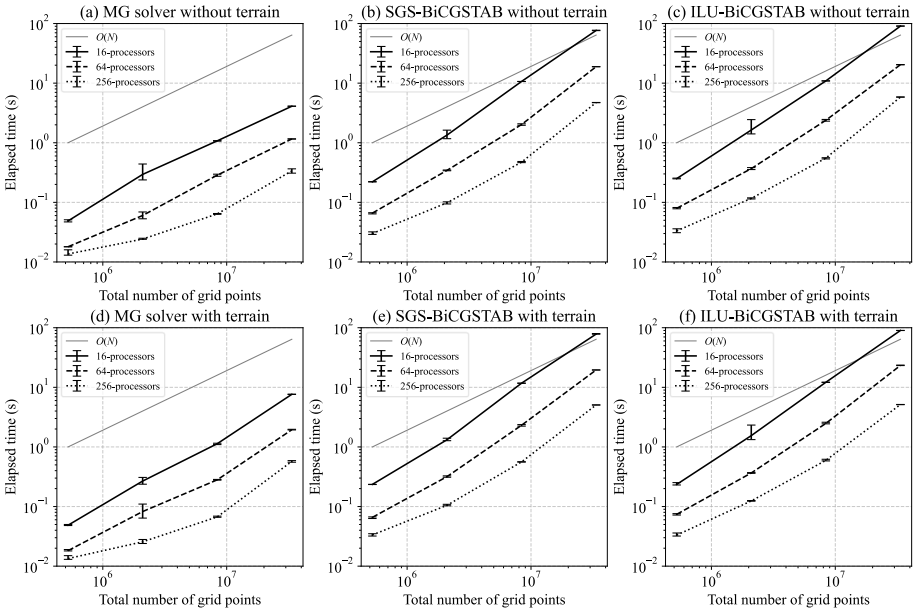

Fig. 3. Problem-size dependence of elapsed time for experiments (a-c) without terrain and (d-f) with terrain, using 16 (solid), 64 (dotted), and 256 (dashed) processors, for (a, d) the MG solver, (b, e) SGS-BiCGSTAB, and (c, f) ILU-BiCGSTAB. The gray solid line shows ideal linear scaling. Error bars show the range between the maximum and minimum elapsed times.

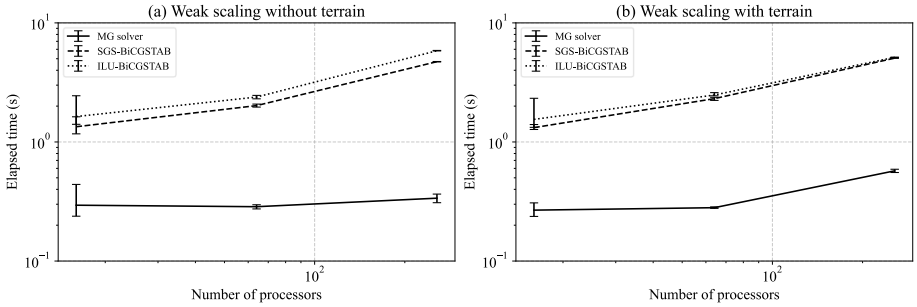

Fig. 4. Weak scaling for experiments (a) without terrain and (b) with terrain, using the MG solver (solid), SGS-BiCGSTAB (dotted) and ILU-BiCGSTAB (dashed). The total number of grid points per processor is 131,072 (=32×64×64).

4 Conclusion and Discussion

In this study, we applied an MG solver with a z-line GS smoother to the Poisson equation in a bulk lightning model (BLM) coupled with a meteorological model. The model typically uses terrain-following coordinates, causing terrain-induced horizontal

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

grid non-uniformity in the vertical grid spacing. We examined the computational performance of the MG solver and the impact of terrain-induced horizontal grid non-uniformity on its performance. For the performance evaluation, we compared the convergence rate of the MG solver with that of preconditioned Krylov subspace methods: SGS-BiCGSTAB and ILU-BiCGSTAB. The results showed that, at high resolutions, the horizontal non-uniformity caused a reduction of over 80% in the convergence rate of the MG solver, while the impact on that of SGS-BiCGSTAB and ILU-BiCGSTAB was small. As the problem size increased, the number of iterations for the MG solver remained almost unchanged, whereas that for both preconditioned BiCGSTAB solvers increased in both experiments regardless of terrain. Consequently, the MG solver limited the increase in iterations to an average rate of 1.231 ($= (28/15)^{1/3}$, using 16 processors), which is smaller than that of both preconditioned BiCGSTAB solvers. Thus, for large-scale problems, the MG solver was more computationally efficient than both preconditioned BiCGSTAB solvers. For example, the MG solver achieved reductions in elapsed time exceeding 85% for the largest test cases. Overall, our results suggest that, for horizontal resolutions on the order of 100 m in atmospheric models, the MG solver remains more computationally efficient than both preconditioned BiCGSTAB solvers even on horizontally non-uniform grids arising from terrain-following coordinates.

However, the reason for the sharp increase in the iteration count of the MG solver for the case (32, 1024, 1024) has not yet been clarified. We hypothesize that there is a terrain-related threshold that significantly reduces the convergence rate of the smoother. However, it is difficult to perform Local Fourier Analysis (LFA) [5] or eigenvalue analysis on coordinates with vertically non-uniform grid spacing and terrain-induced horizontal non-uniformity. Thus, addressing this challenge is left for future work. In addition, developing MG solvers that maintain high computational efficiency under strong horizontal non-uniformity remains an important topic.

Acknowledgments. This research was supported by Grants-in-Aid for Scientific Research (JP24H00257, JP24H00269, and JP25K01067) from the Japan Society for the Promotion of Science (JSPS) and by a grant from the SECOM Science and Technology Foundation. The JMA data used in this study were provided through ”Meteorological Research Consortium”, a framework for research cooperation of JMA and Meteorological Society of Japan. The supercomputer “SQUID” managed by The University of Osaka D3 Center and the supercomputer “Miyabi” managed by both Information Technology Center at The University of Tokyo and Center for Computational Sciences at University of Tsukuba were used in this research. This research was conducted as part of a joint research project between The University of Osaka and the Central Research Institute of Electric Power Industry. We express our gratitude to Dr. Tomita of the RIKEN Center for Computational Science for helpful advice regarding the improvement of the MG solver in SCALE.

References

1. Asai, S., Sato, Y.: Dependency of the horizontal resolution in the charge structure of a convective cloud in a bulk lightning model. SOLA. 21, 208-216 (2025). doi: 10.2151/sola.2025-026

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

2. Babongo, F., Appelqvist, P., Chavez-Demoulin, V., Hameri, A. P., Niemi, T.: Using weather data to improve demand forecasting for seasonal products. Int. J. Serv. Oper. Manag. 31(1), 53-76 (2018). doi: 10.1504/IJSOM.2018.094183
3. Baldauf, M., Seifert, A., Forstner, J., Majewski, D., Raschendorfer, M., Reinhardt, T.: Operational Convective-Scale Numerical Weather Prediction with the COSMO Model: Description and Sensitivities. Mon. Weather Rev. 139(12), 3887-3905 (2011). doi: 10.1175/MWR-D-10-05013.1
4. Beljaars, A.C.M., Holtslag, A.A.M.: Flux parameterization over land surfaces for atmospheric models. J. Appl. Meteorol. 30(3), 327–341 (1991). doi:10.1175/1520-0450(1991)030<0327:FPOLSF>2.0.CO;2
5. Brandt, A.: Multi-Level Adaptive Solutions to Boundary-Value Problems, Math. Comput. 31(138), 333-390 (1977).
6. Bridson, R., Greif, C.: A Multipreconditioned Conjugate Gradient Algorithm. SIAM J. Matrix Anal. Appl. 27(4), 1056-1068 (2006). doi: 10.1137/040620047
7. Brown, A.R., Derbyshire, S.H., Mason, P.J.: Large-eddy simulation of stable atmospheric boundary layers with a revised stochastic subgrid model. Q. J. R. Meteorol. Soc. 120(520), 1485–1512 (1994). doi:10.1002/qj.49712052004
8. Buckeridge, S., Scheichl, R.: Parallel geometric multigrid for global weather prediction. Numer. Linear Algebra Appl. 17(2–3), 325–342 (2010). doi:10.1002/nla.699
9. Dendy, J. E. Jr.: Two multigrid methods for three-dimensional problems with discontinuous and anisotropic coefficients. SIAM J. Sci. Stat. Comput. 8(5), 673–685 (1987).
10. Fierro, A.O., Mansell, E.R., MacGorman, D.R., Ziegler, C.L.: The implementation of an explicit charging and discharge lightning scheme within the WRF-ARW model: Benchmark simulations of a continental squall line, a tropical cyclone, and a winter storm. Mon. Weather Rev. 141(7), 2390–2415 (2013). doi: 10.1175/MWR-D-12-00278.1
11. JMA: Outline of the operational numerical weather prediction at the Japan Meteorological Agency (2022), https://www.jma.go.jp/jma/jma-eng/jma-center/nwp/outline2022-nwp/index.htm, accessed 29 November 2025 UTC
12. Khan, H., Kirschbaum, B. D., Stanley, A. T., Amatya, M. P., Emberson, A. R.: Global Landslide Forecasting System for Hazard Assessment and Situational Awareness. Front. Earth Sci. 10, 878996 (2022). doi: 10.3389/feart.2022.878996
13. Kondo, M., Sato, Y.: Transition of dominant cloud microphysical processes for increasing lightning preceding downbursts in multi-cell convective clouds. Atmos. Res. 324, 108203 (2025). doi: 10.1016/j.atmosres.2025.108203
14. Kusaka, H., Kondo, H., Kikegawa, Y., Kimura, F.: A simple single-layer urban canopy model for atmospheric models: Comparison with multi-layer and slab models. Boundary-Layer Meteorol. 101, 329–358 (2001). doi: 10.1023/A:1019207923078
15. Lac, C. et al.: Overview of the Meso-NH model version 5.4 and its applications. Geosci. Model Dev. 11(5), 1929-1969 (2018). doi: 10.5194/gmd-11-1929-2018
16. Lafore, J. P. et al.: The Meso-NH Atmospheric Simulation System. Part I: adiabatic formulation and control simulations. Ann. Geophys. 16(1), 90-109 (1998). doi:10.1007/s00585-997-0090-6
17. McCaul, E. W. Jr., Goodman, S. J., LaCasse, K. M., Cecil, D. J.: Forecasting Lightning Threat Using Cloud-Resolving Model Simulations. Weather Forecast. 24(3), 709-729 (2009). doi: 10.1175/2008WAF2222152.1
18. Mellor, G. L., Yamada, T.: Development of a turbulence closure model for geophysical fluid problems. Rev. Geophys. 20(4), 851-875 (1982). doi: 10.1029/RG020i004p00851

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1

19. Müller, E., Guo, X., Scheichl, R., Shi, S.: Matrix-free GPU implementation of a preconditioned conjugate gradient solver for anisotropic elliptic PDEs. Comput. Vis. Sci. 16(2), 41-58 (2013). doi: 10.1007/s00791-014-0223-x
20. Nakanishi, M., Niino, H.: An improved Mellor–Yamada level-3 model with condensation physics: Its design and verification. Boundary-Layer Meteorol. 112, 1–31 (2004). doi: 10.1023/B:BOUN.0000020164.04146.98
21. Nishizawa, S., Yashiro, H., Sato, Y., Miyamoto, Y., Tomita, H.: Influence of grid aspect ratio on planetary boundary layer turbulence in large-eddy simulations. Geosci. Model Dev. 8(10), 3393–3419 (2015). doi: 10.5194/gmd-8-3393-2015
22. Performance Development (TOP 500), https://www.top500.org/statistics/perfdevel/, accessed 30 November 2025 UTC
23. Risk Maps (JMA), https://www.jma.go.jp/bosai/en_risk/, accessed 19 February 2026 UTC
24. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003).
25. Saito, K. et al.: The Operational JMA Nonhydrostatic Mesoscale Model. Mon. Weather Rev. 134(4), 1266-1298 (2006). doi:10.1175/MWR3120.1
26. Sato, K., Inatsu, M.: Response of Snow Cloud Bands to Sea Surface Temperatures and Air Temperatures over the Sea of Japan. SOLA. 21, 69-75 (2025). doi:10.2151/sola.2025-009
27. Sato, Y., Nishizawa, S., Yashiro, H., Miyamoto, Y., Kajikawa, Y., Tomita, H.: Impacts of cloud microphysics on trade wind cumulus: Which cloud microphysics processes contribute to the diversity in a large eddy simulation? Prog. Earth Planet. Sci. 2, 23 (2015). doi:10.1186/s40645-015-0053-6
28. Sato, Y., Miyamoto, Y., Tomita, H.: Large dependency of charge distribution in a tropical cyclone inner core upon aerosol number concentration. Prog. Earth Planet. Sci. 6, 62 (2019). doi:10.1186/s40645-019-0309-7
29. Scotti, A., Meneveau, C., Lilly, D.K.: Generalized Smagorinsky model for anisotropic grids. Phys. Fluids A. 5(9), 2306–2308 (1993). doi:10.1063/1.858537
30. Sekiguchi, M., Nakajima, T.: A k-distribution-based radiation code and its computational optimization for an atmospheric general circulation model. J. Quant. Spectrosc. Radiat. Transf. 109(17-18), 2779–2793 (2008). doi:10.1016/j.jqsrt.2008.07.013
31. Skamarock, W.C. et al.: A description of the Advanced Research WRF Version 3. NCAR Tech. Note NCAR/TN–475+STR, National Center for Atmospheric Research, Boulder, CO (2008). doi:10.5065/D68S4MVH
32. Tomioka, T., Sato, Y., Hayashi, S., Yoshida, S., Iwashita, T.: Advantage of bulk lightning models for predicting lightning frequency over Japan. Prog. Earth Planet. Sci. 10, 60 (2023). doi:10.1186/s40645-023-00592-w
33. Tomita, H.: New microphysical schemes with five and six categories by diagnostic generation of cloud ice. J. Meteorol. Soc. Jpn. 86A, 121–142 (2008). doi:10.2151/jmsj.86A.121
34. Trottenberg, U., Oosterlee, C. W., Schüller, A.: Multigrid. Academic Press, San Diego, CA (2001).
35. Van der Vorst, H.A.: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. 13(2), 631–644 (1992). doi:10.1137/0913035

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29912-3_1