

A SELECTIVELY RELAXED ALTERNATING POSITIVE SEMIDEFINITE SPLITTING PRECONDITIONER FOR THE FLUX-LIMITED MULTI-GROUP RADIATION DIFFUSION EQUATIONS

XIAOQIANG YUE, RONG ZHOU, CHUNYAN CHEN, XIAOWEN XU*, AND SHI SHU*

Abstract. In this article, we concentrate on the fast numerical computation of the radiation energy densities together with electron and ion temperatures of three-dimensional multi-group radiation diffusion equations, which is temporally discretized with the adaptive backward Eulerian scheme, linearized iteratively via the method of frozen coefficients and spatially approximated through a cell-centered finite volume discretization on the adaptive unstructured computational meshes. We present, analyze and implement an alternating positive semidefinite splitting preconditioning technique with two selective relaxations and algebraic multigrid subsolves, and provide an algebraic quasi-optimal selection approach to determine the involved parameters. Our parallel implementation is based on the software package jxpang and the preconditioned flexible restarted generalized minimal residual solver has been examined by running realistic simulations of hydrodynamic instability on the Tianhe-2A supercomputer to demonstrate its numerical robustness, computational efficiency, parallel strong and weak scalabilities, and the competitiveness with some existing popular monolithic and block preconditioning strategies.

Key words. Radiation diffusion equations, alternating positive semidefinite splitting, selective relaxation, algebraic multigrid, parallel and distributed computing.

1. Introduction

On a spherically symmetrical bounded geometry, the flux-limited multi-group radiation diffusion (MGD) equations

$$(1) \quad \begin{cases} \frac{\partial E_g}{\partial t} = \nabla \cdot (D_g(E_g) \nabla E_g) + c(\sigma_{Bg} B_g(T_E) - \sigma_{Pg} E_g) + S_g, & g = 1, \dots, G, \\ \rho c_E \frac{\partial T_E}{\partial t} = \nabla \cdot (D_E(T_E) \nabla T_E) - c \sum_{g=1}^G (\sigma_{Bg} B_g(T_E) - \sigma_{Pg} E_g) + w_{IE}(T_I - T_E), \\ \rho c_I \frac{\partial T_I}{\partial t} = \nabla \cdot (D_I(T_I) \nabla T_I) - w_{IE}(T_I - T_E) \end{cases}$$

are the simplest and most extensively used approximation to the spatio-temporal orientation- and frequency-dependent thermal radiation transport equations, which compactly describe the propagations of high-energy photons in a physical system and the interactions with electrons directly and ions indirectly. It must be noticed that the thermal radiation transport process occurs in various branches of physics, such as the optical remote sensings, massive star formations and inertial confinement fusion experiments. The nonlinear PDE system (1) looks for the radiation energy density functions E_1, \dots, E_G , the electron temperature function T_E and the ion temperature function T_I for some given density of medium ρ , the specific heat capacities c_E and c_I , the nonlinear radiation diffusion coefficient $D_g(E_g)$, the

Received by the editors on October 8, 2023 and, accepted on November 28, 2024.
 2000 *Mathematics Subject Classification.* 65F10, 65N55, 65Y05, 65Z05.

*Corresponding author.

scattering and absorption coefficients σ_{B_g} and σ_{P_g} , the source item S_g and the electron scattering energy density $B_g(T_E)$ for the photon frequency group index $g = 1, \dots, G$, the nonlinear thermal-conductivity coefficients $D_I(T_I)$ and $D_E(T_E)$ together with the energy transfer coefficient w_{IE} . In most situations, the analytic solution of problem (1) could not be available for arbitrary geometries and parameters or this nonlinear PDE system may not be directly solvable [33]. As a result, it needs to be discretized in the temporal dimension, at first, with the adaptive backward Eulerian scheme, yielding a series of semi-discrete nonlinear systems of the form

$$(2) \quad \left\{ \begin{array}{l} -\nabla \cdot (D_g(E_g)\nabla E_g) + \left(\frac{1}{\Delta t_{k+1}} + c\sigma_{P_g}\right)E_g - c\sigma_{B_g}B_g(T_E) = S_g + \frac{1}{\Delta t_{k+1}}E_g^{(k)}, \\ \qquad \qquad \qquad g = 1, \dots, G, \\ -\nabla \cdot (D_E(T_E)\nabla T_E) + \left(\frac{\rho c_E}{\Delta t_{k+1}} + w_{IE}\right)T_E + c \sum_{g=1}^G \sigma_{B_g}B_g(T_E) \\ \qquad \qquad \qquad - c \sum_{g=1}^G \sigma_{P_g}E_g - w_{IE}T_I = \frac{\rho c_E}{\Delta t_{k+1}}T_E^{(k)}, \\ -\nabla \cdot (D_I(T_I)\nabla T_I) + \left(\frac{\rho c_I}{\Delta t_{k+1}} + w_{IE}\right)T_I - w_{IE}T_E = \frac{\rho c_I}{\Delta t_{k+1}}T_I^{(k)} \end{array} \right.$$

at the $(k+1)$ -th time level, where $\Delta t_{k+1} = t_{k+1} - t_k$ is the actual time-step size and each continuous item with superscript (k) represents the correlative approximation at the preceding time level. Then, the nonlinear semi-discrete system (2) is linearized iteratively through the method of frozen coefficients [24], where the term $B_g(T_E)$ is approximated by its first-order Taylor series expansion

$$B_g(T_E) \approx B_g^{(\delta)} + \left(\frac{\partial B_g}{\partial T_E}\right)^{(\delta)} (T_E - T_E^{(\delta)})$$

due to its tanglesome nonlinearity while the others are replaced by their constant (0th-order) Taylor approximations at $E_g^{(\delta)}$, $T_I^{(\delta)}$ and $T_E^{(\delta)}$. We immediately obtain a sequence of coupled systems of second-order linear reaction-diffusion equations as follows

$$(3) \quad \left\{ \begin{array}{l} -\nabla \cdot (D_g^{(\delta)}\nabla E_g) + \left(\frac{1}{\Delta t_{k+1}} + c\sigma_{P_g}^{(\delta)}\right)E_g - c\sigma_{B_g}^{(\delta)}\left(\frac{\partial B_g}{\partial T_E}\right)^{(\delta)} T_E \\ \qquad \qquad \qquad = S_g^{(\delta)} + \frac{1}{\Delta t_{k+1}}E_g^{(k)} + c\sigma_{B_g}^{(\delta)} \left[B_g^{(\delta)} - \left(\frac{\partial B_g}{\partial T_E}\right)^{(\delta)} T_E^{(\delta)} \right], \quad g = 1, \dots, G, \\ -\nabla \cdot (D_E^{(\delta)}\nabla T_E) + \left[\frac{\rho c_E^{(\delta)}}{\Delta t_{k+1}} + w_{IE}^{(\delta)} + \sum_{g=1}^G c\sigma_{B_g}^{(\delta)}\left(\frac{\partial B_g}{\partial T_E}\right)^{(\delta)} \right] T_E \\ \qquad \qquad \qquad - \sum_{g=1}^G c\sigma_{P_g}^{(\delta)}E_g - w_{IE}^{(\delta)}T_I = \frac{\rho c_E^{(\delta)}}{\Delta t_{k+1}}T_E^{(k)} - \sum_{g=1}^G c\sigma_{B_g}^{(\delta)} \left[B_g^{(\delta)} - \left(\frac{\partial B_g}{\partial T_E}\right)^{(\delta)} T_E^{(\delta)} \right], \\ -\nabla \cdot (D_I^{(\delta)}\nabla T_I) + \left(\frac{\rho c_I^{(\delta)}}{\Delta t_{k+1}} + w_{IE}^{(\delta)}\right)T_I - w_{IE}^{(\delta)}T_E = \frac{\rho c_I^{(\delta)}}{\Delta t_{k+1}}T_I^{(k)}, \end{array} \right.$$

at the $(\delta + 1)$ -th nonlinear iteration step, where the initial guess at the current time level t_{k+1} , i.e., the particular case when $\delta = 0$, is just the related approximate function at t_k . Finally, we discretize the linear PDE system (3) using a cell-centered locally conservative finite volume scheme, i.e., to approximate these functions by the undermentioned system of linear algebraic equations that involve a finite number of unknowns in the so-called field-by-field ordering

$$(4) \quad \mathbf{A}\mathbf{u} \equiv \begin{bmatrix} A_R & D_{RE} & O \\ D_{ER} & A_E & D_{EI} \\ O & D_{IE} & A_I \end{bmatrix} \begin{bmatrix} e_R \\ t_E \\ t_I \end{bmatrix} = \begin{bmatrix} f_R \\ f_E \\ f_I \end{bmatrix} \equiv \mathbf{f},$$

where the discrete radiation variables are integrated into a single entity, namely,

$$A_R = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_G \end{bmatrix} \in \mathbb{R}^{Gn \times Gn}, \quad D_{RE} = \begin{bmatrix} D_{1E} \\ \vdots \\ D_{GE} \end{bmatrix} \in \mathbb{R}^{Gn \times n},$$

$$e_R = \begin{bmatrix} e_1 \\ \vdots \\ e_G \end{bmatrix} \in \mathbb{R}^{Gn}, \quad f_R = \begin{bmatrix} f_1 \\ \vdots \\ f_G \end{bmatrix} \in \mathbb{R}^{Gn}$$

and $D_{ER} = (D_{E1}, \dots, D_{EG}) \in \mathbb{R}^{n \times Gn}$ while A_E, D_{EI}, A_I and D_{IE} are four real matrices of size $n \times n$. Here n is the total number of possibly unstructured mesh cells. It is worth emphasizing that

- all its diagonal sub-blocks A_g ($g = 1, \dots, G, E, I$), due to (3), originate from discrete reaction-diffusion operators with appropriate boundary conditions and involved coefficients in fairly different scales and discontinuities, resulting in a cluster of sparse symmetric positive definite but ill-conditioned and multi-scale matrices stored in compressed sparse row (CSR) storage formats with the same nonzero structure;
- all its nonzero off-diagonal sub-blocks $D_{gg'}$ ($g \neq g', g, g' = 1, \dots, G, E, I$) are diagonal matrices with non-positive elements, also in enormously different orders of magnitude, and they must satisfy

$$D_{EI} = D_{IE} \text{ and } D_{Eg} \neq D_{gE} \text{ for the photon frequency group index } g = 1, \dots, G,$$

such that the global matrix \mathbf{A} is multi-scale¹, positive definite but necessarily non-symmetric and multi-physics (namely radiation, electron and ion) coupled in intensively time- and position-varying strengths²;

- the number of degrees of freedom of the flux-limited MGD linear system (4) is usually ranged from 10^7 to 10^{11} as a result of the presence of hydrodynamic instabilities together with the wave-like propagation characteristics and multiple spatio-temporal scales in the transient solutions.

Obviously, the flexible restarted generalized minimal residual solver [36, 37], which is to restart after each cycle of m iteration steps and is denoted by FGMRES(m), is generally the method of choice, in despite of the intrinsic appeals (e.g., reliability and accessibility) of sparse direct solvers (e.g., MUMPS [2], PARDISO [40], PaStiX [18], SuperLU [27] and UMFPACK [12]), because of their considerable amounts of memory footprint and difficulties in developing massively parallel implementations.

¹A matrix is said to be multi-scale if its off-diagonal elements span several orders of magnitude. Otherwise, it is defined as having the single-scale property.

²This means that the coupling coefficients $c\sigma_{Bg}^{(\delta)} \left(\frac{\partial B_g}{\partial T_E} \right)^{(\delta)}$, $c\sigma_{Pg}^{(\delta)}$ ($g = 1, \dots, G$) and $w_{IE}^{(\delta)}$ in (3) change dramatically at different physical times and computational locations.

However, the convergence behavior of FGMRES(m) needs to be further boosted via some effective preconditioning algorithm, which invariably transforms the original system of linear equations into a mathematically equivalent linear system, nevertheless, with numerous more beneficial properties, e.g., a rather smaller (spectral) condition number and a much more clustered eigenvalue distribution.

In response to the aforementioned challenging task, numerous scholars have proposed a wide variety of preconditioning approaches over the past several decades, which are mainly categorized into three different types: the monolithic, block (also known as physics-based) and combined preconditioners [3, 5, 9, 16, 21, 22, 32, 41, 43, 44, 45, 47, 49, 51, 54, 48, 53, 50, 55, 56]. However, the accurate, efficient and scalable numerical iterative solution of the flux-limited MGD linear systems still present a formidable computational challenge, such as the relatively slow convergence performance [41] or even sporadic divergence [55] when the involved Schur complement matrices need to be approximated in the preconditioner construction phase. More often, the inverses of certain sub-matrices therein are replaced, in a straightforward manner, by their ‘justified’ diagonal counterparts, which gives rise to significant uncertainty on their veritable preconditioning effects. Another point that should be made is that the two general requirements of modern preconditioning techniques are the numerical robustness (in reference to the geometric, physical and discrete parameters and the number of parallel processor cores) and the implementation scalability (i.e., the setup phase and every iteration step ought to be scalable in a parallel environment) [11]. It should be emphasized that the numerical robustness is, without doubt, a preemptive requirement to arrive at a scalable implementation. Hence, sparse iterative linear solvers paired with advanced modern preconditioning strategies, without introducing any Schur complement matrix that cannot be tackled precisely, is just our sustaining demand for reliable and timely modeling predictions.

In the present article, we investigate the properties of the linear system to design such an advanced modern block preconditioning algorithm of the semi-algebraic category, namely, it does not require any knowledge except the flux-limited MGD coefficient matrix and its inherent physics-informed sparse block structure. Concretely speaking, a new alternating positive semidefinite splitting preconditioner with selective relaxations (APSS-SR) and algebraic multigrid subsolves for scalability is proposed, analyzed, implemented and examined in sequential and parallel circumstances. The body of this work is organized as follows. Section 2 provides a concise review of the research topic on alternating positive semidefinite splitting (APSS) preconditioners. Thereafter, we present the principal contribution of this article in Section 3, including the construction of the APSS-SR preconditioner induced by a two-sweep alternating direction implicit iteration scheme, spectral distribution results and the degree of the minimal polynomial of its preconditioned matrix, an algebraic quasi-optimal choice strategy on the involved parameters as well as its sequential implementation and the two-level parallelization. Numerical simulations involving realistic unstructured problems aimed at examining its numerical robustness, computational efficiency and parallel scaling properties are carried out in Section 4. Section 5 concludes the presentation with some closing remarks and future prospects.

2. Related works

The original idea of the APSS iteration method and its induced preconditioning algorithm is described in [35] for two-by-two block complex-valued saddle point

problems arising from time-harmonic eddy current models. The unconditional convergence analysis is also given in [35], nevertheless, only for the simple topology case. Subsequently, Ke, Ma and Ren proposed two improved APSS variants using different matrix splittings applied to the same model problems and proved their unconditional convergence properties in [26]. Afterwards, two two-parameter modified APSS variants are introduced separately for the simple and general topologies in [25] with their eigenvalue distributions and upper bounds about the degree of the minimal polynomial of the relevant preconditioned matrix. Two remarkable advantages of this approach are its good convergence and easy implementation. This type of algorithm has branched out into solving double saddle point problems [28, 30, 34] and three-by-three block singular and non-singular saddle point problems [4, 10, 39]. For the system of linear equations (4), the relaxed APSS preconditioner is then defined by

$$\begin{aligned}
 \tilde{\mathbf{P}} &= \frac{1}{\tilde{\alpha}} \begin{bmatrix} A_R & D_{RE} & O \\ D_{ER} & \tilde{\alpha}I_\pi & O \\ O & O & \tilde{\alpha}I_\pi \end{bmatrix} \begin{bmatrix} \tilde{\alpha}I_\pi & O & O \\ O & A_E & D_{EI} \\ O & D_{IE} & A_I \end{bmatrix} \\
 (5) \quad &= \begin{bmatrix} A_R & \frac{1}{\tilde{\alpha}}D_{RE}A_E & \frac{1}{\tilde{\alpha}}D_{RE}D_{EI} \\ D_{ER} & A_E & D_{EI} \\ O & D_{IE} & A_I \end{bmatrix},
 \end{aligned}$$

where $\tilde{\alpha}$ is a positive number and I_π represents the identity matrix with appropriate dimension. The preconditioner $\tilde{\mathbf{P}}$, which will be utilized for a comparison study in Section 4, is firstly derived from an alternating positive semidefinite splitting of the global matrix \mathbf{A} :

$$\mathbf{A} = \underbrace{\begin{bmatrix} A_R & D_{RE} & O \\ D_{ER} & O & O \\ O & O & O \end{bmatrix}}_{:=\tilde{\mathbf{A}}_1} + \underbrace{\begin{bmatrix} O & O & O \\ O & A_E & D_{EI} \\ O & D_{IE} & A_I \end{bmatrix}}_{:=\tilde{\mathbf{A}}_2},$$

and then induced by the two-step alternating direction implicit iteration scheme

$$\begin{cases} (\tilde{\alpha}I_\pi + \tilde{\mathbf{A}}_1)\mathbf{u}^{(k+\frac{1}{2})} = (\tilde{\alpha}I_\pi - \tilde{\mathbf{A}}_2)\mathbf{u}^{(k)} + \mathbf{f} \\ (\tilde{\alpha}I_\pi + \tilde{\mathbf{A}}_2)\mathbf{u}^{(k+1)} = (\tilde{\alpha}I_\pi - \tilde{\mathbf{A}}_1)\mathbf{u}^{(k+\frac{1}{2})} + \mathbf{f} \end{cases}, \quad k = 0, 1, \dots$$

on account of two next-mentioned splittings

$$\mathbf{A} = (\tilde{\alpha}I_\pi + \tilde{\mathbf{A}}_1) - (\tilde{\alpha}I_\pi - \tilde{\mathbf{A}}_2) = (\tilde{\alpha}I_\pi + \tilde{\mathbf{A}}_2) - (\tilde{\alpha}I_\pi - \tilde{\mathbf{A}}_1),$$

and finally improved by the so-called relaxed matrix splitting technique [8]. However, the easy implementation feature is gone because it brings the explicit generations and numerical inversions of two Schur complement matrices

$$S_R = A_R - \frac{1}{\tilde{\alpha}}D_{RE}D_{ER} \text{ and } S_E = A_E - D_{EI}A_I^{-1}D_{IE}$$

in solving the general residual equations $\tilde{\mathbf{P}}\mathbf{w} = \mathbf{b}$, which is accomplished by the following procedure:

- (1) solve w_R from $S_R w_R = b_R - \frac{1}{\tilde{\alpha}}D_{RE}b_E$;
- (2) solve z_I from $A_I z_I = b_I$;
- (3) compute $y_E := b_E - D_{ER}w_R - D_{EI}z_I$ and solve w_E from $S_E w_E = y_E$;
- (4) set $p_I := D_{IE}w_E$, solve $A_I q_I = p_I$ and compute $w_I = z_I - q_I$.

This is based on the block-matrix factorization

$$\begin{bmatrix} \frac{1}{\tilde{\alpha}} S_R & \frac{1}{\tilde{\alpha}} D_{RE} & O \\ O & I_\pi & O \\ O & O & I_\pi \end{bmatrix} \begin{bmatrix} I_\pi & O & O \\ \frac{1}{\tilde{\alpha}} D_{ER} & I_\pi & O \\ O & O & I_\pi \end{bmatrix} \begin{bmatrix} \tilde{\alpha} I_\pi & O & O \\ O & S_E & D_{EI} \\ O & O & A_I \end{bmatrix} \begin{bmatrix} I_\pi & O & O \\ O & I_\pi & O \\ O & A_I^{-1} D_{IE} & I_\pi \end{bmatrix} \begin{bmatrix} w_R \\ w_E \\ w_I \end{bmatrix} = \begin{bmatrix} b_R \\ b_E \\ b_I \end{bmatrix},$$

where $\mathbf{w} = (w_R^\top, w_E^\top, w_I^\top)^\top$ is the outgoing Krylov solution vector and $\mathbf{b} = (b_R^\top, b_E^\top, b_I^\top)^\top$ is an arbitrarily incoming Krylov known vector. It is worthwhile to point out that the inverse A_I^{-1} in the Schur complement matrix S_E is frequently approximated by a diagonal matrix, such as the direct, row-maximum, row-infinity and row-Schur diagonalizations (see [51, 56, 54]), which may seriously influence the convergence rate of the left- or right-preconditioned FGMRES(m) solver. Another remark is that the portion $D_{RE}D_{ER}$ causes S_R to be a full block-matrix

$$S_R = \begin{bmatrix} A_1 - \frac{1}{\tilde{\alpha}} D_{1E} D_{E1} & \cdots & -\frac{1}{\tilde{\alpha}} D_{1E} D_{EG} \\ \vdots & \ddots & \vdots \\ -\frac{1}{\tilde{\alpha}} D_{GE} D_{E1} & \cdots & A_G - \frac{1}{\tilde{\alpha}} D_{GE} D_{EG} \end{bmatrix}$$

whose off-diagonal sub-blocks $-\frac{1}{\tilde{\alpha}} D_{jE} D_{Ek}$ ($j \neq k, j, k = 1, \dots, G$) are undoubtedly all diagonal matrices and have, by and large, been neglected when addressing the first step of the above-mentioned algorithmic procedure. Moreover, the practical, algebraic and quasi-optimal choice of the positive parameter $\tilde{\alpha}$ will be discussed in subsection 3.3.

Remark 1. *It is easily seen from the above procedure that (1) an application of $\tilde{\mathbf{P}}^{-1}$ on \mathbf{b} to obtain \mathbf{w} is formed from two subsolves with the coefficient matrix A_I as well as one subsolve with the coefficient matrices S_R and S_E ; (2) in a practical implementation, the involved inverses are achieved by one or two algebraic multigrid V-cycles, rather than performed to calculate their exact solutions by sparse Cholesky or LU factorization in an approximate minimum degree (AMD) or a column AMD reordering.*

3. A selectively relaxed alternating positive semidefinite splitting preconditioner

In this section, we establish a new APSS preconditioner with two selective relaxations for solving the flux-limited MGD linear system (4), provide the eigenvalue and eigenvector distribution results with respect to its left- or right-preconditioned matrix, derive an algebraic, quasi-optimal and easy-to-implement estimation formulae for the involved positive parameters and discuss the sequential algorithmic implementation as well as the two-level parallelization for the preconditioner on the strength of the jxpamg (parallel algebraic multigrid solvers and preconditioners developed by JiuSuo and XTU) software library [46]³.

³It had already served as a scalable third-party library of jasmin (j adaptive structured meshes applications infrastructure) [31] and jaumin (j adaptive unstructured meshes applications infrastructure) [29]. Moreover, we detailed in [46] its algorithms, parallel implementation techniques, software architecture, user interfaces as well as typical applications.

3.1. Preconditioner constitution. The above two disadvantages of the relaxed APSS preconditioning algorithm, when solving the block $(G + 2)$ -by- $(G + 2)$ linear system (4), can be remedied immediately by introducing a different alternating positive semidefinite splitting

$$(6) \quad \mathbf{A} = \underbrace{\begin{bmatrix} A_R & O & O \\ D_{ER} & A_E & O \\ O & O & O \end{bmatrix}}_{:=\mathbf{A}_1} + \underbrace{\begin{bmatrix} O & D_{RE} & O \\ O & O & D_{EI} \\ O & D_{IE} & A_I \end{bmatrix}}_{:=\mathbf{A}_2}$$

and a new-style two-step alternating direction implicit iteration scheme

$$(7) \quad \begin{cases} (\mathbf{\Lambda}_1 + \mathbf{A}_1)\mathbf{u}^{(k+\frac{1}{2})} = (\mathbf{\Lambda}_1 - \mathbf{A}_2)\mathbf{u}^{(k)} + \mathbf{f} \\ (\mathbf{\Lambda}_2 + \mathbf{A}_2)\mathbf{u}^{(k+1)} = (\mathbf{\Lambda}_2 - \mathbf{A}_1)\mathbf{u}^{(k+\frac{1}{2})} + \mathbf{f} \end{cases}, \quad k = 0, 1, \dots$$

based on the two splittings

$$\mathbf{A} = (\mathbf{\Lambda}_1 + \mathbf{A}_1) - (\mathbf{\Lambda}_1 - \mathbf{A}_2) = (\mathbf{\Lambda}_2 + \mathbf{A}_2) - (\mathbf{\Lambda}_2 - \mathbf{A}_1)$$

with two selective relaxations

$$(8) \quad \mathbf{\Lambda}_1 = \begin{bmatrix} O & O & O \\ O & O & O \\ O & O & \alpha I_\pi \end{bmatrix} \quad \text{and} \quad \mathbf{\Lambda}_2 = \begin{bmatrix} \beta I_\pi & O & O \\ O & \gamma I_\pi & O \\ O & O & O \end{bmatrix},$$

including three positive parameters α , β and γ in order to guarantee that the coefficient matrices $\mathbf{\Lambda}_1 + \mathbf{A}_1$ and $\mathbf{\Lambda}_2 + \mathbf{A}_2$ are both nonsingular. Eliminating the intermediate unknown $\mathbf{u}^{(k+\frac{1}{2})}$ in (7) yields

$$\begin{aligned} \mathbf{u}^{(k+1)} &= \underbrace{(\mathbf{\Lambda}_2 + \mathbf{A}_2)^{-1}(\mathbf{\Lambda}_2 - \mathbf{A}_1)(\mathbf{\Lambda}_1 + \mathbf{A}_1)^{-1}(\mathbf{\Lambda}_1 - \mathbf{A}_2)}_{:=\mathbf{G}} \mathbf{u}^{(k)} \\ &\quad + \underbrace{(\mathbf{\Lambda}_2 + \mathbf{A}_2)^{-1}(\mathbf{\Lambda}_2 + \mathbf{A}_1)(\mathbf{\Lambda}_1 + \mathbf{A}_1)^{-1}}_{:=\mathbf{P}^{-1}} \mathbf{f}, \end{aligned}$$

where the iteration matrix can be rewritten in the form

$$\mathbf{G} = [\mathbf{I}_\pi - (\mathbf{\Lambda}_2 + \mathbf{A}_2)^{-1}\mathbf{A}] [\mathbf{I}_\pi - (\mathbf{\Lambda}_1 + \mathbf{A}_1)^{-1}\mathbf{A}] = \mathbf{I}_\pi - \mathbf{P}^{-1}\mathbf{A}$$

and our APSS-SR preconditioning matrix, by the definitions (6) and (8), is defined through

$$(9) \quad \begin{aligned} \mathbf{P} &= (\mathbf{\Lambda}_1 + \mathbf{A}_1)(\mathbf{\Lambda}_2 + \mathbf{A}_1)^{-1}(\mathbf{\Lambda}_2 + \mathbf{A}_2) \\ &= \begin{bmatrix} A_R & \frac{1}{\beta}A_R D_{RE} & O \\ D_{ER} & \frac{1}{\beta}D_{ER} D_{RE} + A_E & \frac{1}{\gamma}A_E D_{EI} \\ O & D_{IE} & A_I \end{bmatrix} \end{aligned}$$

associated with the iteration scheme (7). It follows from (9) that the parameter α has no effect on the convergence performance of the iteration scheme (7) or the preconditioning behavior of \mathbf{P} , namely, what we have developed are a degenerate two-parameter selectively relaxed APSS iteration method and its related preconditioner.

3.2. Spectral properties of the preconditioned matrix. As known to all, the convergence performance of the preconditioned FGMRES(m) solver is not only mightily susceptible to the eigenvalue distribution of its preconditioned matrix, but also remarkably correlated with the condition number of the corresponding eigenvector matrix [7]. With regard to the APSS-SR left-preconditioned matrix $\mathbf{P}^{-1}\mathbf{A}$, or equivalently, the APSS-SR right-preconditioned variant $\mathbf{A}\mathbf{P}^{-1}$, the eigenvalue

and eigenvector distribution results are stated in the following theorem, where $\mathcal{N}(\cdot)$ represents the null space of the relevant matrix.

Theorem 1. *Let sub-matrices A_R , D_{RE} , A_E , D_{ER} , D_{EI} , A_I and D_{IE} be defined by (4), and β and γ be two arbitrarily given positive parameters. Then, the APSS-SR preconditioned matrices $\mathbf{P}^{-1}\mathbf{A}$ and $\mathbf{A}\mathbf{P}^{-1}$ both have the eigenvalue 1 with algebraic multiplicity at least $(G+1)n$ while their remaining eigenvalues are of the form $1 - \mu_j$ with μ_j being the j -th eigenvalue of the real $n \times n$ matrix*

$$(10) \quad Z_\gamma := A_E^{-1}D_{ER}A_R^{-1}D_{RE}(I_\pi + \frac{1}{\gamma}D_{EI}S_I^{-1}D_{IE}) - (\frac{1}{\gamma}I_\pi - A_E^{-1})D_{EI}S_I^{-1}D_{IE}.$$

where the Schur complement matrix

$$S_I = A_I - \frac{1}{\gamma}D_{IE}D_{EI}.$$

Furthermore, assume that the matrix D_{EI} and at least one of the sub-matrices D_{gE} ($g = 1, \dots, G$) are full row-rank, then $\mathbf{P}^{-1}\mathbf{A}$ has the following $Gn + s$ ($0 \leq s \leq 2n$) linearly independent eigenvectors:

- Gn eigenvectors of the form $[u_l^\top, 0^\top, 0^\top]^\top$ ($l = 1, \dots, Gn$) associate with the eigenvalue 1, where $u_l \in \mathbb{R}^{Gn}$ are arbitrary linearly independent vectors;
- s eigenvectors of the form $[\hat{u}_l^\top, \hat{v}_l^\top, \hat{w}_l^\top]^\top$ ($l = 1, 2, \dots, s$) correspond to the eigenvalues $\lambda_l \neq 1$, where $\hat{v}_l \in \mathcal{N}(Q_l) \setminus \{0\}$ with

$$Q_l = (\lambda_l - 1)A_E + D_{ER}A_R^{-1}D_{RE} + D_{EI}A_I^{-1}D_{IE} - \frac{\lambda_l}{\gamma}A_E D_{EI}A_I^{-1}D_{IE}$$

while \hat{u}_l and \hat{w}_l are given by

$$\hat{u}_l = \frac{1}{1 - \lambda_l}(\frac{\lambda_l}{\beta}I_\pi - A_R^{-1})D_{RE}\hat{v}_l \text{ and } \hat{w}_l = -A_I^{-1}D_{IE}\hat{v}_l.$$

Proof. As a result of (9), the respective difference below can be derived

$$\mathbf{R} = \mathbf{P} - \mathbf{A} = \begin{bmatrix} O & (\frac{1}{\beta}A_R - I_\pi)D_{RE} & O \\ O & \frac{1}{\beta}D_{ER}D_{RE} & (\frac{1}{\gamma}A_E - I_\pi)D_{EI} \\ O & O & O \end{bmatrix}.$$

With making use of the undermentioned block factorization

$$(11) \quad \mathbf{P} = \begin{bmatrix} A_R & O & O \\ D_{ER} & A_E & O \\ O & D_{IE} & I_\pi \end{bmatrix} \begin{bmatrix} I_\pi & \frac{1}{\beta}D_{RE} & O \\ O & I_\pi & \frac{1}{\gamma}D_{EI} \\ O & O & S_I \end{bmatrix},$$

it is easy to verify that

$$\mathbf{P}^{-1} = \begin{bmatrix} I_\pi & -\frac{1}{\beta}D_{RE} & \frac{1}{\beta\gamma}D_{RE}D_{EI}S_I^{-1} \\ O & I_\pi & -\frac{1}{\gamma}D_{EI}S_I^{-1} \\ O & O & S_I^{-1} \end{bmatrix} \\ \times \begin{bmatrix} A_R^{-1} & O & O \\ -A_E^{-1}D_{ER}A_R^{-1} & A_E^{-1} & O \\ D_{IE}A_E^{-1}D_{ER}A_R^{-1} & -D_{IE}A_E^{-1} & I_\pi \end{bmatrix}$$

and

$$(12) \quad \mathbf{P}^{-1}\mathbf{R} = \begin{bmatrix} O & \Theta_1 & \Theta_2 \\ O & \Upsilon_{11} & \Upsilon_{12} \\ O & \Upsilon_{21} & \Upsilon_{22} \end{bmatrix}$$

where

$$\begin{aligned}
 \Theta_1 &= \left(\frac{1}{\beta}I_\pi - A_R^{-1}\right)D_{RE} - \frac{1}{\beta}D_{RE}A_E^{-1}D_{ER}A_R^{-1}D_{RE} \\
 &\quad - \frac{1}{\beta\gamma}D_{RE}D_{EI}S_I^{-1}D_{IE}A_E^{-1}D_{ER}A_R^{-1}D_{RE}, \\
 \Theta_2 &= -\frac{1}{\beta}D_{RE}\left(\frac{1}{\gamma}I_\pi - A_E^{-1}\right)D_{EI} - \frac{1}{\beta\gamma}D_{RE}D_{EI}S_I^{-1}D_{IE}\left(\frac{1}{\gamma}I_\pi - A_E^{-1}\right)D_{EI}, \\
 \Upsilon_{11} &= \left(I_\pi + \frac{1}{\gamma}D_{EI}S_I^{-1}D_{IE}\right)A_E^{-1}D_{ER}A_R^{-1}D_{RE}, \\
 \Upsilon_{12} &= \left(I_\pi + \frac{1}{\gamma}D_{EI}S_I^{-1}D_{IE}\right)\left(\frac{1}{\gamma}I_\pi - A_E^{-1}\right)D_{EI}, \\
 \Upsilon_{21} &= -S_I^{-1}D_{IE}A_E^{-1}D_{ER}A_R^{-1}D_{RE}, \\
 \Upsilon_{22} &= -S_I^{-1}D_{IE}\left(\frac{1}{\gamma}I_\pi - A_E^{-1}\right)D_{EI}.
 \end{aligned}$$

Observe from the block structure of $\mathbf{P}^{-1}\mathbf{R}$ that its eigenvalues are 0 with algebraic multiplicity Gn and those of the bottom right $2n \times 2n$ sub-matrix

$$\begin{aligned}
 Y_\gamma &:= \begin{bmatrix} \Upsilon_{11} & \Upsilon_{12} \\ \Upsilon_{21} & \Upsilon_{22} \end{bmatrix} \\
 &= \underbrace{\begin{bmatrix} I_\pi + \frac{1}{\gamma}D_{EI}S_I^{-1}D_{IE} \\ -S_I^{-1}D_{IE} \end{bmatrix}}_{:=U_\gamma} \underbrace{\begin{bmatrix} A_E^{-1}D_{ER}A_R^{-1}D_{RE} & \left(\frac{1}{\gamma}I_\pi - A_E^{-1}\right)D_{EI} \end{bmatrix}}_{:=V_\gamma},
 \end{aligned}$$

which has the same nonzero eigenvalues as Z_γ defined in (10) and calculated by the formula $Z_\gamma = V_\gamma U_\gamma$ while the other eigenvalues are 0 with algebraic multiplicity at least n . Therefore, the first desired result can be directly verified through the relation

$$\mathbf{P}^{-1}\mathbf{A} = \mathbf{P}^{-1}(\mathbf{P} - \mathbf{R}) = \mathbf{I}_\pi - \mathbf{P}^{-1}\mathbf{R}$$

and the similarity transformation $\mathbf{A}\mathbf{P}^{-1} = \mathbf{P}(\mathbf{P}^{-1}\mathbf{A})\mathbf{P}^{-1}$ between $\mathbf{A}\mathbf{P}^{-1}$ and $\mathbf{P}^{-1}\mathbf{A}$.

The second portion of this theorem can now be proved. Let λ be an eigenvalue of $\mathbf{P}^{-1}\mathbf{A}$ and $\mathbf{q} = [u^\top, v^\top, w^\top]^\top$ be the eigenvector associated with it. From the relation $\mathbf{P}^{-1}\mathbf{A}\mathbf{q} = \lambda\mathbf{q}$, we get

$$\begin{aligned}
 &\begin{bmatrix} A_R & D_{RE} & O \\ D_{ER} & A_E & D_{EI} \\ O & D_{IE} & A_I \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \\
 &= \lambda \begin{bmatrix} A_R & \frac{1}{\beta}A_R D_{RE} & O \\ D_{ER} & \frac{1}{\beta}D_{ER}D_{RE} + A_E & \frac{1}{\gamma}A_E D_{EI} \\ O & D_{IE} & A_I \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix},
 \end{aligned}$$

which can equivalently be reformulated into

$$(13) \quad \begin{cases} (\lambda - 1)A_R u + \left(\frac{\lambda}{\beta}A_R - I_\pi\right)D_{RE}v = 0 \\ (\lambda - 1)D_{ER}u + (\lambda - 1)A_E v + \frac{\lambda}{\beta}D_{ER}D_{RE}v + \left(\frac{\lambda}{\gamma}A_E - I_\pi\right)D_{EI}w = 0 \\ (\lambda - 1)(D_{IE}v + A_I w) = 0 \end{cases}$$

We first assume that $\lambda = 1$. Then (13) can be reduced to

$$\begin{cases} (\frac{1}{\beta}A_R - I_\pi)D_{RE}v = 0 \\ \frac{1}{\beta}D_{ER}D_{RE}v + (\frac{1}{\gamma}A_E - I_\pi)D_{EI}w = 0 \end{cases}.$$

Since $\frac{1}{\beta}A_R - I_\pi$ is nonsingular, this shows that $v \in \mathcal{N}(D_{RE})$. Moreover, if D_{EI} and at least one of the sub-matrices D_{gE} ($g = 1, \dots, G$) are full row-rank, then $v = 0$ and $w = 0$ as a result of the nonsingularity of $\frac{1}{\gamma}A_E - I_\pi$. As a consequence, there are Gn linearly independent eigenvectors $[u_l^\top, 0^\top, 0^\top]^\top$ ($l = 1, \dots, Gn$) corresponding to the eigenvalue 1, where u_l are arbitrary linearly independent vectors in \mathbb{R}^{Gn} . Secondly, under the assumption that $\lambda \neq 1$, we have

$$w = -A_I^{-1}D_{IE}v \text{ and } u = \frac{1}{1-\lambda}(\frac{\lambda}{\beta}I_\pi - A_R^{-1})D_{RE}v$$

from the third and first equations in (13), respectively. Substituting these two expressions into the second equation of (13) yields $v \in \mathcal{N}(Q) \setminus \{0\}$, where

$$Q = (\lambda - 1)A_E + D_{ER}A_R^{-1}D_{RE} + D_{EI}A_I^{-1}D_{IE} - \frac{\lambda}{\gamma}A_E D_{EI}A_I^{-1}D_{IE}.$$

Since v is nonzero, this yields s linearly independent eigenvectors $[\hat{u}_l^\top, \hat{v}_l^\top, \hat{w}_l^\top]^\top$ ($l = 1, 2, \dots, s$) which correspond to the eigenvalue $\lambda_l \neq 1$. Finally, we prove that the above $Gn + s$ eigenvectors are linearly independent, i.e.,

$$(14) \quad \underbrace{\begin{bmatrix} u_1 & \cdots & u_{Gn} \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix}}_{:=U} \underbrace{\begin{bmatrix} \vartheta_1 \\ \vdots \\ \vartheta_{Gn} \end{bmatrix}}_{:=\vartheta} + \underbrace{\begin{bmatrix} \hat{u}_1 & \cdots & \hat{u}_s \\ \hat{v}_1 & \cdots & \hat{v}_s \\ \hat{w}_1 & \cdots & \hat{w}_s \end{bmatrix}}_{:=\hat{U}} \underbrace{\begin{bmatrix} \hat{\vartheta}_1 \\ \vdots \\ \hat{\vartheta}_s \end{bmatrix}}_{:=\hat{\vartheta}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

is valid only when the vectors ϑ and $\hat{\vartheta}$ are both zero. Multiplying all members of the above equation from the left by $\mathbf{P}^{-1}\mathbf{A}$ leads to

$$(15) \quad \begin{bmatrix} u_1 & \cdots & u_{Gn} \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \vartheta_1 \\ \vdots \\ \vartheta_{Gn} \end{bmatrix} + \begin{bmatrix} \hat{u}_1 & \cdots & \hat{u}_s \\ \hat{v}_1 & \cdots & \hat{v}_s \\ \hat{w}_1 & \cdots & \hat{w}_s \end{bmatrix} \begin{bmatrix} \lambda_1 \hat{\vartheta}_1 \\ \vdots \\ \lambda_s \hat{\vartheta}_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

The reason is that the first matrix U in (14) is composed of the eigenvectors associated with the eigenvalue 1 while the matrix \hat{U} is made up of the eigenvectors corresponding to the eigenvalues that are different from 1, both column by column. By subtracting (14) from (15), it holds that

$$\begin{bmatrix} \hat{u}_1 & \cdots & \hat{u}_s \\ \hat{v}_1 & \cdots & \hat{v}_s \\ \hat{w}_1 & \cdots & \hat{w}_s \end{bmatrix} \begin{bmatrix} (\lambda_1 - 1)\hat{\vartheta}_1 \\ \vdots \\ (\lambda_s - 1)\hat{\vartheta}_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

The linear independence of s eigenvectors $[\hat{u}_l^\top, \hat{v}_l^\top, \hat{w}_l^\top]^\top$ ($l = 1, 2, \dots, s$) and the eigenvalues $\lambda_l \neq 1$ for $l = 1, \dots, s$ can be used to obtain $\hat{\vartheta} = 0$. Substituting this into (14) yields $\vartheta = 0$, which completes the proof. \blacksquare

One of the best-known sufficient conditions to search an effective preconditioning strategy should capacitate us to lower the degree of the minimal polynomial of the preconditioned matrix as much as possible [42]. Next, we conclude an upper bound

for the degree of the minimal polynomial of the APSS-SR preconditioned matrix $\mathbf{P}^{-1}\mathbf{A}$ by exploiting Theorem 1 in a similar way to [6, Proposition 2.2] and [17, Theorem 4].

Theorem 2. *Assume that the condition of Theorem 1 holds. Then, the degree of the minimal polynomial of $\mathbf{P}^{-1}\mathbf{A}$, i.e., the dimension of the Krylov subspace $\mathcal{K}(\mathbf{P}^{-1}\mathbf{A}, \mathbf{f} - \mathbf{A}\mathbf{u}_0)$ for an arbitrarily given initial guess vector \mathbf{u}_0 , is no more than $2n + 1$, where n is the size of each sub-matrix of \mathbf{A} .*

3.3. An algebraic selection strategy for determining the involved two parameters. If these two parameters β and γ are such two values that \mathbf{P} is as close to \mathbf{A} as possible, then the block $(G+2)$ -by- $(G+2)$ matrix \mathbf{R} would be approaching the zero matrix. Therefore, it can be seen that the iteration scheme (7) would gain a fast convergence behavior and the preconditioned matrix is typically with a nice clustering of its eigenvalues. In short, now consider a similar idea to [23] in which the two parameters β and γ can be sought to minimize the objective function

$$\eta(\beta, \gamma) = \|\mathbf{R}\|_F^2 = \text{trace}(\mathbf{R}\mathbf{R}^\top).$$

Then, by immediate calculations, we obtain

$$\begin{aligned} \eta(\beta, \gamma) &= \text{trace}\left(\left(\frac{1}{\beta}A_R - I_\pi\right)D_{RE}D_{RE}^\top\left(\frac{1}{\beta}A_R - I_\pi\right)\right. \\ &\quad \left. + \frac{1}{\beta^2}D_{ER}D_{RE}D_{RE}^\top D_{ER}^\top + \left(\frac{1}{\gamma}A_E - I_\pi\right)D_{EI}^2\left(\frac{1}{\gamma}A_E - I_\pi\right)\right) \\ (16) \quad &= \frac{1}{\beta^2}k_1 - \frac{1}{\beta}k_2 + \frac{1}{\gamma^2}k_3 - \frac{1}{\gamma}k_4 + k_5 \\ &= k_1\left(\frac{1}{\beta} - \frac{k_2}{2k_1}\right)^2 + k_3\left(\frac{1}{\gamma} - \frac{k_4}{2k_3}\right)^2 + k_5 - \frac{k_2^2}{4k_1} - \frac{k_4^2}{4k_3} \end{aligned}$$

where

$$\begin{aligned} k_1 &= \text{trace}(A_R D_{RE} D_{RE}^\top A_R + D_{ER} D_{RE} D_{RE}^\top D_{ER}^\top), \\ k_2 &= \text{trace}(A_R D_{RE} D_{RE}^\top + D_{RE} D_{RE}^\top A_R), \\ k_3 &= \text{trace}(A_E D_{EI}^2 A_E), \\ k_4 &= \text{trace}(A_E D_{EI}^2 + D_{EI}^2 A_E), \\ k_5 &= \text{trace}(D_{RE} D_{RE}^\top + D_{EI}^2). \end{aligned}$$

From (16), it can be deduced that

$$(17) \quad \beta^* = \frac{2k_1}{k_2} \quad \text{and} \quad \gamma^* = \frac{2k_3}{k_4}$$

are the desired quasi-optimal choices.

Remark 2. *It is able to be derived in exactly the same way that the quasi-optimal choice of the positive parameter $\tilde{\alpha}$ in the relaxed APSS preconditioner $\tilde{\mathbf{P}}$, defined by (5), is*

$$\tilde{\alpha}^* = \frac{\text{trace}(D_{RE} A_E^2 D_{RE}^\top + D_{RE} D_{EI}^2 D_{RE}^\top)}{\text{trace}(D_{RE} A_E D_{RE}^\top)}.$$

By comparing with the relaxed APSS preconditioner $\tilde{\mathbf{P}}$ defined by (5), there are two positive parameters β and γ which serve the purpose of approximating A_R and A_E in the APSS-SR preconditioner \mathbf{P} defined by (9) while only one positive parameter $\tilde{\alpha}$ is in $\tilde{\mathbf{P}}$. From this fact one should expect the flexible restarted GMRES solver preconditioned by \mathbf{P} to converge in a smaller number of iterations.

3.4. Sequential and parallel implementation procedures. When the flexible restarted GMRES solver with the APSS-SR preconditioning algorithm \mathbf{P} is applied to solve the flux-limited MGD linear system (4) with $(G+2)$ -by- $(G+2)$ sparse block structure, we need to carry out a solution procedure for the following generalized residual linear system at each step

$$(18) \quad \mathbf{P}\mathbf{w} \equiv \begin{bmatrix} A_R & \frac{1}{\beta}A_RD_{RE} & O \\ D_{ER} & \frac{1}{\beta}D_{ER}D_{RE} + A_E & \frac{1}{\gamma}A_ED_{EI} \\ O & D_{IE} & A_I \end{bmatrix} \begin{bmatrix} w_R \\ w_E \\ w_I \end{bmatrix} = \begin{bmatrix} b_R \\ b_E \\ b_I \end{bmatrix} \equiv \mathbf{b},$$

where \mathbf{w} and \mathbf{b} are respectively the unknown outgoing Krylov solution vector and an arbitrarily given incoming Krylov right-hand side vector. By utilizing the matrix decomposition (11), the generalized residual equations (18) must be coped with by consecutively solving two undermentioned systems of linear equations

$$\begin{bmatrix} A_R & O & O \\ D_{ER} & A_E & O \\ O & D_{IE} & I_\pi \end{bmatrix} \begin{bmatrix} u_R \\ u_E \\ u_I \end{bmatrix} = \begin{bmatrix} b_R \\ b_E \\ b_I \end{bmatrix}$$

and

$$\begin{bmatrix} I_\pi & \frac{1}{\beta}D_{RE} & O \\ O & I_\pi & \frac{1}{\gamma}D_{EI} \\ O & O & A_I - \frac{1}{\gamma}D_{IE}D_{EI} \end{bmatrix} \begin{bmatrix} w_R \\ w_E \\ w_I \end{bmatrix} = \begin{bmatrix} u_R \\ u_E \\ u_I \end{bmatrix},$$

namely, the specifically algorithmic implementation procedure to figure up the generalized residual solution vector \mathbf{w} is described below:

- (1) solve u_R from $A_R u_R = b_R$;
- (2) compute $\tilde{b}_E := b_E - D_{ER}u_R$ and solve u_E from $A_E u_E = \tilde{b}_E$;
- (3) calculate $u_I := b_I - D_{IE}u_E$ and solve w_I from $(A_I - \frac{1}{\gamma}D_{IE}D_{EI})w_I = u_I$;
- (4) set $w_E := u_E - \frac{1}{\gamma}D_{EI}w_I$ and $w_R := u_R - \frac{1}{\beta}D_{RE}w_E$.

It can be observed from the above procedure that there are $G+2$ sparse linear sub-systems in the same nonzero structure to be solved by either a specified number of the best practices algebraic multigrid V-cycles [13] or iterating this type of algebraic multigrid V-cycles till a prescribed relative or absolute tolerance is reached.

Remark 3. *It is worth highlighting from the implementation procedure of $\tilde{\mathbf{P}}\mathbf{w} = \mathbf{b}$ in Section 2 that (1) $G+3$ sparse sub-matrix inverses must be operated for $\tilde{\mathbf{P}}$ (hence, in a more expensive computational overhead per iteration) and (2) the Schur complement matrices in $\tilde{\mathbf{P}}$ are just approximated while the one in \mathbf{P} can be explicitly formed, as $A_I - \frac{1}{\gamma}D_{IE}D_{EI}$ is shaped like A_I with some modifications to its diagonal entries, because the contribution item $-\frac{1}{\gamma}D_{IE}D_{EI}$ is just a diagonal matrix.*

Because of the advancement of the massively distributed-memory supercomputing, parallel computer simulations have become an enabling technology supporting a wide range of supersized applications in science and engineering, where the superfine computational mesh with an ultrahigh mesh resolution must be used to resolve very small spatial scales (e.g., not a few defects in geometry, splicing spots of different materials and sudden but drastic changes in values of many physical quantities at a certain region) to offer sufficiently accurate numerical solutions when simulating the hydrodynamic Rayleigh-Taylor instability during the deceleration phase of laser indirect-driven spherical implosions [15]. Also, appropriately designed, extraordinarily efficient and highly parallel preconditioners are in urgent need for an effective utilization of modern hardware resources, which are interconnected via multifarious

high-speed networks and carry out data exchanges among parallel processor cores by utilizing the message passing interface (MPI) library. It is apparent by now that the increase in the computing power is no longer from faster processors, but from the rapid increase in the number of physical processors, which makes numerical and parallel scalabilities of the preconditioned FGMRES(m) algorithm much more important than ever. It should be mentioned that the most momentous component is the preconditioner, without which the FGMRES(m) solver does not converge or converges quite slowly, and a good choice of preconditioning algorithm would accelerate the convergence significantly. Furthermore, it is supposed to be scalable to a good deal of parallel processor cores and to guarantee the numerical robustness with respect to different geometries, physical parameters, spatio-temporal discrete meshes and processor counts. However, three issues needed to be addressed are the design of an MPI processor topology for the preconditioning algorithm, the parallel matrix and vector data structures together with the communication overhead of data flows. Here, we point out that the two-level parallelization strategy described below is ideally suitable for effectively implementing the APSS-SR preconditioner \mathbf{P} defined by (9) in parallel to bring down the data communication overhead as much as possible:

- Within the first-level parallelization stage, we need to uniformly partition the global communicator, which is assumed to be composed of $(G + 2)q$ parallel processor cores used for the parallel computing, into $G + 2$ communication sub-groups. This type of procedure is done by the MPI function ‘MPI_Comm_split’. The communication sub-groups are distinguished by their owned ‘color’ value, i.e., COMM_R (consisting of COMM_1, \dots , COMM_G), COMM_I and COMM_E, and their respective q parallel processor cores are labeled as their local continuous ‘key’ values (starting from 0 to $q - 1$), as illustrated in Figure 1. Each sparse linear subsystem along with the correlative off-diagonal sub-blocks (stored in the vector form) are assigned to one communication sub-group according to the sequence number of physical variables.
- The principal mission and responsibility of the second-level parallelization stage is to distribute the rows of the sub-matrix A_g , all of nonzero sub-vectors $D_{gg'}$ ($g \neq g'$) in the coefficient matrix \mathbf{A} and a number of additional auxiliary sub-vectors as evenly as possible (to ensure the load balancing and guarantee the parallel performance) onto different parallel processor cores within the communication sub-group COMM_ g ($g = 1, \dots, G, I, E$), i.e., compared with the other processor cores, at most one more row of nonzero entries are descended into a certain processor core. More concretely, the data structure used to manage the sub-matrix A_g is the parallel compressed sparse row (CSR) matrix storage format while the diagonal entries are preferentially stored at each row, that is, the ParCSRMatrix container in the open-source software package hypre (high performance preconditioners and solvers featuring multigrid) [14] from Lawrence Livermore National Laboratory, while the ParVector container in hypre is applied to tackle all of nonzero sub-vectors.

It is important to emphasize that the communication mechanism for message exchanges among these communication sub-groups is fairly simple and easy to implement: message exchanges only take place between two parallel processor cores with the same local identifier (i.e., ‘key’ value).

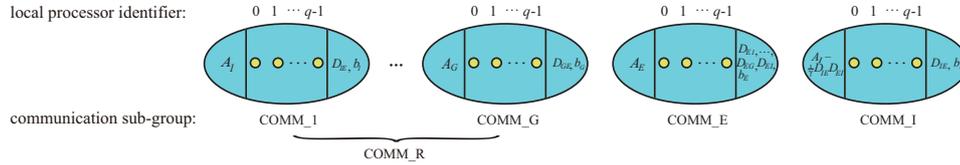


FIGURE 1. An abridged general view of the $G + 2$ communication sub-groups derived from a quick and easy disassembly of the global communicator containing $(G + 2)q$ parallel processor cores. Each of them is accompanied by the coefficient matrix (left), an MPI topology of q parallel processor cores (middle) and the correlative off-diagonal sub-blocks together with the right-hand side vector (right).

Remark 4. *The quasi-optimal positive parameters β^* and γ^* defined by (17) would have to be determined at the cost of the embarrassingly parallel computations of all diagonal entries of $G + 1$ ParCSRMatrix-by-ParCSRMatrix multiplications, i.e., $A_E D_{EI}^2 A_E$ and $A_g D_{gE}^2 A_g$ ($g = 1, \dots, G$), which is mathematically equivalent to one ParCSRMatrix-by-ParCSRMatrix multiplication within a certain communication sub-group. We note that the matrix-product should not be completely calculated, because only the diagonal (i.e., first) entries need to be generated, which can be accomplished via certain modifications to the subroutine ‘hypr-ParMatmul’ in hypr. Furthermore, the MPI functions ‘MPI_Reduce’ and ‘MPI_Bcast’ are also needed to complete the whole calculation.*

The proposed preconditioning algorithm is implemented on the top of the software library ‘jxpang’ [46]. The parallel calculation workflow and network topology structure on information communications of a single application of \mathbf{P} is depicted in Figure 2, whose complete particular description is as follows:

- parallel calculation step1: within the communication sub-group COMM- g ($g = 1, \dots, G$), numerically solve $A_g u_g = b_g$ to obtain u_g via the Boomer-AMG solver (a frequently-used and prestigious parallel implementation of classical algebraic multigrid in hypr) [19] with a specified maximum number of iterations n_g^{\max} and a prescribed relative tolerance δ_g for stopping of inner iterations.
- data transfer (a): send the real arrays u_g ($g = 1, \dots, G$) from COMM- g to COMM.E between processor cores of the same ‘key’ value.
- parallel calculation step2: within the communication subgroup COMM.E, receive the data packets, compute $\tilde{b}_E := b_E - \sum_{g=1}^G D_{Eg} u_g$ and, by invoking the BoomerAMG solver, determine the numerical solution u_E from $A_E u_E = \tilde{b}_E$, using the fixed n_E^{\max} and δ_E for the inner stopping criterion.
- data transfer (b): between two processor cores of the same ‘key’ value, send the real array u_E from COMM.E to COMM.I.
- parallel calculation step3: after the piece of data is received within the communication subgroup COMM.I, generate $u_I := b_I - D_{IE} u_E$ and seek for the approximate solution w_I from

$$(A_I - \frac{1}{\gamma} D_{IE} D_{EI}) w_I = u_I$$

by exploiting the BoomerAMG solver with n_I^{\max} and δ_I being prescribed.

- data backhaul (c): send the resulting real array w_I from COMM_I to COMM_E between processor cores of the same ‘key’ value.
- parallel calculation step4: within the communication subgroup COMM_E, compute $w_E := u_E - \frac{1}{\gamma} D_{EI} w_I$ when the piece of data has been received.
- data backhaul (d): from COMM_E to COMM_g ($g = 1, \dots, G$), send the resultant real array w_E between two processor cores of the same ‘key’ value.
- parallel calculation step5: after the piece of data is received, generate $w_g := u_g - \frac{1}{\beta} D_{gE} w_E$ within the communication sub-group COMM_g ($g = 1, \dots, G$).

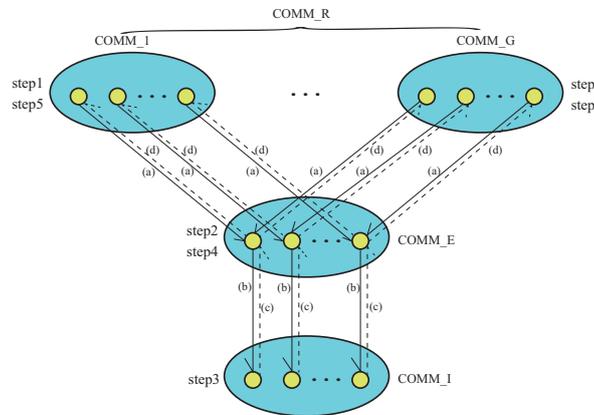


FIGURE 2. A graphic representation of the parallel calculation workflow and network topology structure on information communications of a single application of the APSS-SR preconditioner \mathbf{P} .

4. Numerical results and discussion

This section is devoted to investigating and discussing the numerical robustness, computational efficiency and parallel strong and weak scaling properties of the proposed APSS-SR preconditioner \mathbf{P} , compared with the monolithic Boomer-AMG preconditioner [19], the physical-variable based coarsening two-level (PCTL) preconditioner [22, 44, 55] and the relaxed APSS preconditioner $\tilde{\mathbf{P}}$ defined by (5), which are effectively implemented in hypre, jxpamg and jxpamg, respectively⁴.

4.1. Experimental setup. The experimental study and numerical comparison are carried out using three successively refined three-dimensional (adaptive) unstructured computational meshes (referred to as \mathcal{M}_0 represented in Figure 3, \mathcal{M}_1 and \mathcal{M}_2 , respectively, with 93,177, 745,416 and 5,963,328 grid cells) under the jaumin framework [29] and seven flux-limited MGD linear systems (symbolized by U26-2, U88-1, U42-3, U97-2, U39-3, U110-1 and U169-2, arising, respectively, from the discrete linear second-order reaction-diffusion equations in the 2nd, 1st, 3rd, 2nd, 3rd, 1st and 2nd nonlinear iterations at the 26th, 88th, 42nd, 97th, 39th, 110th and 169th time-levels of three different real-world sixty-four-group ($G = 64$) capsule implosion simulations). The initial and boundary value conditions for the nonlinear PDE system (1) are: the initial radiation, electron and ion temperatures

⁴These implementations can be made freely available upon reasonable request while the authors do not have permission to share data.

are all set to 3.0×10^{-4} while radiation energy densities are computed through the Plank interpolation formula and the initial radiation temperature; the zero-flow boundary condition is imposed at all physical boundaries of the electron and ion temperature variables together with the angle direction and spherical center of the radiation energy density variables, at whose outer radius are the different inflow boundary conditions.

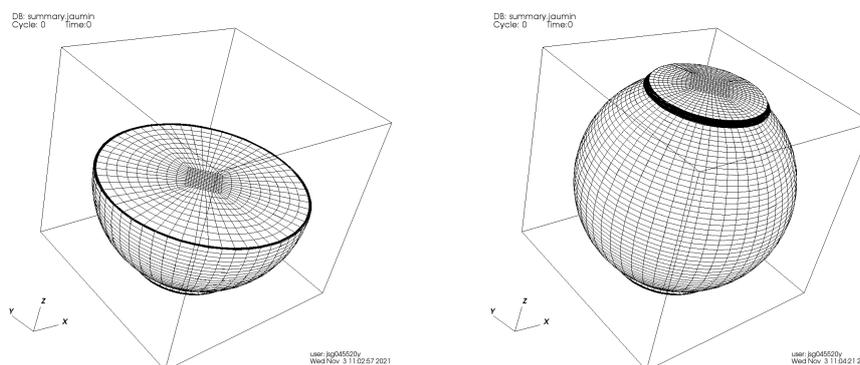


FIGURE 3. The half and full side views of the coarsest computational grid \mathcal{M}_0 with 93,177 mesh cells.

All computations of the before-mentioned challenging realistic unstructured capsule implosion test cases are performed on the Tianhe-2A supercomputer, which is deployed at China's National Supercomputer Center in Guangzhou and currently ranked 24th in the November 2024 TOP500 list. It exercises the Kylin Linux operating system and delivers 100.68 petaflops peak performance in theory and 61.44 petaflops Linpack performance with 17,792 computing nodes. Each of them is assembled with dual 12-core Intel Ivy Bridge Xeon E5-2692v2 central processing units (24 parallel processor cores in total), which are all clocked at 2.2 gigahertz and have 64.0 gigabytes of DDR3 main memory and the Matrix-2000 processors for performance acceleration. The proprietary high-performance TH Express-2 interconnect network topology is an opto-electronic hybrid and hierarchical fat tree. We make use of the Intel C compiler (icc) with Tianhe's self-optimized mpich-3.2 in an MPI-only mode (i.e., the 'configure' script is executed with '--with-MPI' and '--without-openmp' options so that the pure MPI implementation is under investigation), take the optimization flag '-O3 -mavx' and link all C codes to Intel MKL composer_xe_2015.1.133. All of the 24 MPI parallel processes in each computing node are utilized to run parallel C codes.

For solutions of all the sparse linear subsystems involved in the practical implementations of PCTL, $\tilde{\mathbf{P}}$ and \mathbf{P} , a single BoomerAMG V(1,1)-cycle (with 1 pre- and 1 post-smoothing steps) is applied, namely, $n_g^{\max} = 1$ and $\delta_g = 10^{-6}$ (which is generally unreachable) in the parallel operations step1, step2 and step3 for the index $g = 1, \dots, G, I, E$. Another important remark is that all applications of the BoomerAMG solver are invoked in its 'best practices' scenario [13], which is commonly recommended by the hypre developers and consists of the following components:

- in the SETUP phase, a strength-of-connection measure of 0.25, the hybrid modified independent set coarsening strategy (coarsen.type = 10), the aggressive coarsening scheme on the finest level (agg_num.levels = 1), the

‘extended+i’ distance-two interpolation strategy (`interp_type = 6`) followed by a truncation to no more than four nonzero entries per row (`P_max_elmts = 4`), the coarse-grid operator calculated algebraically via the Galerkin approach (i.e., the triple sparse matrix product where restriction is given by the transpose of interpolation) and the coarsening process terminated when the coarse-grid size is less than 100 (`max_coarse_size = 100`);

- in the CYCLE phase, the Gaussian elimination smoother (`grid_relax_type[3] = 9`) put into use at the coarsest grid level, while, at the other grid levels, one sweep of the hybrid ℓ_1 Gauss-Seidel smoothing in the ‘Coarse-Fine’ forward ordering on the down cycle (`relax_type = 13`) and in the ‘Fine-Coarse’ backward ordering on the up cycle (`relax_type = 14`).

Furthermore, in regard to the preconditioned FGMRES(m) solver, we make a choice of the restart parameter $m = 30$, make use of a relative tolerance of 10^{-8} between Euclidean norms of the current residual vector and the right-hand side vector (since the initial guess vector is set to zero) and the maximum number of outer iterations $iter_{\max} = 200$ as our stopping criterion, and report the realistic number of iteration steps $iter_{np}$ and the total elapsed time-to-solution $time_{np}^{tot}$ in seconds (averaged over 20 test runs to avoid disturbance and measured via the MPI function ‘MPI.Wtime’) required to converge while our MPI-parallel C codes are executed across np MPI parallel tasks. Besides, the parallel strong or weak efficiency $efcy_p^k$, respectively evaluated by $time_p^{tot} / (k \cdot time_{kp}^{tot})$ or $time_p^{tot} / time_{kp}^{tot}$ [38], is also provided.

4.2. Convergence and efficiency comparisons. Of particularly practical interest are the dependence of convergence factor on the computational mesh refinement (i.e., \mathcal{M}_0 and \mathcal{M}_1 , with 6,149,682 and 49,197,456 unknown quantities, respectively, because there are 66 degrees of freedom per grid cell) and the physical parameter variation (i.e., seven real-world flux-limited sixty-four-group linear systems in different physical parameter settings, e.g., the radiative free paths of potentially quite different scales) and the concrete computational efficiency on one processor core.

We tabulate in Table 1 the specific numerical performance, from which we can observe that

- the convergence of the FGMRES(30) solvers right-preconditioned by the relaxed APSS preconditioner $\tilde{\mathbf{P}}$ and the APSS-SR preconditioner \mathbf{P} is both achieved with almost unchanged iteration counts regardless of the problem size and numeration, i.e., an overall similar but good enough robustness and a very low convergence factor are obtained;
- the FGMRES(30) solver right-preconditioned by BoomerAMG does not iterate robustly in regard to the spatial mesh sizes and physical parameters and exhibits a low convergence rate for U88-1, U94-2, U110-1 and U169-2;
- there are two test cases (i.e., U97-2 and U110-1) which cannot be solvable via the FGMRES(30) solver right-preconditioned by PCTL, while it solves the other test cases with a moderately lower convergence rate than that of $\tilde{\mathbf{P}}$ and \mathbf{P} , however, also in a robust manner;
- the fastest reduction in relative residual norms (i.e., the best preconditioning behavior) is achieved by \mathbf{P} , which results in an average of 16.57 and 1.31 times faster than BoomerAMG and $\tilde{\mathbf{P}}$ on \mathcal{M}_0 , respectively, while, on the refined grid \mathcal{M}_1 , the corresponding speedup ratios are 24.03 and 1.26;
- for those five test problems which can be tackled by PCTL (i.e., U26-2, U88-1, U42-3, U39-3 and U169-2), the FGMRES(30) solver right-preconditioned

TABLE 1. The iteration counts and elapsed time-to-solutions of four types of right-preconditioned FGMRES(30) solvers applied to solve seven sixty-four-group linear systems over \mathcal{M}_0 (top) and \mathcal{M}_1 (bottom). Dashed entries (-) manifest the numerical solutions failed to converge after 200 iteration steps while values in parentheses signify the order of magnitude of the final relative residual norm.

	\mathcal{M}_0							
	BoomerAMG		PCTL		relaxed APSS		APSS-SR	
	$iter_1$	$time_1^{tot}$	$iter_1$	$time_1^{tot}$	$iter_1$	$time_1^{tot}$	$iter_1$	$time_1^{tot}$
U26-2	23	53.84	13	10.77	6	6.54	4	5.24
U88-1	62	107.33	15	11.61	9	8.25	6	6.37
U42-3	29	62.09	14	11.19	8	7.68	4	5.22
U97-2	81	133.38	- (10^{-4})	-	10	8.82	6	6.31
U39-3	27	59.37	14	11.23	7	7.11	4	5.28
U110-1	96	153.95	- (10^{-4})	-	9	8.23	7	6.85
U169-2	84	137.51	17	12.45	11	9.40	8	7.42

	\mathcal{M}_1							
	BoomerAMG		PCTL		relaxed APSS		APSS-SR	
	$iter_1$	$time_1^{tot}$	$iter_1$	$time_1^{tot}$	$iter_1$	$time_1^{tot}$	$iter_1$	$time_1^{tot}$
U26-2	35	697.72	14	107.09	7	57.03	4	44.52
U88-1	93	1383.11	18	119.92	10	68.44	6	51.69
U42-3	44	804.06	16	113.53	8	60.87	5	48.10
U97-2	106	1536.75	- (10^{-4})	-	9	64.62	6	51.64
U39-3	41	768.64	17	116.75	7	57.08	5	48.13
U110-1	129	1808.42	- (10^{-3})	-	10	68.49	7	55.27
U169-2	112	1607.58	19	123.11	12	76.05	8	58.81

by \mathbf{P} runs averagely 1.94 and 2.31 times faster than that right-preconditioned by PCTL on \mathcal{M}_0 and \mathcal{M}_1 , respectively.

4.3. Scalability tests on the Tianhe-2A supercomputer. Simulation time is a matter of considerable interest in thermal radiation transport applications. A parallel scalable preconditioned FGMRES(30) solver is able to cut down the simulation time by aggrandizing the number of parallel processor cores used in the simulation, which corresponds to the so-called strong scaling test. Without doubt, the weak scaling property is another important parallel performance measurement indicator of the preconditioned FGMRES(30) solver. Notice that the PCTL algorithm is not considered here because of its insensible convergence behavior when solving U97-2 and U110-1.

A strong scaling analysis is carried out to deal with the seven test problems on \mathcal{M}_1 using the FGMRES(30) solvers right-preconditioned by BoomerAMG, $\tilde{\mathbf{P}}$ and \mathbf{P} . The number of parallel processor cores used is altered from 132 to 1,056 (doubling every time), which means that 372,708, 186,354, 93,177 and 46589 or 46588 degrees of freedom are distributed on each physical processor core, respectively.

As shown in Table 2, we can deduce that the FGMRES(30) solvers right-preconditioned by BoomerAMG, $\tilde{\mathbf{P}}$ and \mathbf{P} , in the strong sense, all make clear their remarkable numerical and parallel scaling properties, i.e., their iteration counts and elapsed time-to-solutions required for convergence persist essentially unchanged and adequately shortened as for the increase on the number of parallel processor cores: their average strong parallel efficiencies are 93.3%, 86.5% and 76.3% for BoomerAMG, 84.2%, 79.2% and 72.6% for the relaxed APSS preconditioner $\tilde{\mathbf{P}}$ while 79.6%,

TABLE 2. A strong scalability investigation on the iteration counts, elapsed time-to-solutions and parallel efficiencies of three right-preconditioned FGMRES(30) solvers.

\mathcal{M}_1	BoomerAMG											
	np=132			np=264			np=528			np=1,056		
	iter ₁₃₂	time ₁₃₂ ^{tot}	iter ₂₆₄	time ₂₆₄ ^{tot}	efcy ₁₃₂ ²	iter ₅₂₈	time ₅₂₈ ^{tot}	efcy ₂₆₄ ²	iter ₁₀₅₆	time ₁₀₅₆ ^{tot}	efcy ₅₂₈ ²	
U26-2	39	78.57	41	43.81	89.7%	37	24.25	90.3%	46	17.57	69.0%	
U88-1	98	152.08	96	78.67	96.6%	102	46.54	84.5%	101	29.21	79.7%	
U42-3	51	93.52	49	48.86	95.7%	55	30.43	80.3%	57	19.89	76.5%	
U97-2	114	172.04	118	92.63	92.9%	113	50.31	92.0%	122	33.65	74.7%	
U39-3	48	89.79	53	51.42	87.3%	56	30.77	83.5%	54	19.26	79.9%	
U110-1	140	204.35	147	110.95	92.1%	144	60.92	91.1%	151	39.78	76.6%	
U169-2	117	175.71	112	88.79	98.9%	121	53.04	83.7%	125	34.24	77.4%	

\mathcal{M}_1	relaxed APSS											
	np=132			np=264			np=528			np=1,056		
	iter ₁₃₂	time ₁₃₂ ^{tot}	iter ₂₆₄	time ₂₆₄ ^{tot}	efcy ₁₃₂ ²	iter ₅₂₈	time ₅₂₈ ^{tot}	efcy ₂₆₄ ²	iter ₁₀₅₆	time ₁₀₅₆ ^{tot}	efcy ₅₂₈ ²	
U26-2	7	6.47	7	3.83	84.6%	7	2.38	80.2%	8	1.68	71.1%	
U88-1	11	8.19	11	4.72	86.9%	12	3.01	78.4%	12	1.97	75.6%	
U42-3	8	6.88	9	4.29	80.8%	9	2.65	81.1%	10	1.82	71.8%	
U97-2	9	7.32	9	4.26	85.8%	10	2.76	77.4%	12	2.01	69.3%	
U39-3	8	6.91	8	4.05	85.2%	9	2.62	76.8%	10	1.85	71.8%	
U110-1	10	7.76	11	4.69	82.3%	12	2.99	78.4%	12	1.99	75.6%	
U169-2	12	8.63	13	5.14	83.5%	13	3.14	82.3%	14	2.15	73.0%	

\mathcal{M}_1	APSS-SR											
	np=132			np=264			np=528			np=1,056		
	iter ₁₃₂	time ₁₃₂ ^{tot}	iter ₂₆₄	time ₂₆₄ ^{tot}	efcy ₁₃₂ ²	iter ₅₂₈	time ₅₂₈ ^{tot}	efcy ₂₆₄ ²	iter ₁₀₅₆	time ₁₀₅₆ ^{tot}	efcy ₅₂₈ ²	
U26-2	4	4.53	5	3.01	75.3%	5	1.92	78.3%	5	1.34	71.5%	
U88-1	6	5.36	6	3.24	82.8%	7	2.20	73.5%	7	1.52	71.3%	
U42-3	5	4.95	6	3.27	76.4%	8	2.34	69.1%	8	1.67	71.2%	
U97-2	7	5.78	7	3.46	83.4%	7	2.18	78.7%	8	1.63	67.0%	
U39-3	5	4.92	6	3.25	75.7%	6	2.06	78.5%	7	1.55	66.8%	
U110-1	7	5.81	7	3.43	84.7%	8	2.37	74.0%	9	1.76	67.3%	
U169-2	8	6.19	9	3.92	79.0%	9	2.48	78.9%	9	1.73	71.7%	

75.9% and 69.6% for the APSS-SR preconditioner \mathbf{P} , respectively. In addition, the FGMRES(30) solver right-preconditioned by \mathbf{P} achieves an average speedup of 17.3 and 1.2 over those FGMRES(30) solvers right-preconditioned by BoomerAMG and $\tilde{\mathbf{P}}$ when using 1,056 MPI ranks.

A weak scalability investigation is proceeded to compare BoomerAMG, $\tilde{\mathbf{P}}$ and \mathbf{P} by exploiting 66, 528 and 4,224 parallel processor cores (octupling every time) with 93,177 degrees of freedom per physical processor core. It signifies that the three particular cases $np = 66$, $np = 528$ and $np = 4,224$ are associated with the coarsest, intermediate and finest computational meshes \mathcal{M}_0 , \mathcal{M}_1 and \mathcal{M}_2 , respectively.

By inspecting the elapsed wall-clock time of three right-preconditioned FGMRES(30) solvers against the number of parallel processor cores in Table 3, it can be noticed that the average parallel efficiencies in this weak scaling examination for BoomerAMG, $\tilde{\mathbf{P}}$ and \mathbf{P} are 76.6%, 70.6% and 66.4%, respectively, when using 4,224 MPI ranks, in which case we further observe 16.4 and 1.2 times higher computational performance for \mathbf{P} than for BoomerAMG and $\tilde{\mathbf{P}}$. It is obvious that the two preconditioners $\tilde{\mathbf{P}}$ and \mathbf{P} both weakly scale well from the viewpoint of numerical and parallel scalabilities (with respect to the number of iterations and the elapsed wall-clock time, respectively) and have the same desirable weak scaling property.

5. Conclusions and perspectives

An adaptive backward Eulerian scheme and a cell-centered finite volume method are applied to discretize the three-dimensional flux-limited multi-group radiation diffusion equations in the temporal and spatial directions, respectively. We then take advantage of a parallel FGMRES(m) solver with the proposed selectively relaxed alternating positive semidefinite splitting preconditioner to accelerate the timeliness of the thermal radiation transport simulation. We also present its parallel

TABLE 3. The number of iterations, elapsed wall-clock time and parallel efficiencies of three right-preconditioned FGMRES(30) solvers in a weak scalability investigation.

	BoomerAMG								
	$np=66, \mathcal{M}_0$			$np=528, \mathcal{M}_1$			$np=4,224, \mathcal{M}_2$		
	$iter_{66}$	$time_{66}^{tot}$		$iter_{528}$	$time_{528}^{tot}$	$efcy_{66}^s$	$iter_{4,224}$	$time_{4,224}^{tot}$	$efcy_{528}^s$
U26-2	35	18.48		37	24.25	76.3%	51	35.60	68.1%
U88-1	94	40.21		102	46.54	86.4%	107	57.82	80.5%
U42-3	47	22.90		55	30.43	75.3%	62	40.01	76.1%
U97-2	109	45.73		113	50.31	90.9%	123	64.15	78.4%
U39-3	43	21.42		56	30.77	69.7%	64	40.83	75.3%
U110-1	132	54.19		144	60.92	89.0%	156	77.47	78.6%
U169-2	111	46.47		121	53.04	87.6%	130	67.12	79.0%

	relaxed APSS								
	$np=66, \mathcal{M}_0$			$np=528, \mathcal{M}_1$			$np=4,224, \mathcal{M}_2$		
	$iter_{66}$	$time_{66}^{tot}$		$iter_{528}$	$time_{528}^{tot}$	$efcy_{66}^s$	$iter_{4,224}$	$time_{4,224}^{tot}$	$efcy_{528}^s$
U26-2	7	1.92		7	2.38	80.8%	8	3.34	71.3%
U88-1	11	2.39		12	3.01	79.8%	14	4.43	68.1%
U42-3	9	2.14		9	2.65	81.5%	10	3.70	71.6%
U97-2	9	2.16		10	2.76	78.3%	10	3.72	74.3%
U39-3	8	2.04		9	2.62	77.9%	10	3.71	70.7%
U110-1	12	2.52		12	2.99	84.3%	13	4.27	70.0%
U169-2	12	2.55		13	3.14	81.2%	15	4.62	67.9%

	APSS-SR								
	$np=66, \mathcal{M}_0$			$np=528, \mathcal{M}_1$			$np=4,224, \mathcal{M}_2$		
	$iter_{66}$	$time_{66}^{tot}$		$iter_{528}$	$time_{528}^{tot}$	$efcy_{66}^s$	$iter_{4,224}$	$time_{4,224}^{tot}$	$efcy_{528}^s$
U26-2	5	1.49		5	1.92	77.5%	6	3.06	62.7%
U88-1	7	1.72		7	2.20	78.0%	7	3.24	68.0%
U42-3	7	1.71		8	2.34	73.2%	8	3.43	68.1%
U97-2	6	1.59		7	2.18	73.4%	8	3.45	63.5%
U39-3	6	1.61		6	2.06	77.7%	6	3.08	67.3%
U110-1	8	1.82		8	2.37	77.1%	9	3.59	66.1%
U169-2	8	1.83		9	2.48	73.8%	9	3.57	69.2%

performance on the Tianhe-2A supercomputer, and the results on seven real-world representative linear systems show that the proposed preconditioner achieves, averagely, 69.6% and 66.4% parallel strong and weak efficiency with 1,056 and 4,224 parallel processor cores, respectively. In line with current trends, for such kind of problem, it is quite likely that the future will see many parallel-in-time algorithms utilizing mixed-precision arithmetic operations [52] and mixed-precision preconditioning and solution algorithms [1, 20].

Acknowledgments

The authors are grateful to the anonymous referees for their careful readings of a preliminary version of our manuscript and their valuable suggestions and important comments, which greatly improved the quality of this paper. This work was supported by the 2023rd Open Fund of Tibet Yangbajing High Altitude Electrical Safety & Electromagnetic Environment National Observation and Research Station (China Electric Power Research Institute) with its title ‘‘Fast Finite Element Method for Three-Dimensional Total Electric Field of Complex Connecting Busbars in DC Yards of High-Altitude Converter Stations Based on Algebraic Multigrid’’ (Grant No. GYB51202301653).

References

- [1] A. Abdelfattah, H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, A. Fox, M. Gates, N. J. Higham, X. S. Li, J. Loe, P. Luszczek, S. Pranesh, S. Rajamanickam, T. Ribizel, B. F. Smith, K. Swirydowicz, S. Thomas, S. Tomov, Y. M. Tsai, U. M. Yang, A survey of numerical linear algebra methods utilizing mixed-precision arithmetic, *Int. J. High Perform. C.* 35 (2021) 344–369.
- [2] P. R. Amestoy, I. S. Duff, J.-Y. L’Excellent, J. Koster, MUMPS: a general purpose distributed memory sparse solver, *Lect. Notes Comput. Sci.* 1947 (2001) 121–130.
- [3] H. B. An, Z. Y. Mo, X. W. Xu, X. W. Jia, Operator-based preconditioning for the 2-D 3-T energy equations in radiation hydrodynamics simulations, *J. Comput. Phys.* 385 (2019) 51–74.
- [4] H. Aslani, D. K. Salkuyeh, Semi-convergence of the APSS method for a class of nonsymmetric three-by-three singular saddle point problems, arXiv: 2208.00814.
- [5] C. Baldwin, P. N. Brown, R. Falgout, F. Graziani, J. Jones, Iterative linear solvers in 2D radiation-hydrodynamics code: methods and performance, *J. Comput. Phys.* 154 (1999) 1–40.
- [6] Z.-Z. Bai, M. K. Ng, On inexact preconditioners for nonsymmetric matrices, *SIAM J. Sci. Comput.* 26 (2005) 1710–1724.
- [7] M. Benzi, G. H. Golub, J. Liesen, Numerical solution of saddle point problems, *Acta Numer.* 14 (2005) 1–137.
- [8] M. Benzi, M. Ng, Q. Niu, Z. Wang, A Relaxed Dimensional Factorization preconditioner for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 230 (2011) 6185–6202.
- [9] P. N. Brown, C. S. Woodward, Preconditioning strategies for fully implicit radiation diffusion with material-energy transfer, *SIAM J. Sci. Comput.* 23 (2001) 499–516.
- [10] F. Chen, B.-C. Ren, A modified alternating positive semidefinite splitting preconditioner for block three-by-three saddle point problems, *Electron. Trans. Numer. Anal.* 58 (2023) 84–100.
- [11] A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, G. N. Miranda, J. W. Ruge, Robustness and scalability of algebraic multigrid, *SIAM J. Sci. Comput.* 21 (2000) 1886–1908.
- [12] T. A. Davis, Algorithm 832: UMFPACK—an unsymmetric-pattern multifrontal method with a column reordering strategy, *ACM Trans. Math. Softw.* 30 (2004) 196–199.
- [13] R. D. Falgout, J. B. Schroder, Non-Galerkin coarse grids for algebraic multigrid, *SIAM J. Sci. Comput.* 36 (2014) C309–C334.
- [14] R. D. Falgout, U. M. Yang, hypre: a library of high performance preconditioners, *Lect. Notes Comput. Sci.* 2331 (2002) 632–641.
- [15] Z. F. Fan, S. P. Zhu, W. B. Pei, W. H. Ye, M. Li, X. W. Xu, J. F. Wu, Z. S. Dai, L. F. Wang, Numerical investigation on the stabilization of the deceleration phase Rayleigh-Taylor instability due to alpha particle heating in ignition target, *EPL* 99 (2012) 65003.
- [16] R. Glowinski, J. Toivanen, A multigrid preconditioner and automatic differentiation for non-equilibrium radiation diffusion problems, *J. Comput. Phys.* 207 (2005) 354–374.
- [17] L. Grigori, Q. Niu, Y. X. Xu, Stabilized dimensional factorization preconditioner for solving incompressible Navier-Stokes equations, *Appl. Numer. Math.* 146 (2019) 309–327.
- [18] P. Hénon, P. Ramet, J. Roman, PaStiX: a high-performance parallel direct solver for sparse symmetric definite systems, *Parallel Comput.* 28 (2002) 301–321.
- [19] V. E. Henson, U. M. Yang, BoomerAMG: a parallel algebraic multigrid solver and preconditioner, *Appl. Numer. Math.* 41 (2002) 155–177.
- [20] N. J. Higham, T. Mary, Mixed precision algorithms in numerical linear algebra, *Acta Numer.* 31 (2022) 347–414.
- [21] Q. Y. Hu, L. Zhao, Domain decomposition preconditioners for the system generated by discontinuous Galerkin discretization of 2D-3T heat conduction equations, *Commun. Comput. Phys.* 22 (2017) 1069–1100.
- [22] S. L. Huang, X. Q. Yue, X. W. Xu, α Setup-PCTL: an adaptive setup-based two-level preconditioner for sequence of linear systems of three-temperature energy equations, *Commun. Comput. Phys.* 32 (2022) 1287–1309.
- [23] Y.-M. Huang, A practical formula for computing optimal parameters in the HSS iteration methods, *J. Comput. Appl. Math.* 255 (2014) 142–149.
- [24] L. M. Kačanov, Variational methods of solution of plasticity problems, *J. Appl. Math. Mech.* 23 (1959) 880–883.
- [25] Y. F. Ke, C. F. Ma, Modified alternating positive semidefinite splitting preconditioner for time-harmonic eddy current models, *J. Comput. Math.* 39 (2021) 733–754.

- [26] Y. F. Ke, C. F. Ma, Z. R. Ren, A new alternating positive semidefinite splitting preconditioner for saddle point problems from time-harmonic eddy current models, *Front. Math. China* 13 (2018) 313–340.
- [27] X. S. Li, An overview of SuperLU: algorithms, implementation, and user interface, *ACM Trans. Math. Softw.* 31 (2005) 302–325.
- [28] Z.-Z. Liang, G.-F. Zhang, Alternating positive semidefinite splitting preconditioners for double saddle point problems, *Calcolo* 56 (2019) 26.
- [29] Q. K. Liu, Z. Y. Mo, A. Q. Zhang, Z. Yang, JAUMIN: a programming framework for large-scale numerical simulation on unstructured meshes, *CCF Trans. HPC* 1 (2019) 35–48.
- [30] L. S. Meng, J. Li, S.-X. Miao, A variant of relaxed alternating positive semi-definite splitting preconditioner for double saddle point problems, *Jpn. J. Ind. Appl. Math.* 38 (2021) 979–998.
- [31] Z. Y. Mo, A. Q. Zhang, X. L. Cao, Q. K. Liu, X. W. Xu, H. B. An, W. B. Pei, S. P. Zhu, JASMIN: a parallel software infrastructure for scientific computing, *Front. Comput. Sci.* 4 (2010) 480–488.
- [32] V. A. Mousseau, D. A. Knoll, New physics-based preconditioning of implicit methods for non-equilibrium radiation diffusion, *J. Comput. Phys.* 190 (2003) 42–51.
- [33] H. M. Peng, Radiation transport and radiation hydrodynamics in plasmas, Beijing: National Defense Industry Press, 2008.
- [34] B.-C. Ren, F. Chen, X.-L. Wang, Improved splitting preconditioner for double saddle point problems arising from liquid crystal director modeling, *Numer. Algorithms* 91 (2022) 1363–1379.
- [35] Z.-R. Ren, Y. Cao, An alternating positive-semidefinite splitting preconditioner for saddle point problems from time-harmonic eddy current models, *IMA J. Numer. Anal.* 36 (2016) 922–946.
- [36] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, *SIAM J. Sci. Comput.* 14 (1993) 461–469.
- [37] Y. Saad, M. H. Schultz, GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems, *SIAM J. Sci. Stat. Comput.* 7 (1986) 856–869.
- [38] S. Sahni, V. Thanvantri, Performance metrics: keeping the focus on runtime, *IEEE Parall. Distrib.* 4 (1996) 43–56.
- [39] D. K. Salkuyeh, H. Aslani, Z.-Z. Liang, An alternating positive semidefinite splitting preconditioner for the three-by-three block saddle point problems, *Math. Commun.* 26 (2021) 177–195.
- [40] O. Schenk, K. Gärtner, W. Fichtner, A. Stricker, PARDISO: a high-performance serial and parallel sparse linear solver in semiconductor device simulation, *Future Gener. Comput. Syst.* 18 (2001) 69–78.
- [41] S. Shu, M. H. Liu, X. W. Xu, X. Q. Yue, S. G. Li, Algebraic multigrid block triangular preconditioning for multidimensional three-temperature radiation diffusion equations, *Adv. Appl. Math. Mech.* 13 (2021) 1203–1226.
- [42] A. J. Wathen, Preconditioning, *Acta Numer.* 24 (2015) 329–376.
- [43] X. W. Xu, Z. Y. Mo, Algebraic interface-based coarsening AMG preconditioner for multiscale sparse matrices with applications to radiation hydrodynamics computation, *Numer. Linear Algebra Appl.* 24 (2017) e2078.
- [44] X. W. Xu, Z. Y. Mo, H. B. An, Algebraic two-level iterative method for 2-D 3-T radiation diffusion equations, *Chinese J. Comput. Phys.* 26 (2009) 1–8.
- [45] X. W. Xu, Z. Y. Mo, X. Q. Yue, H. B. An, S. Shu, α Setup-AMG: an adaptive-setup-based parallel AMG solver for sequence of sparse linear systems, *CCF Trans. HPC* 2 (2020) 98–110.
- [46] X. W. Xu, X. Q. Yue, R. Z. Mao, Y. T. Deng, S. L. Huang, H. F. Zou, X. Liu, S. L. Hu, C. S. Feng, S. Shu, Z. Y. Mo, JXPAMG: a parallel algebraic multigrid solver for extreme-scale numerical simulations, *CCF Trans. HPC* 5 (2023) 72–83.
- [47] X. Q. Yue, J. M. He, X. W. Xu, S. Shu, L. B. Wang, Two physics-based Schwarz preconditioners for three-temperature radiation diffusion equations in high dimensions, *Commun. Comput. Phys.* 32 (2022) 829–849.
- [48] X. Q. Yue, J. B. Hou, L. Yuan, X. W. Xu, S. Shu, A physics-wise splitting preconditioner with selective relaxation for the multi-group radiation diffusion equations in three dimensions, *Commun. Comput. Phys.* (accepted).
- [49] X. Q. Yue, S. Shu, J. X. Wang, Z. Y. Zhou, Substructuring preconditioners with a simple coarse space for 2-D 3-T radiation diffusion equations, *Commun. Comput. Phys.* 23 (2018) 540–560.

- [50] X. Q. Yue, S. Shu, X. W. Xu, Z. Y. Zhou, An adaptive combined preconditioner with applications in radiation diffusion equations, *Commun. Comput. Phys.* 18 (2015) 1313–1335.
- [51] X. Q. Yue, C. Q. Wang, X. W. Xu, L. B. Wang, S. Shu, A new relaxed splitting preconditioner for the multidimensional multi-group radiation diffusion equations, *East Asian J. Appl. Math.* 12 (2022) 163–184.
- [52] X. Q. Yue, Z. Y. Wang, S.-L. Wu, Convergence analysis of a mixed precision Parareal algorithm, *SIAM J. Sci. Comput.* 45 (2023) A2483–A2510.
- [53] X. Q. Yue, S. Xia, C. Y. Chen, X. W. Xu, S. Shu, A generalized selectively relaxed matrix splitting preconditioning strategy for the three-dimensional flux-limited multi-group radiation diffusion equations, *Numer. Math. Theor. Meth. Appl.* 17 (2024) 630–657.
- [54] X. Q. Yue, C. X. Zhang, C. Y. Chen, X. W. Xu, S. Shu, A selectively relaxed splitting preconditioning strategy for the flux-limited multi-group radiation diffusion equations in three dimensions, *East Asian J. Appl. Math.* (accepted).
- [55] X. Q. Yue, S. L. Zhang, X. W. Xu, S. Shu, W. D. Shi, Algebraic multigrid block preconditioning for multi-group radiation diffusion equations, *Commun. Comput. Phys.* 29 (2021) 831–852.
- [56] X. Q. Yue, Z. K. Zhang, X. W. Xu, S. Y. Zhai, S. Shu, An algebraic multigrid-based physical factorization preconditioner for the multi-group radiation diffusion equations in three dimensions, *Numer. Math. Theor. Meth. Appl.* 16 (2023) 410–432.

Tibet Yangbajing High Altitude Electrical Safety & Electromagnetic Environment National Observation and Research Station (China Electric Power Research Institute), Beijing 100192, China; National Center for Applied Mathematics in Hunan, Key Laboratory of Intelligent Computing & Information Processing of Ministry of Education, Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan University, Xiangtan 411105, China

E-mail: yuexq@xtu.edu.cn

National Center for Applied Mathematics in Hunan, Key Laboratory of Intelligent Computing & Information Processing of Ministry of Education, Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan University, Xiangtan 411105, China

E-mail: 202321511280@smail.xtu.edu.cn and 914668645@qq.com

Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing 100094, China

E-mail: xwxu@iapcm.ac.cn

National Center for Applied Mathematics in Hunan, Key Laboratory of Intelligent Computing & Information Processing of Ministry of Education, Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan University, Xiangtan 411105, China

E-mail: shushi@xtu.edu.cn