East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.310121.230821

Vol. **12**, No. 1, pp. 163-184
February 2022

A New Relaxed Splitting Preconditioner for Multidimensional Multi-Group Radiation Diffusion Equations

Xiaoqiang Yue[1], Chunqing Wang[1], Xiaowen Xu[2], Libo Wang[3] and Shi Shu[1,*]

[1] *Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Key Laboratory of Intelligent Computing & Information Processing of Ministry of Education, School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, China.*
[2] *Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing 100094, China.*
[3] *College of Information Science and Technology, Jinan University, Guangzhou 510632, China.*

Received 31 January 2021; Accepted (in revised version) 23 August 2021.

Abstract. Motivated by the ideas of Frigo *et al.* [SIAM J. Sci. Comput. 41 (2019) B694–B720], we develop a novel relaxed splitting preconditioner and consider its parallel implementation. Fully-coupled fully-implicit linearised algebraic systems arising from the multidimensional multi-group radiation diffusion equations are solved by using algebraic multigrid subsolvers. Spectral properties of the relaxed splitting right-preconditioned matrix are studied. This allows to introduce an easily implementable algebraic selection strategy for finding the corresponding relaxation parameter. Numerical experiments show that the new preconditioner outperforms some existing popular preconditioners in robustness and efficiency and is well scalable both algorithmically and in parallel.

AMS subject classifications: 65F10, 65N55, 65Y05, 65Z05

Key words: Radiation diffusion equations, relaxed splitting, algebraic multigrid, incomplete LU factorization, parallel computing.

1. Introduction

The radiation transport processes describe the transmission, scattering and interaction of photons in vacuum or multicomponent background mediums. They are used in various coupled multiphysics applications such as astrophysical phenomena, biomedicine, and

[*] Corresponding author. *Email addresses:* yuexq@xtu.edu.cn (X. Q. Yue), xwxu@iapcm.ac.cn (X. W. Xu), shushi@xtu.edu.cn (S. Shu)

laser indirect-drive inertial confinement fusions (ICFs). The simulations of the radiation transport are very difficult because of a highly nonlinear behavior in optically thick mediums. In such circumstances, the radiation transport is modeled by a flux-limited time-dependent highly nonlinear and discontinuous multi-group radiation diffusion (MGD) approximation [4, 23]. The corresponding analysis is based on the efficient solution of fully-coupled sparse systems of linear algebraic equations with the degree of freedom (DoF) ranging from 10^7 to 10^{11}. Implicit time integration approaches are desired to overcome time-step constraints. In order to evaluate the nonlinear parts at the previous nonlinear iteration level, Kačanov [29] employed the method of frozen coefficients. An important observation is that because of the complex volatile nonlinear coupling of various physical quantities from numerous interacting spatial and temporal scales, the MGD equations are often discretised by finite volume schemes [14,24,38,45,47,52]. This leads to the series of nonsymmetric but positive definite linear systems, which have to be solved at each time-step and/or nonlinear iterations. We note that the finding of the corresponding numerical solutions is time-consuming, mainly because the coefficient matrices are highly ill-conditioned and their conditioning deteriorates as the mesh-size gets smaller. It takes more than 80% of the entire ICF simulation time in general. We note that the memory requirements of direct linear solvers severely limit the number of frequency groups and spatial mesh-size that can be tackled [3,16,30]. In order to deal with this challenging task and to exploit the ever-increasing computing power, robust and scalable iterative linear solvers must be combined with the preconditioners requiring minimal user input. The state-of-the-art iterative methods are the Krylov subspace solvers — cf. [26, 34, 42, 48]. The convergence of such methods is usually based on the conditioning of an associated matrix and on the clustering of its eigenvalues.

Over the past two decades, a variety of block preconditioners (also called physics-based preconditioners) have been developed in coupled multiphysics PDE applications, where system matrices have an underlying block structure. Block preconditioners are usually used in order to split a problem into a series of subproblems (which can be easily solved) and to establish an object-oriented framework incorporating off-the-shelf single-physics solvers and preconditioners. Thus substantial efforts have been spent on physics-based preconditioning strategies with algebraic multigrid (AMG) as an essential ingredient. This approach turned out to be successful in various coupled multiphysics PDE simulations — e.g. in poroelasticity [1], geophysical electromagnetics [8], Cahn-Hilliard Navier-Stokes systems [9], multiphase poromechanics of heterogeneous media [11], linear elasticity in mixed form [13], incompressible (reduced) resistive magnetohydrodynamic [15], incompressible flow simulations [21], elliptic optimal control problems [22], fluid-structure interaction problems in hemodynamics [32], models of coupled magma/mantle dynamics [39] and incompressible Navier-Stokes problems [44]. For fully implicit discretisations of multidimensional radiation diffusion equations, An *et al.* [2], Feng *et al.* [18], and Mousseau *et al.* [33] proposed various operator-based preconditioners in the Jacobian-free Newton-Krylov framework. Brown and Woodward [10] considered the Schur complement preconditioned generalised minimum residual (GMRES) solver without restarting based on higher-order time integration. Shu *et al.* [46] constructed theoretical and practical lower and upper block triangular preconditioners and bounded the eigenvalues of the corresponding preconditioned matrices. Xu *et al.* [50] developed a physical-variable based coarsening two-level preconditioner, which was improved and generalised [51, 53]. Besides, [51] presents two more Schur complement preconditioners regarded as sparse approximate inverses and combined with adaptive strategies based on the inter-block coupling strength and intra-block diagonal dominance. It is worth noting that the last five types of block preconditioners are constructed in a fully algebraic framework, which uses only the coefficient matrix and physical quantities needed to extract subsystems related to the serial/parallel implementation.

This work is aimed at the construction of a scalable and efficient block preconditioning algorithm for MGD linear systems. It is inspired by the sequential relaxed physical factorisation (RPF), which does not depend on accurate sparse approximations of Schur complements and its convergence is accelerated by a nearly-optimal relaxation parameter — cf. [19]. The remainder of the manuscript is structured as follows. In Section 2, we introduce the MGD model equations and the linear system arising from the spatiotemporal discretisation. We establish a new formula for the inverse of the corresponding coefficient matrix. It accelerates the convergence of the GMRES algorithm with a restart after m iterations. Section 3 describes the implementation details of the two-level parallelization, the eigenspectrum of the preconditioned matrix and an algebraic selection strategy on the relaxation parameter. In Section 4, we discuss a lower block triangular preconditioner for comparison purposes. The results of numerical experiments concerning weak and strong scalability properties in a moderate number of CPU cores are presented in Section 5. Concluding remarks are given in Section 6.

2. MGD Model Equations and Spatiotemporal Discretisation

We consider the following time-dependent MGD equations modeling energy exchange between photons of different frequencies, electron and ion over a spherical symmetrical geometry $\Omega \subset \mathbb{R}^d$ for $d = 2$ or $d = 3$:

$$\frac{\partial E_g}{\partial t} = \nabla \cdot \big(D_g(E_g)\nabla E_g\big) + c\big(\sigma_{Bg}B_g(T_E) - \sigma_{Pg}E_g\big) + S_g, \quad g = 1, \ldots, G, \tag{2.1}$$

$$\rho c_E \frac{\partial T_E}{\partial t} = \nabla \cdot \big(D_E(T_E)\nabla T_E\big) - c\sum_{g=1}^{G}\big(\sigma_{Bg}B_g(T_E) - \sigma_{Pg}E_g\big) + \omega_{IE}(T_I - T_E), \tag{2.2}$$

$$\rho c_I \frac{\partial T_I}{\partial t} = \nabla \cdot \big(D_I(T_I)\nabla T_I\big) - \omega_{IE}(T_I - T_E), \tag{2.3}$$

where G is the number of energy groups, c the velocity of light, ρ the density of the medium updated in the hydrodynamics process, ω_{IE} the energy transfer coefficient between ion and electron, E_g, $g = 1, \ldots, G$ the g-th radiation energy density, T_E and T_I are respectively the electron and ion temperatures [37]. Furthermore, $D_g(E_g)$ and $B_g(T_E)$ in (2.1) denote respectively the g-th nonlinear radiation diffusion coefficient and the electron scattering energy density, σ_{Bg} and σ_{Pg} the scattering and absorption coefficients of the Planck-averaged electron energy, and S_g is the radiation source item. Besides, in the Eqs. (2.2) and (2.3), c_α

and $D_\alpha(T_\alpha)$ respectively refer to specific heat capacities and nonlinear thermal-conductivity coefficients of electron if $\alpha = E$ and ion if $\alpha = I$.

The PDE system (2.1)-(2.3) is discretised by the (adaptive) backward Eulerian scheme in time, linearised via the method of frozen coefficients to yield a series of steady linear PDE systems of the form

$$
\begin{aligned}
&-\nabla\cdot\left(D_g^{(\delta)}\nabla E_g\right)+\left(\frac{1}{\Delta t_{n+1}}+c\sigma_{Pg}\right)E_g-\sigma_{Bg}\left(\frac{\partial B_g}{\partial T_E}\right)^{(\delta)}T_E\\
=&\frac{1}{\Delta t_{n+1}}E_g^{(n)}+\sigma_{Bg}\left[B_g^{(\delta)}-\left(\frac{\partial B_g}{\partial T_E}\right)^{(\delta)}T_E^{(\delta)}\right],\quad g=1,\dots,G,\\
&-\nabla\cdot\left(D_E^{(\delta)}\nabla T_E\right)+\left[\frac{\rho c_E^{(\delta)}}{\Delta t_{n+1}}+\omega_{IE}^{(\delta)}+\sum_{g=1}^{G}\sigma_{Bg}\left(\frac{\partial B_g}{\partial T_E}\right)^{(\delta)}\right]T_E-\sum_{g=1}^{G}c\sigma_{Pg}E_g-\omega_{IE}^{(\delta)}T_I\\
=&\frac{\rho c_E^{(\delta)}}{\Delta t_{n+1}}T_E^{(n)}-\sum_{g=1}^{G}\sigma_{Bg}\left[B_g^{(\delta)}-\left(\frac{\partial B_g}{\partial T_E}\right)^{(\delta)}T_E^{(\delta)}\right],\\
&-\nabla\cdot\left(D_I^{(\delta)}\nabla T_I\right)+\left(\frac{\rho c_I^{(\delta)}}{\Delta t_{n+1}}+\omega_{IE}^{(\delta)}\right)T_I-\omega_{IE}^{(\delta)}T_E=\frac{\rho c_I^{(\delta)}}{\Delta t_{n+1}}T_I^{(n)}
\end{aligned}
$$

and studied through the cell-centered positivity-preserving finite volume discretisation in space. Note that Δt_{n+1} is the current time-step size and the terms with the superscripts (n) and (δ) represent the previous time-level and recent approximations, respectively. Each unknown or degree of freedom describes a physical quantity at a mesh point.

Let n be the number of control volumes of the associated dual mesh. Grouping together the unknowns related to the same physical quantities, one can write the resulting discrete linearised algebraic systems of size $N = n(G+2)$ in a sparse block structured form — viz.

$$
\underbrace{\begin{bmatrix}
A_1 & & & D_{1E} & \\
& \ddots & & \vdots & \\
& & A_G & D_{GE} & \\
D_{E1} & \cdots & D_{EG} & A_E & D_{EI}\\
& & & D_{IE} & A_I
\end{bmatrix}}_{:=\mathbf{A}\in\mathbb{R}^{N\times N}}
\underbrace{\begin{bmatrix} E_1\\ \vdots\\ E_G\\ T_E\\ T_I\end{bmatrix}}_{:=\mathbf{u}\in\mathbb{R}^N}
=
\underbrace{\begin{bmatrix} f_1\\ \vdots\\ f_G\\ f_E\\ f_I\end{bmatrix}}_{:=\mathbf{f}\in\mathbb{R}^N},
\tag{2.4}
$$

where the diagonal sub-blocks A_g, $g=1,\dots,G,E,I$ of size n have the same nonzero structure of the discrete elliptic problem, while all variable cross-coupling items $D_{gg'}$, $g\neq g'$ of size n are diagonal matrices such that

$$
D_{EI}=D_{IE}\quad\text{and}\quad D_{Eg}\neq D_{gE}\quad\text{for}\quad g=1,\dots,G.
$$

This renders $\mathbf{A}$ generally positive definite but necessarily nonsymmetric. In addition, various parameters of different scales are involved in $\mathbf{A}$, e.g. the time and space step-sizes, radiative free path and characteristics of the background medium, which would noticeably affect the conditioning and eigenvalue distribution of $\mathbf{A}$. The inherent coupling also poses serious challenges regarding robustness in allusion to the crucial system parameters. Hence, the practical approach is to apply a GMRES(m) algorithm paired with suitable block preconditioning schemes. In this work, we use the right preconditioning and reformulate (2.4) as

$$(\mathbf{A}\mathbf{M}^{-1})\mathbf{z} = \mathbf{f} \quad \text{and} \quad \mathbf{M}^{-1}\mathbf{z} = \mathbf{u},$$

where $\mathbf{M}$ is an easily invertible preconditioning matrix, which approximates $\mathbf{A}$ and ensures a good convergence of the iteration procedure.

3. A New Relaxed Splitting Preconditioner

The RPF block preconditioning algorithm is introduced in [19] for mixed finite element discretisations of coupled poromechanics. It was inspired by the relaxed dimensional factorisation in [6] with the algebraic parameter estimation of [5]. A similar idea allows us to extend the original formulation to coefficient matrices in a different nonzero pattern and the number of blocks with two types of inherent physical-variable based partitioning — viz.

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} A_1 & & & D_{1E} & \\ & \ddots & & \vdots & \\ & & A_G & D_{GE} & \\ D_{E1} & \cdots & D_{EG} & A_E & \\ & & & & O \end{bmatrix} + \begin{bmatrix} O & & & & \\ & \ddots & & & \\ & & O & & \\ & & & O & D_{EI} \\ & & & D_{IE} & A_I \end{bmatrix} \\
&= \mathbf{A}_1^M + \mathbf{A}_1^F = \big(\mathbf{A}_1^M + \beta_1 \mathbf{I}_\pi\big) - \big(\beta_1 \mathbf{I}_\pi - \mathbf{A}_1^F\big)
\end{aligned} \tag{3.1}$$

and

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} A_1 & & & D_{1E} & \\ & \ddots & & \vdots & \\ & & A_G & D_{GE} & \\ D_{E1} & \cdots & D_{EG} & O & \\ & & & & O \end{bmatrix} + \begin{bmatrix} O & & & & \\ & \ddots & & & \\ & & O & & \\ & & & A_E & D_{EI} \\ & & & D_{IE} & A_I \end{bmatrix} \\
&= \mathbf{A}_2^F + \mathbf{A}_2^M = \big(\mathbf{A}_2^M + \beta_2 \mathbf{I}_\pi\big) - \big(\beta_2 \mathbf{I}_\pi - \mathbf{A}_2^F\big)
\end{aligned} \tag{3.2}$$

with real relaxation parameters $\beta_1, \beta_2 > 0$, the identity $N \times N$ matrix $\mathbf{I}_\pi$, and the zero $n \times n$ matrix O. The representations (3.1) and (3.2) lead to the physical alternating-type iterative procedure

$$\begin{aligned}
\big(\mathbf{A}_1^M + \beta_1 \mathbf{I}_\pi\big)\mathbf{u}^{(k+\frac{1}{2})} &= \big(\beta_1 \mathbf{I}_\pi - \mathbf{A}_1^F\big)\mathbf{u}^{(k)} + \mathbf{f}, \\
\big(\mathbf{A}_2^M + \beta_2 \mathbf{I}_\pi\big)\mathbf{u}^{(k+1)} &= \big(\beta_2 \mathbf{I}_\pi - \mathbf{A}_2^F\big)\mathbf{u}^{(k+\frac{1}{2})} + \mathbf{f}
\end{aligned}$$

for solving the Eq. (2.4) with iteration count $k = 0, 1, \ldots$ and with an arbitrary vector $\mathbf{u}^{(0)}$ of size N. Eliminating $\mathbf{u}^{(k+1/2)}$ from the above two expressions results in the stationary scheme

$$\mathbf{u}^{(k+1)} = \left(\mathbf{A}_2^M + \beta_2 \mathbf{I}_\pi\right)^{-1} \left(\beta_2 \mathbf{I}_\pi - \mathbf{A}_2^F\right) \left(\mathbf{A}_1^M + \beta_1 \mathbf{I}_\pi\right)^{-1} \left(\beta_1 \mathbf{I}_\pi - \mathbf{A}_1^F\right) \mathbf{u}^{(k)} + \tilde{\mathbf{c}} \tag{3.3}$$

with the iteration matrix

$$\mathbf{G}_\beta = \left(\mathbf{A}_2^M + \beta_2 \mathbf{I}_\pi\right)^{-1} \left(\beta_2 \mathbf{I}_\pi - \mathbf{A}_2^F\right) \left(\mathbf{A}_1^M + \beta_1 \mathbf{I}_\pi\right)^{-1} \left(\beta_1 \mathbf{I}_\pi - \mathbf{A}_1^F\right)$$

and a vector $\tilde{\mathbf{c}}$. Benzi and Szyld [7] showed that there exists a unique splitting $\mathbf{A} = \mathbf{P}_\beta - (\mathbf{P}_\beta - \mathbf{A})$ with a nonsingular $\mathbf{P}_\beta$ such that $\mathbf{G}_\beta$ in (3.3) can be written in the form $\mathbf{G}_\beta = \mathbf{I}_\pi - \mathbf{P}_\beta^{-1}\mathbf{A}$ and $\tilde{\mathbf{c}} = \mathbf{P}_\beta^{-1}\mathbf{f}$. Straightforward calculations yield

$$\mathbf{P}_\beta = \left(\mathbf{A}_1^M + \beta_1 \mathbf{I}_\pi\right)\left[(\beta_1 + \beta_2)\mathbf{I}_\pi - \mathbf{A}_2^F + \mathbf{A}_1^M\right]^{-1} \left(\mathbf{A}_2^M + \beta_2 \mathbf{I}_\pi\right).$$

We note that the factor $\beta_1 + \beta_2$ has to be replaced by β_2. Besides, for practical implementation and the analysis purposes, the factors β_1 and β_2 are chosen to equal a real relaxation parameter β. It follows from the definitions of $\mathbf{A}_1^M$, $\mathbf{A}_1^F$, $\mathbf{A}_2^F$ and $\mathbf{A}_2^M$ that

$$\begin{aligned}\mathbf{P}_\beta = & \begin{bmatrix} A_1 + \beta I_\pi & & & D_{1E} & \\ & \ddots & & \vdots & \\ & & A_G + \beta I_\pi & D_{GE} & \\ D_{E1} & \cdots & D_{EG} & A_E + \beta I_\pi & \\ & & & & \beta I_\pi \end{bmatrix} \\ & \times \begin{bmatrix} I_\pi & & & & \\ & \ddots & & & \\ & & I_\pi & & \\ & & & I_\pi & (A_E + \beta I_\pi)^{-1} D_{EI} \\ & & & \beta^{-1} D_{IE} & \beta^{-1}(A_I + \beta I_\pi) \end{bmatrix}. \end{aligned} \tag{3.4}$$

Comparing it with (2.4) yields

$$\mathbf{P}_\beta - \mathbf{A} = \begin{bmatrix} \beta I_\pi & & & & D_{1E}(A_E + \beta I_\pi)^{-1} D_{EI} \\ & \ddots & & & \vdots \\ & & \beta I_\pi & & D_{GE}(A_E + \beta I_\pi)^{-1} D_{EI} \\ & & & \beta I_\pi & O \\ & & & & \beta I_\pi \end{bmatrix} \tag{3.5}$$

with the $n \times n$ identity matrix I_π. This result can be substantially improved if the representation (3.4) is replaced by

$$\begin{aligned}
\mathbf{Q}_\beta &= \begin{bmatrix} A_1 & & & D_{1E} & \\ & \ddots & & \vdots & \\ & & A_G & D_{GE} & \\ D_{E1} & \cdots & D_{EG} & A_E+\beta I_\pi & \\ & & & & I_\pi \end{bmatrix} \begin{bmatrix} I_\pi & & & & \\ & \ddots & & & \\ & & I_\pi & & \\ & & & I_\pi & (A_E+\beta I_\pi)^{-1}D_{EI} \\ & & & D_{IE} & A_I \end{bmatrix} \\
&= \mathbf{Q}_1\mathbf{Q}_2.
\end{aligned} \tag{3.6}$$

The latter is actually our relaxed splitting preconditioning matrix. The respective difference $\mathbf{R}_\beta$ can be now written as

$$\mathbf{R}_\beta = \mathbf{Q}_\beta - \mathbf{A} = \begin{bmatrix} O & & & & D_{1E}(A_E+\beta I_\pi)^{-1}D_{EI} \\ & \ddots & & & \vdots \\ & & O & & D_{GE}(A_E+\beta I_\pi)^{-1}D_{EI} \\ & & & \beta I_\pi & O \\ & & & & O \end{bmatrix}. \tag{3.7}$$

Unlike to the representation of $\mathbf{P}_\beta - \mathbf{A}$ in (3.5), the $G+1$ diagonal sub-blocks are now zero. However, any other nonzero block remains unchanged. We can heuristically observe that the choice of β requires a balanced consideration: the contributions from (E,E) and (g,I) blocks of $\mathbf{R}_\beta$ for $g=1,\ldots,G$ tend to become larger and smaller, respectively, as β increases.

Remark 3.1. The main effect of the relaxation parameter β in [19] is to replace the last diagonal block P, which may be (nearly) singular, by βI_π to be inverted in a numerically stable way. The constraint $\beta \geq \|P\|_\infty$ must be imposed in an optimal selection of β.

Remark 3.2. In the particular case $\beta = 0$, the relation $\mathbf{P}_\beta = \mathbf{Q}_\beta$ holds and there are only G nonzero blocks in (3.5) and (3.7). However, this choice is probably not optimal because the off-diagonal blocks have the form $D_{gE}A_E^{-1}D_{EI}$ for $g=1,\ldots,G$.

It is straightforward to verify that both $\mathbf{Q}_1$ and $\mathbf{Q}_2$ defined by (3.6) can be written in the factored form

$$\begin{aligned}
\mathbf{Q}_1 = &\begin{bmatrix} I_\pi & & & & \\ & \ddots & & & \\ & & I_\pi & & \\ D_{E1}A_1^{-1} & \cdots & D_{EG}A_G^{-1} & I_\pi & \\ & & & & I_\pi \end{bmatrix} \\
&\times \begin{bmatrix} A_1 & & & D_{1E} & \\ & \ddots & & \vdots & \\ & & A_G & D_{GE} & \\ & & & A_E+\beta I_\pi - \sum_{g=1}^{G} D_{Eg}A_g^{-1}D_{gE} & \\ & & & & I_\pi \end{bmatrix},
\end{aligned}$$

$$\mathbf{Q}_2 = \begin{bmatrix} I_\pi & & & & \\ & \ddots & & & \\ & & I_\pi & & \\ & & & I_\pi & \\ & & & D_{IE} & I_\pi \end{bmatrix} \begin{bmatrix} I_\pi & & & & \\ & \ddots & & & \\ & & I_\pi & & \\ & & & I_\pi & (A_E+\beta I_\pi)^{-1}D_{EI} \\ & & & & A_I - D_{IE}(A_E+\beta I_\pi)^{-1}D_{EI} \end{bmatrix}.$$

These factorisations show that to find the outgoing solution $\mathbf{w} = (w_1^\top, \cdots, w_G^\top, w_E^\top, w_I^\top)^\top$ for a given incoming vector $\mathbf{b} = (b_1^\top, \cdots, b_G^\top, b_E^\top, b_I^\top)^\top$ provided by the product of the system matrix and a certain residual vector, one can implement our relaxed splitting preconditioner — i.e. the preconditioning operation $\mathbf{w} = \mathbf{Q}_\beta^{-1}\mathbf{b}$ in the following steps:

1. Intermediate electron segment.

$$w_E^* = \left(A_E + \beta I_\pi - \sum_{g=1}^{G} D_{Eg} A_g^{-1} D_{gE} \right)^{-1} \left(b_E - \sum_{g=1}^{G} D_{Eg} A_g^{-1} b_g \right).$$

2. $w_g = A_g^{-1}\big(b_g - D_{gE} w_E^*\big)$, $g = 1, \ldots, G$.
3. $w_I = \big[A_I - D_{IE}(A_E + \beta I_\pi)^{-1} D_{EI}\big]^{-1}\big(b_I - D_{IE} w_E^*\big)$.
4. $w_E = w_E^* - (A_E + \beta I_\pi)^{-1} D_{EI} w_I$.

We note that the matrices

$$S_E := A_E + \beta I_\pi - \sum_{g=1}^{G} D_{Eg} A_g^{-1} D_{gE}, \quad \text{and} \quad S_I := A_I - D_{IE}(A_E + \beta I_\pi)^{-1} D_{EI} \tag{3.8}$$

are the exact Schur complements of

$$\begin{bmatrix} A_1 & & & D_{1E} \\ & \ddots & & \vdots \\ & & A_G & D_{GE} \\ D_{E1} & \cdots & D_{EG} & A_E + \beta I_\pi \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} A_E + \beta I_\pi & D_{EI} \\ D_{IE} & A_I \end{bmatrix}$$

with respect to the variables T_E and T_I, respectively. They should not be formed explicitly because A_g^{-1}, $g = 1, \ldots, G$ and $(A_E + \beta I_\pi)^{-1}$ are dense. Therefore, proper approximations have to be used. Let us note the following features.

1. The construction of S_E may be avoided, not the exact block but a diagonal approximation

$$\tilde{S}_E = A_E + \beta I_\pi - \sum_{g=1}^{G} D_{Eg} \Delta_g^{-1} D_{gE}, \tag{3.9}$$

   where Δ_g, $g = 1, \ldots, G$ can be chosen as

Option 1. $\zeta_g^{-1}\mathbf{diag}(A_g)$ with the parameter $\zeta_g = \|\mathbf{diag}^{-1}(A_g)A_g\|_{\infty}^{-1}$, where $\mathbf{diag}(A_g)$ denoting the diagonal matrix comprised of A_g diagonal entries.

Option 2. Row-maximum norm: $\mathrm{diag}(\max_j |a_{1j}^g|, \cdots, \max_j |a_{nj}^g|)$ with A_g entries denoted by a_{ij}^g for $i,j = 1,\ldots,n$.

Option 3. Row-infinity norm: $\mathrm{diag}(\sum_{j=1}^n |a_{1j}^g|, \cdots, \sum_{j=1}^n |a_{nj}^g|)$.

Option 4. Row-Schur norm: $\mathrm{diag}(\sqrt{\sum_{j=1}^n (a_{1j}^g)^2}, \cdots, \sqrt{\sum_{j=1}^n (a_{nj}^g)^2})$, where $\mathrm{diag}(\cdot)$ indicates a block diagonal matrix using input entries.

2. S_I can be efficiently approximated by the matching argument from [36] for PDE-constrained optimisation. This Schur complement matching strategy results in the following approximation:

$$\tilde{S}_I = (A_I + D_{IE})(A_E + \beta I_\pi)^{-1}(A_E + \beta I_\pi - D_{EI}). \tag{3.10}$$

The approximations of steps 1 and 3 lead to our practical implementations — viz.

1†. $w_E^* = \tilde{S}_E^{-1}\left(b_E - \sum_{g=1}^G D_{Eg}A_g^{-1}b_g\right)$;

3†. $w_I^{(1)} = (A_I + D_{IE})^{-1}(b_I - D_{IE}w_E^*)$; $w_I^{(2)} = (A_E + \beta I_\pi)w_I^{(1)}$; $w_I = (A_E + \beta I_\pi - D_{EI})^{-1}w_I^{(2)}$.

This bring about two additional nonzero blocks in $\mathbf{R}_\beta$. Furthermore, steps 1†, 2, 3† and 4 show that our relaxed splitting application requires $G+4$ preconditioner setups and $2G+3$ inner solutions with the submatrices A_g, $\tilde{S}_E$, $A_E + \beta I_\pi$ and $A_E + \beta I_\pi - D_{EI}$, whose inverses are replaced by a number of AMG V-cycles.

Remark 3.3. In practical computations, $A_E + \beta I_\pi$, $\tilde{S}_E$ and $A_E + \beta I_\pi - D_{EI}$ have not be explicitly formed because βI_π, $\beta I_\pi - \sum_{g=1}^G D_{Eg}\Delta_g^{-1}D_{gE}$ and $\beta I_\pi - D_{EI}$ are diagonal.

Remark 3.4. It is worth noting that if $G > 3$, the relaxed splitting preconditioner contains less nonzero blocks in the difference matrix than the two types of block Schur complement preconditioners in [51] with $G^2 + G$ and $G^2 - G$ nonzero blocks, respectively. Observe that both these two block Schur complement preconditioners require $G+2$ preconditioner setups as well as $G+4$ and $G+3$ preconditioner applications, respectively. Therefore, the relaxed splitting preconditioner is more efficient if it converges to a relative tolerance in at most one half iterations.

Remark 3.5. Note that S_I can also be approximated by the above diagonal approximation. However, this produces a relatively worse convergence than the approximation (3.10), so that additional overhead might be a price to pay for a satisfactory convergence.

3.1. Parallel implementation issues

Parallel computing using the message passing interface (MPI) [20] is the de-facto standard for solving large-scale linear systems on modern distributed memory high performance computing architectures. The global communicator is subdivided into $G+2$ non-overlapping communication subgroups using the MPI function `MPI_Comm_split` and each subsystem owns a standalone subgroup, which allows to distribute DoFs onto different compute nodes in a roughly uniform way. Fig. 1 shows a diagram illustrating this procedure with the subgroups COMM_1, $\cdots$, COMM_G, COMM_E and COMM_I. Each of them possesses an ordered collection of processor identifiers with their `key` values $0,1,\ldots,q-1$ of the same `color` value. It should be noted that we use high performance preconditioners and solvers featuring multigrid (HYPRE) library [17] as the distributed linear algebra backend and good candidate for fast approximations of submatrix inverses that arise. The subblocks $A_E+\beta I_\pi$, A_I and A_g, $g=1,\ldots,G$, are stored separately in ParCSR form while the patches $D_{gg'}$ $(g\neq g')$, E_g, T_E, T_I and $f_{g'}$, $g,g'=1,\ldots,G,E,I$ are deposited in ParVector scheme. Each processor contains a stripe of rows with continuous indices. We refer the reader to [17] for a more detailed description of ParCSR and ParVector data structures. The detailed exposition of Fig. 1 is as follows:

- Operation (a). Solve for v_g from $A_g v_g = b_g$ by BoomerAMG [25] with prescribed relative tolerance δ_g and maximum number of iterations $n_g^{\max}$ within subgroups COMM_g for $g=1,\ldots,G$.

- Data transfer (A). Send v_g, Δ_g and D_{gE} from COMM_g to COMM_E among the processors of the same `key`.

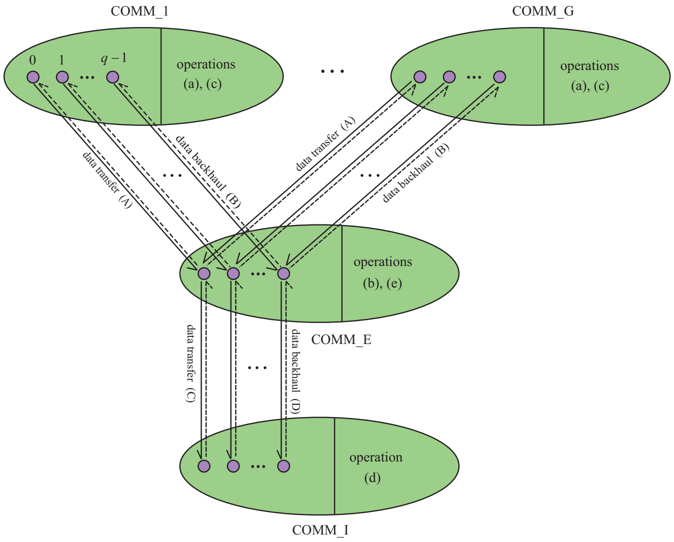

Figure 1: Graphical representation of the $(G+2)q$-processors partitioning, MPI communication pattern of $\mathbf{Q}_\beta$ and its parallel operations labeled using (a)-(e) which are simply executed in sequence.

- Operation (b). After packet is received, the matrix $\tilde{S}_E$ is computed by formula (3.9), the vector b_E is updated by $b_E \leftarrow b_E - \sum_{g=1}^{G} D_{Eg} v_g$ and then the intermediate electron segment w_E^* is obtained from $\tilde{S}_E w_E^* = b_E$ by BoomerAMG with δ_E and $n_E^{\max}$ prescribed within the subgroup COMM_E.

- Data backhaul/transfer (B). Send w_E^* from COMM_E to COMM_g, $g = 1, \ldots, G$ and COMM_I among processors with the same `key` value.

- Operation (c). Receive the message, update $b_g \leftarrow b_g - D_{gE} w_E^*$ and apply A_g^{-1} to the updated vector b_g to get w_g within subgroups COMM_g for $g = 1, \ldots, G$.

- Operation (d). Within the subgroup COMM_I, receive the patch w_E^*, update $b_I \leftarrow b_I - D_{IE} w_E^*$ and yield the solution $w_I^{(1)}$ by virtue of $(A_I + D_{IE}) w_I^{(1)} = b_I$ with δ_I and $n_I^{\max}$ prescribed.

- Data backhaul (C). Within the subgroup COMM_I, send $w_I^{(1)}$ to COMM_E using the same `key`.

- Operation (e). Calculate $w_I^{(2)} \leftarrow (A_E + \beta I_\pi) w_I^{(1)}$, solve for w_I by $(A_E + \beta I_\pi - D_{EI}) w_I = w_I^{(2)}$, send the resultant vector w_I to COMM_I, compute $v_E \leftarrow D_{EI} w_I$, seek the approximate solution s_E given by $(A_E + \beta I_\pi) s_E = v_E$ and obtain the patch $w_E = w_E^* - s_E$ within the subgroup COMM_E.

MPI functions *MPI_Bcast*, *MPI_Send* and *MPI_Recv* are used to exchange data during data backhauls/transfers (A), (B) and (C) processes.

3.2. Spectral property and algebraic selection of parameter β

This subsection is devoted to the spectral property of the relaxed splitting right-preconditioned matrix $\mathbf{A}\mathbf{Q}_\beta^{-1}$ and to the algebraic choice of the relaxation parameter β. It is convenient to introduce $G \times G$ block matrices

$$\mathfrak{A}_R = \mathbf{diagm}(A_1, \cdots, A_G), \quad \mathfrak{I}_\pi = \mathbf{diagm}(I_\pi, \cdots, I_\pi), \quad \mathfrak{O} = \mathbf{diagm}(O, \cdots, O)$$

and G-dimensional block vectors

$$\mathfrak{D}_{RE} = [D_{1E}, \cdots, D_{GE}]^\top, \quad \mathfrak{D}_{ER} = [D_{E1}, \cdots, D_{EG}],$$

where $\mathbf{diagm}(\cdot)$ is the block diagonal matrix assembled by the input matrices. The inverses of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ can be easily obtained in 3×3 block form — viz.

$$\mathbf{Q}_1^{-1} := \begin{bmatrix} \mathfrak{S}_R^{-1} & -\mathfrak{A}_R^{-1}\mathfrak{D}_{RE} S_E^{-1} & \\ -(A_E + \beta I_\pi)^{-1} \mathfrak{D}_{ER} \mathfrak{S}_R^{-1} & S_E^{-1} & \\ & & I_\pi \end{bmatrix},$$

$$\mathbf{Q}_2^{-1} := \left[\begin{array}{ccc} \mathfrak{I}_\pi & & \\ & I_\pi + (A_E+\beta I_\pi)^{-1} D_{EI} S_I^{-1} D_{IE} & -(A_E+\beta I_\pi)^{-1} D_{EI} S_I^{-1} \\ & -S_I^{-1} D_{IE} & S_I^{-1} \end{array}\right],$$

where

$$\mathfrak{S}_R = \mathfrak{A}_R - \mathfrak{D}_{RE}(A_E+\beta I_\pi)^{-1}\mathfrak{D}_{ER},$$

and S_E and S_I are defined by (3.8). The following theorem characterises the eigenvalue distribution of $\mathbf{A}\mathbf{Q}_\beta^{-1}$, or, equivalently, of $\mathbf{Q}_\beta^{-1}\mathbf{A} = \mathbf{I}_\pi - \mathbf{Q}_2^{-1}\mathbf{Q}_1^{-1}\mathbf{R}_\beta$ with $\mathbf{R}_\beta$ defined by (3.7).

Theorem 3.1. *Let* $\mathbf{A}$ *and* $\mathbf{Q}_\beta$ *be given in* (2.4) *and* (3.6). *Then* $(G+1)n$ *eigenvalues of the right-preconditioned matrix* $\mathbf{A}\mathbf{Q}_\beta^{-1}$ *are* 1 *and the remaining eigenvalues are* $1-\mu_i$, $i=1,\dots,n$, *where* μ_i *is the i-th eigenvalue of the matrix*

$$Z_\beta = \beta S_E^{-1} + \left[\beta S_E^{-1} + (A_E+\beta I_\pi)^{-1}\mathfrak{D}_{ER}\mathfrak{S}_R^{-1}\mathfrak{D}_{RE}\right](A_E+\beta I_\pi)^{-1} D_{EI} S_I^{-1} D_{IE}. \tag{3.11}$$

Proof. The proof is analogous to that of [19, Theorem 2.1] and is omitted here. □

Our next goal is to choose such a parameter β, which would make the eigenvalues of Z_β to cluster around zero. This would reduce the number of iteration steps in the right-preconditioned GMRES(m). According to the arguments of Benzi *et al.* [5] and Frigo *et al.* [19], we note that

$$\beta_\dagger = \arg\min_{\beta>0}\left[(G+2)n - \operatorname{trace}(\mathbf{A}\mathbf{Q}_\beta^{-1})\right] = \arg\min_{\beta>0}\left[\operatorname{trace}(Z_\beta)\right], \tag{3.12}$$

where Z_β is defined by (3.11). It is an expensive task to determine analytically the trace of Z_β, since it is a dense matrix because of the presence of the inverses S_E^{-1}, $(A_E+\beta I_\pi)^{-1}$, $\mathfrak{S}_R^{-1}$ and S_I^{-1}. In order to make (3.12) affordable, in our implementation these inverses are replaced by diagonal matrices similar to (3.9). As the result, we obtain the cost functional

$$\operatorname{trace}(Z_\beta) \simeq \sum_{i=1}^{n} Z_\beta^{(i)},$$

where

$$\begin{aligned} Z_\beta^{(i)} = {} & \frac{\beta}{a_{ii}^E + \beta - \sum_{g=1}^G d_i^{Eg} a_{ii}^g d_i^{gE}} + \frac{d_i^{EI} d_i^{IE}}{a_{ii}^E + \beta}\left(a_{ii}^I - \frac{d_i^{EI} d_i^{IE}}{a_{ii}^E+\beta}\right) \\ & \times \left[\frac{\beta}{a_{ii}^E + \beta - \sum_{g=1}^G d_i^{Eg} a_{ii}^g d_i^{gE}} + \sum_{g=1}^{G}\frac{d_i^{Eg} d_i^{gE}}{(a_{ii}^E+\beta)a_{ii}^g - d_i^{Eg} d_i^{gE}}\right] \end{aligned} \tag{3.13}$$

with a_{ii}^E, d_i^{Eg}, a_{ii}^g, d_i^{gE}, d_i^{EI} and d_i^{IE} are the i-th diagonal elements of A_E, D_{Eg}, A_g, D_{gE}, D_{EI} and D_{IE}, respectively. It is easily seen that if trace(Z_β) is the minimum subject to the constraint $\beta > 0$, then so is each entry $Z_\beta^{(i)}$. By zeroing out the derivative of $Z_\beta^{(i)}$ with regard to β, the minimum of $Z_\beta^{(i)}$ is reached at a certain positive scalar β_i. Such a cost is perfectly acceptable. Then their least-square solution takes the form

$$\beta_\ddagger = \frac{1}{n}\sum_{i=1}^{n}\beta_i. \tag{3.14}$$

Unlike (3.12), only n simple nonlinear equations would have to be minimised in parallel followed by the MPI function *MPI_Reduce* is invoked to compute $\beta_\ddagger$ and then *MPI_Bcast* to broadcast $\beta_\ddagger$ within subgroup COMM_E just before the application of our relaxed splitting preconditioning.

Remark 3.6. The function `fmincon` provided by MATLAB can be utilised via a mixed programming to solve the minimisation problem of one variable (3.13) satisfying the constraint $\beta > 0$.

4. Lower Block Triangular Preconditioner

In working toward a lower block triangular preconditioner, we follow the ideas of Ipsen [28] thus obtaining the preconditioner

$$\mathbf{P}_t = \begin{bmatrix} \hat{A}_1 & & & & \\ & \ddots & & & \\ & & \hat{A}_G & & \\ D_{E1} & \cdots & D_{EG} & S & \\ & & & D_{IE} & \hat{A}_I \end{bmatrix},$$

which we propose to use with the GMRES(m) algorithm. Here, S is the Schur complement matrix defined by

$$S = A_E - \sum_{g=1}^{G} D_{Eg}A_g^{-1}D_{gE} - D_{EI}A_I^{-1}D_{IE} \tag{4.1}$$

and $\hat{A}_g$, $g = 1,\ldots,G,I$ are chosen so that the inequality

$$\bar{c}_g \le \frac{\langle A_g y, y\rangle}{\langle \hat{A}_g y, y\rangle} \le \bar{C}_g \tag{4.2}$$

is valid for any nonzero vector $y \in \mathbb{R}^n$ and $\bar{c}_g$, $\bar{C}_g$ are positive constants independent of the step-size and the physical parameters. Similar to the previous considerations, the term S of (4.1) is approximated by

$$\tilde{S} = A_E - \sum_{g=1}^{G} D_{Eg}\Delta_g^{-1}D_{gE} - D_{EI}\Delta_I^{-1}D_{IE}, \tag{4.3}$$

where Δ_g and Δ_I are diagonal matrices implemented according to Options 1-4 above. In this way we obtain a specific implementation procedure for the preconditioning operation $\mathbf{w} = \mathbf{P}_t^{-1}\mathbf{b}$, viz.

1. $w_g = \hat{A}_g^{-1} b_g,\ g = 1,\ldots,G;$

2. $w_E = \tilde{S}^{-1}\big(b_E - \sum_{g=1}^{G} D_{Eg} w_g\big);$

3. $w_I = \hat{A}_I^{-1}(b_I - D_{IE} w_E).$

Remark 4.1. We note that the explicit representation of the sub-blocks $\hat{A}_g$, $g = 1,\ldots,G,I$ is not needed. The constraints (4.2) are introduced for computationally efficient choice of preconditioners, including AMG and domain decomposition ones [12,49], in order to make $\bar{c}_g$ and $\bar{C}_g$ sufficiently close to each other.

Fig. 2 schematically shows the parallel flow diagram of $\mathbf{P}_t$.

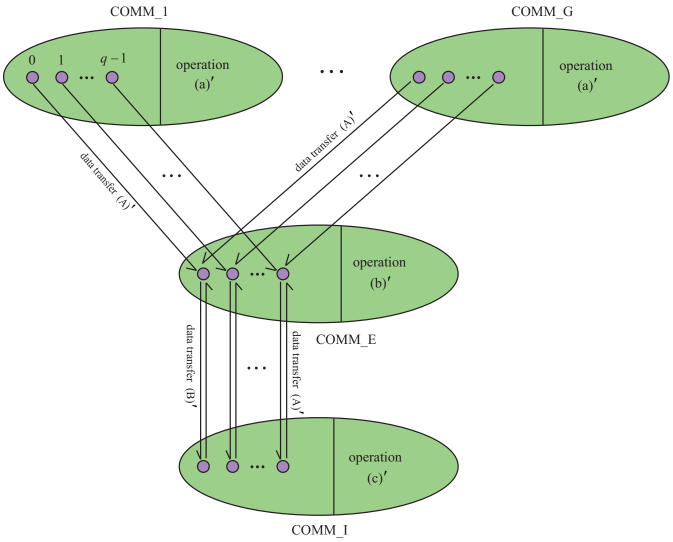

Figure 2: Schematic representation of the MPI communication pattern of $\mathbf{P}_t$ and its parallel operations labeled using (a)′-(c)′.

Specifically, we have:

- Operation (a)′. Determine w_g from $\hat{A}_g w_g = b_g$, $g = 1,\ldots,G$ within subgroups COMM_g.

- Data transfer (A)′. Among processors of the same `key`, send w_g, Δ_g and D_{gE} from COMM_g to COMM_E, and send Δ_I from COMM_I to COMM_E, while D_{IE} need not be sent since $D_{EI} = D_{IE}$.

- Operation (b)′. The matrix $\tilde{S}$ is calculated by (4.3), the vector b_E is updated by $b_E \leftarrow b_E - \sum_{g=1}^{G} D_{Eg} w_g$. The solution w_E is then obtained from $\tilde{S} w_E = b_E$ within the subgroup COMM_E.

- Data transfer (B)′. Send w_E from COMM_E to COMM_I among processors with the same `key`.

- Operation (c)$'$. Update $b_I \leftarrow b_I - D_{IE}w_E$ and apply $\hat{A}_I^{-1}$ to b_I to get w_I within the subgroup COMM_I.

5. Performance Assessment

The numerical tests in this section are aimed to demonstrate the efficiency, robustness, and the strong and weak parallel scaling properties of our implementation integrated into JXPAMG (parallel AMG solver developed by IAPCM and XTU) library [35], by a comparison with the BoomerAMG [25] and Euclid(1) [27] preconditioners implemented in HYPRE library (version 2.20.0) [17]. The BoomerAMG parameters are a strength-of-connection tolerance of 0.25, a single V(1,1)-cycle with Falgout (a hybrid Ruge-Stüben / CLJP) coarsening, classical modified interpolation to transfer solutions between adjacent levels, coarse-grid matrices obtained algebraically using the Galerkin projection, aggressive coarsening on the finest level, hybrid Gauss-Seidel smoothing in the symmetric ordering and at most 100 DoFs on the coarsest level. We use the pure MPI-based parallelism. Results are obtained over Tianhe-2 supercomputer [31], whose compute nodes are interconnected via a proprietary high-speed network. Each compute node has two 12-core 2.2 gigahertz Intel Xeon E5-2692 v2 CPUs and 64 gigabyte memory. These 24 CPU cores per compute node are all used while the Intel Xeon Phi coprocessors are not utilised. Codes are compiled with the Tianhe's self-actualised mpich-3.2 using the icc compiler version 18.0.0 and -O2 optimisation level. The right-preconditioned GMRES(m) iterations are halted once the ratio of the Euclidean norm between the current residual vector and the right-hand side is smaller than 10^{-7} or the maximum iteration count of 200 is achieved, where the zero vector is taken as the initial guess. Herein we set $\delta_g = 10^{-4}$ and $n_g^{\max} = 3$ in Operations (a), (c) and (a)$'$ as well as $\delta_E = \delta_I = 10^{-2}$ and $n_E^{\max} = n_I^{\max} = 1$ in Operations (b), (d), (e), (b)$'$ and (c)$'$. The numerical performance is assessed in accordance with the number of iterations n_{it}, CPU time-to-solution measured in seconds t_{cpu}^p by the MPI function *MPI_Wtime* using p processors and the parallel weak efficiency e_p^k obtained according to t_{cpu}^p / t_{cpu}^{kp} [43].

5.1. Numerical results on one processor

We first exploit the impact of four different representations of Δ_g in (3.9) on our relaxed splitting right-preconditioned GMRES(30) algorithms using $\beta = \beta_{opt}$, where β_{opt} is calculated empirically by the trial-and-error procedure. Herein we consider 6 representative twenty-group MGD linearised algebraic systems, denoted by M_1-M_6, respectively from the 1-st nonlinear iteration at the 22-nd, 41-st, 42-nd, 56-th, 58-th and 59-th time-level on 16.000×48 grid with the problem size roughly 16.9M DoFs. The comparison results are presented in Table 1, from which we can see that Option 4 yields the fastest rate of decrease for the residual norm.

Next we study the dependence of the relaxed splitting right-preconditioned GMRES(30) iteration count and CPU time-to-solution on the relaxation parameter β using Option 4. The results refer to problems M_1-M_6 as shown in Table 2, where $\beta_{\ddagger}$ is computed by the

Table 1: Effect of different representations of Δ_g on the right-preconditioned GMRES(30) iteration counts on 16.000×48 grid.

	Option 1		Option 2		Option 3		Option 4	
	n_{it}	t^1_{cpu}	n_{it}	t^1_{cpu}	n_{it}	t^1_{cpu}	n_{it}	t^1_{cpu}
M_1	3	35.3	4	51.1	3	36.4	2	25.0
M_2	3	34.9	4	47.1	3	34.6	3	35.6
M_3	4	46.8	6	72.9	4	46.1	3	35.7
M_4	4	51.0	6	74.2	5	63.7	4	49.8
M_5	4	50.6	5	63.5	4	50.9	3	38.4
M_6	4	50.3	7	89.7	5	63.0	4	49.3

criterion (3.14). Fig. 3 gives more details on how the relaxed splitting right-preconditioned GMRES(30) iteration counts behave with respect to β. We also note the following features.

(i) $\beta_{opt} \in (0,1)$.

(ii) $\beta_{\ddagger}$ is very close to β_{opt} with only one more step and $\beta_{\ddagger}$ is much easier to find.

(iii) The case $\beta = 0$ leads to poor performance compared with $\beta_{\ddagger}$.

(iv) The relaxed splitting right-preconditioned GMRES(30) converges robustly with respect to problem character.

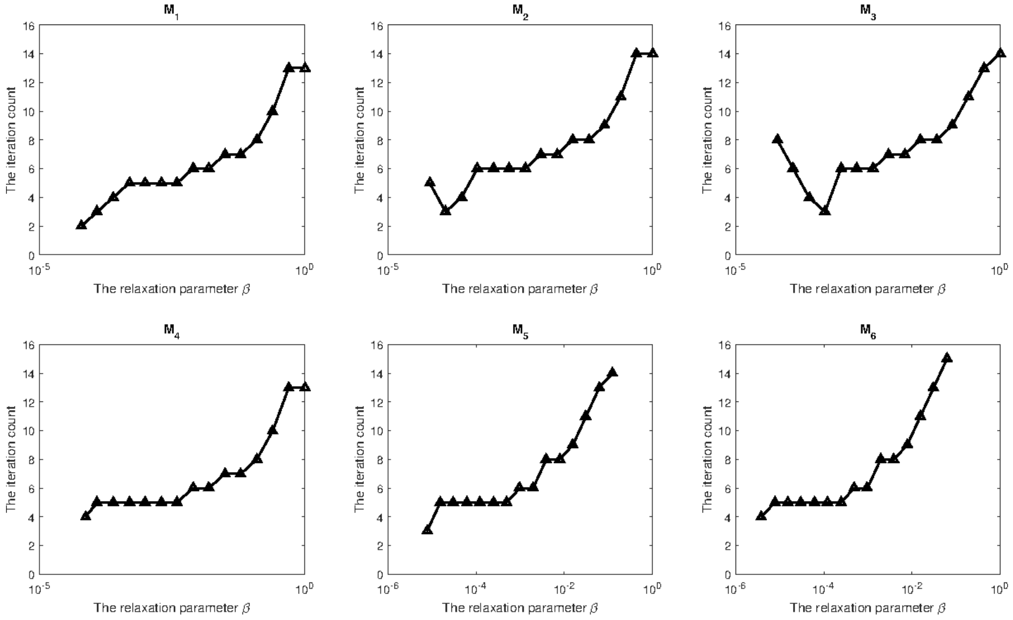

Figure 3: Relaxed splitting right-preconditioned GMRES(30) iteration counts regarding the choice of β.

Table 2: Comparison of the iteration count and CPU time-to-solution between 0, $\beta_{\ddagger}$ and β_{opt} on $16,000 \times 48$ grid.

	$\beta_{\ddagger}$	β_{opt}	n_{it}			t^1_{cpu}		
			$\beta=0$	$\beta=\beta_{\ddagger}$	$\beta=\beta_{opt}$	$\beta=0$	$\beta=\beta_{\ddagger}$	$\beta=\beta_{opt}$
M_1	6.28E-5	6.10E-5	4	3	2	49.1	35.4	25.0
M_2	1.07E-4	1.22E-4	5	4	3	60.9	47.3	35.6
M_3	4.26E-4	4.85E-4	8	4	3	104.4	45.9	35.7
M_4	8.32E-5	7.29E-5	6	5	4	73.8	58.6	49.8
M_5	8.49E-6	7.63E-6	7	4	3	85.7	46.8	38.4
M_6	4.52E-6	3.81E-6	6	5	4	74.4	58.4	49.3

Table 3: Iteration count and CPU time-to-solution comparison of relaxed splitting preconditioner with three popular preconditioners on 16.000×48 grid.

	RS-AMG		ILU(1)		lower block triangular		relaxed splitting	
	n_{it}	t^1_{cpu}	n_{it}	t^1_{cpu}	n_{it}	t^1_{cpu}	n_{it}	t^1_{cpu}
M_1	43	211.0	2	9.4	17	106.5	3	35.4
M_2	27	131.9	18	35.6	14	81.6	4	47.3
M_3	26	126.3	2	9.3	16	93.2	4	45.9
M_4	53	253.4	2	9.6	18	106.1	5	58.6
M_5	34	174.1	> 200	-	16	94.3	4	46.8
M_6	37	185.6	> 200	-	16	92.7	5	58.4

These results show that the criterion (3.14) is sufficient to guarantee a relatively excellent performance.

We report in Table 3 the iteration count comparison among the classical Ruge-Stüben style AMG (RS-AMG) [40], incomplete LU factorisation with one fill-in denoted by ILU(1) [41], lower block triangular and relaxed splitting right-preconditioned GMRES(30) solvers. It is observed that M_5 and M_6 are not solvable by ILU(1) within the maximum number of steps allowed, although ILU(1) runs averagely 2.9 times faster than the relaxed splitting preconditioner for M_1-M_4. Hence, the convergence of ILU(1)-GMRES(30) cannot be predicted in advance. The lower block triangular and relaxed splitting converge robustly, however, on average, the latter converges 3.9 times faster. A notable disadvantage of AMG right-preconditioned GMRES(30) is the substantially varied and excessive number of iterations to achieve convergence. In average, this yields 1.9 slower convergence that the lower block triangular preconditioner and 3.7 times slower than the relaxed splitting one.

5.2. Parallel results

We now vary the number of processors to perform a strong scaling test for fixed global mesh 64.000×96 in order to demonstrate numerically the robust convergence of these four preconditioners and commendable numerical scalability in the strong sense. The local

Table 4: Iteration count comparison of four GMRES(30) solvers in strong scaling test.

	BoomerAMG				Euclid(1)			
np	352	704	1408	2816	352	704	1408	2816
M_1	47	49	49	51	3	3	3	3
M_2	31	33	34	33	21	21	21	22
M_3	29	31	31	30	2	2	3	2
M_4	58	59	57	59	4	4	4	4
M_5	41	42	41	43	> 200	> 200	> 200	> 200
M_6	44	43	46	46	> 200	> 200	> 200	> 200
	lower block triangular				relaxed splitting			
np	352	704	1408	2816	352	704	1408	2816
M_1	19	19	20	21	3	4	3	3
M_2	16	16	18	18	4	4	4	4
M_3	17	17	18	19	4	4	4	4
M_4	20	21	22	22	5	5	5	5
M_5	17	17	18	18	4	4	5	4
M_6	18	18	18	19	5	5	5	5

Table 5: CPU time-to-solution comparison of four GMRES(30) solvers in strong scaling test.

	BoomerAMG				Euclid(1)			
np	352	704	1408	2816	352	704	1408	2816
M_1	18.4	16.1	12.7	9.5	1.8	1.2	0.8	0.5
M_2	12.2	10.9	8.9	6.6	12.7	7.7	4.8	2.9
M_3	11.0	9.8	8.1	6.1	1.3	0.8	0.5	0.3
M_4	20.1	17.6	14.0	10.3	2.6	1.6	0.9	0.6
M_5	14.3	12.7	10.2	7.7	-	-	-	-
M_6	16.2	14.2	11.1	8.3	-	-	-	-
	lower block triangular				relaxed splitting			
np	352	704	1408	2816	352	704	1408	2816
M_1	22.2	12.7	7.6	4.8	6.6	5.2	2.3	1.4
M_2	18.1	10.2	6.5	3.9	9.2	5.4	3.2	1.9
M_3	19.6	10.8	6.4	4.0	9.1	5.3	3.1	1.8
M_4	22.3	12.9	7.9	4.4	11.1	6.5	3.8	2.2
M_5	19.7	11.0	6.8	4.0	9.2	5.4	4.0	1.9
M_6	20.0	11.2	6.3	3.8	11.2	6.6	3.9	2.3

problem size is 48.000 DoFs for 2816 CPU cores. Tables 4 and 5 show that, as the local number of DoFs decreases, the relaxed splitting preconditioner runs on average 2.2 and 4.2 times faster than the lower block triangular preconditioner and BoomerAMG on 2816 CPU cores.

Table 6: Comparisons of four GMRES(30) solvers in weak scaling test. Top: Iteration count. Bottom: CPU time-to-solution.

	BoomerAMG			Euclid(1)			lower block triangular			relaxed splitting		
np	176	704	2816	176	704	2816	176	704	2816	176	704	2816
M_1	47	49	51	2	3	3	21	22	21	3	3	3
M_2	30	32	33	20	22	22	18	18	18	4	4	4
M_3	28	30	30	2	3	2	19	19	19	4	4	4
M_4	54	57	59	3	4	4	21	22	22	5	5	5
M_5	41	43	43	> 200	> 200	> 200	18	18	18	4	5	4
M_6	42	45	46	> 200	> 200	> 200	19	19	19	5	5	5
	BoomerAMG			Euclid(1)			lower block triangular			relaxed splitting		
np	176	704	2816	176	704	2816	176	704	2816	176	704	2816
M_1	4.97	6.83	9.49	0.17	0.34	0.48	3.27	4.12	4.82	0.86	1.08	1.41
M_2	3.42	4.81	6.62	1.41	2.09	2.93	2.63	3.17	3.89	1.14	1.43	1.87
M_3	3.22	4.55	6.08	0.15	0.31	0.29	2.73	3.28	4.03	1.11	1.39	1.82
M_4	5.36	7.46	10.31	0.24	0.43	0.61	2.82	3.56	4.38	1.38	1.72	2.24
M_5	4.15	5.74	7.67	-	-	-	2.69	3.24	3.97	1.16	1.81	1.89
M_6	4.29	6.07	8.29	-	-	-	2.60	3.13	3.84	1.41	1.76	2.30

Finally, we carry out weak scaling analysis for a fixed local mesh 16.000 × 24. The number of CPU cores is increased from 176 to 2816 with 48.000 DoFs per processor, each time quadrupling their number. The global size of the resulting system matrix varies from 8.5M to 135.2M. From Table 6, it is clear that in numerical sense these four preconditioners weakly scale well. The average parallel weak efficiencies of BoomerAMG, Euclid(1), the lower block triangular and relaxed splitting right-preconditioned GMRES(30) solvers are 73.2%, 73.6%, 82.2% and 79.7% using 2816 CPU cores compared to 704 CPU cores, respectively.

6. Conclusion

We introduced a relaxed splitting framework with an inexpensive but stable strategy to estimate the involved relaxation parameter. It is a promising alternative for the efficient monolithic solutions of the multidimensional MGD equations. Numerical experiments show that our relaxed splitting preconditioner scales well both algorithmically and in parallel and the sensitivity to model parameters is small.

Acknowledgments

The authors gratefully acknowledges the editor and anonymous reviewers for their constructive suggestions and comments.

This study is financially supported by Science Challenge Project (TZZT2016002), National Natural Science Foundation of China (11601462, 11971414, 62102167, 62032025), National Key R&D Program of China (2017YFB0202103), Hunan National Applied Mathematics Center (2020ZYT003) and the Project of Scientific Research Fund of Hunan Provincial Science and Technology Department (2018WK4006).

References

[1] J.H. Adler, F.J. Gaspar, X. Hu, P. Ohm, C. Rodrigo and L.T. Zikatanov, *Robust preconditioners for a new stabilized discretization of the poroelastic equations*, SIAM J. Sci. Comput. **42**, B761–B791 (2020).

[2] H.B. An, Z.Y. Mo, X.W. Xu and X.W. Jia, *Operator-based preconditioning for the 2-D 3-T energy equations in radiation hydrodynamics simulations*, J. Comput. Phys. **385**, 51–74 (2019).

[3] P.R. Amestoy, A. Buttari, J.-Y. L'Excellent and T. Mary, *Performance and scalability of the block low-rank multifrontal factorization on multicore architectures*, ACM Trans. Math. Softw. **45**, 2:1–2:26 (2019).

[4] T.S. Axelrod, P.F. Dubois and C.E. Rhoades Jr., *An implicit scheme for calculating time- and frequency-dependent flux limited radiation diffusion in one dimension*, J. Comput. Phys. **54**, 205–220 (1984).

[5] M. Benzi, S. Deparis, G. Grandperrin and A. Quarteroni, *Parameter estimates for the relaxed dimensional factorization preconditioner and application to hemodynamics*, Comput. Methods Appl. Mech. Engrg. **300**, 129–145 (2016).

[6] M. Benzi, M. Ng, Q. Niu and Z. Wang, *A relaxed dimensional factorization preconditioner for the incompressible Navier-Stokes equations*, J. Comput. Phys. **230**, 6185–6202 (2011).

[7] M. Benzi and D.B. Szyld, *Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods*, Numer. Math. **76**, 309–321 (1997).

[8] H. Bin Zubair Syed, C. Farquharson and S. Maclachlan, *Block preconditioning techniques for geophysical electromagnetics*, SIAM J. Sci. Comput. **42**, B696–B721 (2020).

[9] J. Bosch, C. Kahle and M. Stoll, *Preconditioning of a coupled Cahn-Hilliard Navier-Stokes system*, Commun. Comput. Phys. **23**, 603–628 (2018).

[10] P.N. Brown and C.S. Woodward, *Preconditioning strategies for fully implicit radiation diffusion with material-energy transfer*, SIAM J. Sci. Comput. **23**, 499–516 (2001).

[11] Q.M. Bui, D. Osei-Kuffuor, N. Castelletto and J.A. White, *A scalable multigrid reduction framework for multiphase poromechanics of heterogeneous media*, SIAM J. Sci. Comput. **42**, B379–B396 (2020).

[12] T.F. Chan and T.P. Mathew, *Domain decomposition algorithms*, Acta Numer. **3**, 61–143 (1994).

[13] L. Chen, J. Hu and X.H. Huang, *Fast auxiliary space preconditioners for linear elasticity in mixed form*, Math. Comput. **87**, 1601–1633 (2018).

[14] X. Cui, Z.J. Shen and G.W. Yuan, *Asymptotic-preserving discrete schemes for non-equilibrium radiation diffusion problem in spherical and cylindrical symmetrical geometries*, Commun. Comput. Phys. **23**, 198–229 (2018).

[15] E.C. Cyr, J.N. Shadid, R.S. Tuminaro, R.P. Pawlowski and L. Chacón, *A new approximate block factorization preconditioner for two-dimensional incompressible (reduced) resistive MHD*, SIAM J. Sci. Comput. **35**, B701–B730 (2013).

[16] T.A. Davis, S. Rajamanickam and W.M. Sid-Lakhdar, *A survey of direct methods for sparse linear systems*, Acta Numer. **25**, 383–566 (2016).

[17] R.D. Falgout, J.E. Jones and U.M. Yang, *Pursuing scalability for hypre's conceptual interfaces*, ACM Trans. Math. Softw. **31**, 326–350 (2005).

[18] T. Feng, X.J. Yu, H.B. An, Q. Li and R.P. Zhang, *The preconditioned Jacobian-free Newton-Krylov methods for nonequilibrium radiation diffusion equations*, J. Comput. Appl. Math. **255**, 60–73 (2014).

[19] M. Frigo, N. Castelletto and M. Ferronato, *A relaxed physical factorization preconditioner for mixed finite element coupled poromechanics*, SIAM J. Sci. Comput. **41**, B694–B720 (2019).

[20] E. Gabriel, G.E. Fagg, G. Bosilca, T. Angskun, J.J. Dongarra, J.M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R.H. Castain, D.J. Daniel, R.L. Graham and T.S. Woodall, *Open MPI: goals, concept, and design of a next generation MPI implementation*, Lect. Notes Comput. Sci. **3241**, 97–104 (2004).

[21] D. Goik and K. Banaś, *A block preconditioner for scalable large scale finite element incompressible flow simulations*, Lect. Notes Comput. Sci. **12139**, 199–211 (2020).

[22] W. Gong, Z.Y. Tan and S. Zhang, *A robust optimal preconditioner for the mixed finite element discretization of elliptic optimal control problems*, Numer. Linear Algebra Appl. **25**, e2129 (2018).

[23] X.D. Hang, J.H. Li and G.W. Yuan, *Convergence analysis on splitting iterative solution of multi-group radiation diffusion equations*, Chinese J. Comput. Phys. **30**, 111–119 (2013).

[24] P. Hassanzadeh and G.D. Raithby, *Finite-volume solution of the second-order radiative transfer equation: accuracy and solution cost*, Numer. Heat Trans. B **53**, 374–382 (2008).

[25] V.E. Henson and U.M. Yang, *BoomerAMG: a parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math. **41**, 155–177 (2002).

[26] M.R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards **49**, 409–436 (1952).

[27] D. Hysom and A. Pothen, *A scalable parallel algorithm for incomplete factor preconditioning*, SIAM J. Sci. Comput. **22**, 2194–2215 (2001).

[28] I.C.F. Ipsen, *A note on preconditioning nonsymmetric matrices*, SIAM J. Sci. Comput. **23**, 1050–1051 (2001).

[29] L.M. Kačanov, *Variational methods of solution of plasticity problems*, J. Appl. Math. Mech. **23**, 880–883 (1959).

[30] X.S. Li, *An overview of SuperLU: algorithms, implementation, and user interface*, ACM Trans. Math. Softw. **31**, 302–325 (2005).

[31] X.K. Liao, L.Q. Xiao, C.Q. Yang and Y.T. Lu, *MilkyWay-2 supercomputer: system and application*, Front. Comput. Sci. **8**, 345–356 (2014).

[32] J. Liu, W.G. Yang, M. Dong and A.L. Marsden, *The nested block preconditioning technique for the incompressible Navier-Stokes equations with emphasis on hemodynamic simulations*, Comput. Methods Appl. Mech. Engrg. **367**, 113122 (2020).

[33] V.A. Mousseau, D.A. Knoll and W.J. Rider, *Physics-based preconditioning and the Newton-Krylov method for non-equilibrium radiation diffusion*, J. Comput. Phys. **160**, 743–765 (2000).

[34] C.C. Paige and M.A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal. **12**, 617–629 (1975).

[35] *Parallel algebraic multigrid solver software package (JXPAMG)*, `http://www.multigrid.org/solver/jxpamg.html`.

[36] J.W. Pearson and A.J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl. **19**, 816–829 (2012).

[37] W.B. Pei, *The construction of simulation algorithms for laser fusion*, Commun. Comput. Phys. **2**, 255–270 (2007).

[38] G. Peng, Z.M. Gao, W.J. Yan and X.L. Feng, *A positivity-preserving finite volume scheme for three-temperature radiation diffusion equations*, Appl. Numer. Math. **152**, 125–140 (2020).

[39] S. Rhebergen, G.N. Wells, A.J. Wathen and R.F. Katz, *Three-field block preconditioners for models of coupled magma/mantle dynamics*, SIAM J. Sci. Comput. **37**, A2270–A2294 (2015).

[40] J.W. Ruge and K. Stüben, *Algebraic multigrid*, in: *Multigrid Methods*, Front. Appl. Math. **3**, 73–130 (1987).

[41] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM (2003).

[42] Y. Saad and M.H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. **7**, 856–869 (1986).

[43] S. Sahni and V. Thanvantri, *Performance metrices: keeping the focus on runtime*, IEEE Parall. Distrib. **4**, 43–56 (1996).

[44] A. Segal, M. ur Rehman and C. Vuik, *Preconditioners for incompressible Navier-Stokes solvers*, Numer. Math. Theor. Meth. Appl. **3**, 245–275 (2010).

[45] Z.Q. Sheng, J.Y. Yue and G.W. Yuan, *Monotone finite volume schemes of nonequilibrium radiation diffusion equations on distorted meshes*, SIAM J. Sci. Comput. **31**, 2915–2934 (2009).

[46] S. Shu, M.H. Liu, X.W. Xu, X.Q. Yue and S.G. Li, *Algebraic multigrid block triangular preconditioning for multidimensional three-temperature radiation diffusion equations*, Adv. Appl. Math. Mech. **13**, 1203–1226 (2021).

[47] S. Su and J.M. Wu, *A vertex-centered and positivity-preserving finite volume scheme for two-dimensional three-temperature radiation diffusion equations on general polygonal meshes*, Numer. Math. Theor. Meth. Appl. **13**, 220–252 (2020).

[48] H.A. van der Vorst, *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. **13**, 631–644 (1992).

[49] J. Xu and L. Zikatanov, *Algebraic multigrid methods*, Acta Numer. **26**, 591–721 (2017).

[50] X.W. Xu, Z.Y. Mo and H.B. An, *Algebraic two-level iterative method for 2-D 3-T radiation diffusion equations*, Chinese J. Comput. Phys. **26**, 1–8 (2009).

[51] X.Q. Yue, X.W. Xu and S. Shu, *JASMIN-based two-dimensional adaptive combined preconditioner for radiation diffusion equations in inertial fusion research*, East Asian J. Appl. Math. **7**, 495–507 (2017).

[52] X.Q. Yue, S.L. Zhang, X.W. Xu, S. Shu and W.D. Shi, *Algebraic multigrid block preconditioning for multi-group radiation diffusion equations*, Commun. Comput. Phys. **29**, 831–852 (2021).

[53] Z.Y. Zhou, X.W. Xu, S. Shu, C.S. Feng and Z.Y. Mo, *An adaptive two-level preconditioner for 2-D 3-T radiation diffusion equations*, Chinese J. Comput. Phys. **29**, 475–483 (2012).