Neumann Series in MGS-GMRES and Inner-Outer Iterations

Preprint

Stephen Thomas,[1] Katarzyna Swirydowicz,[2] Ruipeng Li,[3] and Paul Mullowney[1]

1 National Renewable Energy Laboratory
2 Pacific Northwest National Laboratory
3 Lawrence Livermore National Laboratory

Presented at SISC Copper Mountain Special Section
March 25-30, 2021

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-2C00-80343
February 2022

Neumann Series in MGS-GMRES and Inner-Outer Iterations

Preprint

Stephen Thomas,[1] Katarzyna Swirydowicz,[2] Ruipeng Li,[3] and Paul Mullowney[1]

1 National Renewable Energy Laboratory
2 Pacific Northwest National Laboratory
3 Lawrence Livermore National Laboratory

Suggested Citation
Thomas, Stephen, Katarzyna Swirydowicz, Ruipeng Li, and Paul Mullowney. 2022. *Neumann Series in MGS-GMRES and Inner-Outer Iterations: Preprint*. Golden, CO: National Renewable Energy Laboratory. NREL/CP-2C00-80343. https://www.nrel.gov/docs/fy22osti/80343.pdf.

Conference Paper
NREL/CP-2C00-80343
February 2022

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Science, Advanced Scientific Computing Research. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

NEUMANN SERIES IN MGS-GMRES AND INNER-OUTER ITERATIONS

STEPHEN THOMAS*, KATARZYNA ŚWIRYDOWICZ†, RUIPENG LI‡, AND PAUL MULLOWNEY*

Abstract. A low-synchronization MGS-GMRES Krylov solver employing a truncated Neumann series for the inverse compact WY MGS correction matrix T is presented. A corollary to the backward stability result of Paige et al. [1] establishes that $T = I - L_k$ is sufficient for convergence of GMRES when $\|L\|_F^p = \mathcal{O}(\varepsilon^p)\kappa_F^p(B)$, $p > 1$ where $B = [r_0, AV_m]$ with condition number $\kappa(B)$. The columns of the strictly lower triangular matrix L are defined by matrix-vector products of Krylov vectors $V_{1:k-2}^T \mathbf{v}_{k-1}$. The preconditioner is the classical Rüge-Stuben AMG algorithm with compatible relaxation and inner-outer Gauss-Seidel smoother. This smoother may also be expressed as a truncated Neumann series. Despite the rapid convergence of GMRES-AMG, the cost of an elliptic pressure solver (e.g. for the Navier-Stokes equations), is still substantial. Drop tolerances are applied to the strictly lower triangular matrices arising in the smoother in order to reduce the number of non-zeros and accelerate the time to solution. The number of small matrix elements are found to increase from fine to coarse levels and thus the efficiency gains are greater for large problems with many levels in the V-cycle. The solver is applied to the pressure continuity equation for the incompressible Navier-Stokes equations. The pressure solve time is reduced considerably without a change in the convergence rate.

1. Introduction. The generalized minimal residual (GMRES) Krylov subspace method [2] is often employed to solve the large linear systems arising in high-resolution physics based simulations using the incompressible Navier-Stokes equations in fluid mechanics. Świrydowicz et al. [3] recently improved the parallel strong-scaling of the algorithm by reducing the associated communication requirements to a minimum, while maintaining the numerical stability and robustness of the original algorithm. In order to achieve fast convergence of an elliptic solver, such as the pressure continuity equation, the classical Ruge-Stüben [4] algebraic multigrid (C-AMG) solver is employed as a preconditioner. Both the one-reduce GMRES iteration and sequential Gauss-Seidel smoother employ triangular solves. The triangular solve in MGS-GMRES is relatively small and local to each MPI rank. For Gauss-Seidel smoothers it is much larger and requires global communication. Triangular solvers are, in general, difficult to implement in parallel on many-core architectures. In this study, they are replaced with truncated Neumann series expansions. They result in a highly efficient and stable approach for Exascale class computers based on graphical processing units (GPUs).

Let A be an $n \times n$ real-valued matrix. In the present study, consider the solution of the linear system $A\mathbf{x} = \mathbf{b}$, with the MGS-GMRES [2] Krylov subspace method, using one V-cycle of the classical algebraic multigrid C-AMG algorithm of Rüge-Stuben as the preconditioner. Inside MGS-GMRES, the Arnoldi QR algorithm is applied to generate an orthonormal basis V_m for the Krylov subspace $\mathcal{K}_m$ and the Hessenberg matrix $H_{m+1,m}$ in the Arnoldi expansion. The modified Gram-Schmidt algorithm produces a QR factorization of the matrix $B = [\, b,\, AV_m \,]$. The size of this basis is $m \ll n$.

The orthogonality of the basis, V_m, for the Krylov subspace $\mathcal{K}_m(B)$ is desirable for convergence of Krylov methods for linear system solvers. However, in finite-precision arithmetic, V_m may "lose" orthogonality. The loss of orthogonality of the computed basis – as measured by $\|I - V_m^T V_m\|_F$ – may deviate substantially from machine precision $\mathcal{O}(\varepsilon)$, (see Giraud et al. [5]). When linear independence is completely lost, the Krylov iterations may fail to converge. For example, the GMRES iteration will stall and fail to converge if linear independence of the Krylov vectors is completely lost. This is the case when $\|S\|_2 = 1$ as described by Paige [6], where the matrix S was introduced in Paige et al. [1].

The development of low-synchronization Gram-Schmidt and generalized minimal residual algorithms by Świrydowicz et al. [3] was largely driven by applications that need stable, yet scalable solvers. Both the modified (MGS) and classical Gram-Schmidt (CGS2) with re-orthogonalization are stable algorithms for a GMRES solver. Indeed, CGS2 produces an $\mathcal{O}(\varepsilon)$ loss of orthogonality, which suffices for GMRES to converge. Paige et al. [1] show that, despite $\mathcal{O}(\varepsilon)\kappa(B)$ loss of orthogonality, MGS-GMRES is backward stable for the solution of linear systems. Here, the condition number of the matrix B is given by $\kappa(B) = \sigma_{\max}(B)/\sigma_{\min}(B)$, where $\sigma_i(B)$ are the singular values of the matrix B.

A one-reduce modified Gram-Schmidt QR factorization algorithm is presented in [3] and is based

*National Renewable Energy Laboratory, Golden, Colorado
†Pacific Northwest National Laboratory, Richland, WA
‡Lawrence Livermore National Laboratory, Livermore CA

upon the application of a projector

$$P = I - Q\,T\,Q^T, \quad T = (\,Q^TQ\,)^{-1}$$

where Q is $m \times n$, I is the identity, and T is an $n \times n$ correction matrix. To obtain a one-reduce MGS algorithm, the normalization is delayed to the next iteration. The correction matrix T is obtained from $L = \text{tril}(\,Q^TQ,\,-1\,)$ (see Table 1 for notation reference), the strictly lower triangular part of Q^TQ. Note that, because Q has almost orthonormal columns, the norm of L is small, and T is close to I.

A Neumann series expansion for the inverse of the lower triangular correction matrix T results in the inverse compact WY representation of the projector P for the modified Gram-Schmidt factorization of the Arnoldi matrix B,

$$P = I - V_k\,T\,V_k^T, \quad T = (\,I + L_k\,)^{-1} = I - L_k + L_k^2 - \cdots + L_k^p,$$

where the columns of the strictly lower triangular matrix L_k are defined by the matrix-vector products $V_{k-2}^T\mathbf{v}_{k-1}$. The sum is finite because the strictly lower triangular matrix L_k is idempotent, as originally noted by Ruhe [7]. Here, a corollary to the backward stability result of Paige et al. [1] demonstrates that $T = I - L_k$ is sufficient for convergence of MGS-GMRES when $\|L_k\|_F^p = \mathcal{O}(\varepsilon^p)\kappa_F^p(B)$, where $p > 1$. A new formulation of GMRES based upon the truncated Neumann series for the correction matrix T is presented here along with loss of orthogonality experiments. In particular, the loss of orthogonality for the ICWY-MGS and original GMRES algorithms are compared. For the ill-conditioned matrices examined by Greenbaum et al. [8], the convergence history of the truncated Neumann series version of the algorithm is shown to be identical to the original algorithm introduced by Saad and Schultz [2]. In particular, when the norm-wise relative backward error (NRBE) reaches machine precision, the Krylov vectors lose linear independence and convergence stalls. For this reason, Paige and Strakoš [9] recommended that the NBRE be applied as the stopping criterion.

Our proof of the corollary relies on the equivalence, in finite precision arithmetic, of the modified Gram-Schmidt and Householder QR factorizations for an augmented matrix as discussed in Paige and Björck [10]. Most notably, the inverse compact WY MGS projector derived in Świrydowicz et al. [3] can be obtained from the $(2,2)$ block of the augmented Householder transformation matrix P and the identities arising from the 'correctly' normalized matrices $\tilde{V}_k$ defined by Paige et al. [1]. These augmented results are also examined in Giraud et al. [11].

With respect to the C-AMG preconditioner, the two-stage, or inner-outer Gauss-Seidel preconditioners and smoothers studied by Szyld [12–14] can also be expressed in terms of a degree-s Neumann series expansion as described in the recent papers by Thomas et al. [15] and Mullowney et al. [16]. Given the matrix splitting (note that the matrix L in Gauss-Seidel smoother is different from the L used in Gram-Schmidt projection) $A = M - N$, $A = D + L + U$, $M = D + L$ and $N = -U$, the two-stage Gauss-Seidel iteration is given by

$$\mathbf{x}_{k+1} := \mathbf{x}_k + (I + D^{-1}L)^{-1}\,D^{-1}\mathbf{r}_k = \mathbf{x}_k + \sum_{j=0}^{s}(-D^{-1}L)^j D^{-1}\mathbf{r}_k$$

In the present study, the above iteration is applied as a smoother for C-AMG, which is applied as a preconditioner to accelerate the convergence of MGS-GMRES. For the matrix M arising in the matrix splitting employed by the smoother, small elements are dropped in order to reduce the number of non-zeros at each level of the V-cycle, and thereby reduce the computation time, while maintaining the same convergence rate of the solver.

An additional contribution of our paper is to combine three different strategies to create fast and robust solvers. The time to solution is the most important metric versus the number of solver iterations taken in our fluid mechanics simulations. The low-synchronization MGS-GMRES may take a larger number of iterations at little or no extra cost if the preconditioner is less expensive. However, the sparse triangular solvers employed by smoothers are not efficient on multi-core architectures. This observation prompted the introduction of inner-outer AMG smoothers in [16], which significantly lowers the cost of the V-cycle. This cost can be further lowered by applying the dropping strategy proposed herein.

The paper is organized as follows. Low synchronization and generalized minimal residual algorithms are discussed in section 2. A corollary to Paige et al. [1] in Section 3 establishes that a truncated

Neumann series $T = I - L_k$ is sufficient for MGS-GMRES to converge. In section 4, a variant of MGS-GMRES is derived that uses this T. Numerical experiments then illustrate the properties of the new algorithm. The inner-outer Gauss-Seidel smoother is reviewed in Section 5 and it is shown how dropping reduces the number of non-zeros per row in the smoother without a deterioration in the convergence rate.

Symbol/expression/function	Explanation
Lowercase bold letters, i.e., $\mathbf{v}$	a column vector
Uppercase letters, i.e., V, A, H, D	a matrix
Lowercase letter with two subscripts, i.e., a_{ij}	element of matrix A in row i and column j
Uppercase P	projection operator
Uppercase T	triangular correction matrix in MGS projector
Uppercase letter with subscript, i.e.,V_k	matrix with k columns
Lowercase letter with subscript, i.e., $\mathbf{v}_k$	k–th column of matrix V
Uppercase L	strictly lower triangular matrix
Uppercase R and U	upper triangular matrices
ε	machine epsilon
Uppercase matrix $H_{k+1,k}$	Hessenberg matrix H size $(k+1) \times k$
`tril(A, -n)`	Lower triangular part of A, starting at n diagonals below main diagonal
$\kappa(A)$	condition number of matrix A

Table 1: Notation used in this paper for vectors, matrices and operators.

2. Low-Synchronization Gram-Schmidt Algorithms. As mentioned previously, Krylov linear system solvers are often required for extreme scale physics simulations and implemented on parallel (distributed memory) machines with many-core accelerators. Their strong-scaling is limited by the number and frequency of global reductions in the form of `MPI_AllReduce`. These communication patterns are expensive [17]. Low-synchronization orthogonalization algorithms are designed such that they require only one reduction per iteration to normalize each vector and apply projections.

A review of Gram Schmidt algorithms and their computational costs is given in [18]. The inverse compact WY form for MGS is the lower triangular matrix $T = (I + L_k)^{-1}$, analogous to Puglisi [19]. An upper triangular T for MGS which is constructed with recursive matrix products was introduced Malard and Paige [20] and corresponds to the Schreiber and Van Loan compact WY Householder [21] representation. This was recently generalized to block Gram-Schmidt algorithms by Barlow [22]. The inverse compact WY form of MGS was derived in Świrydowicz et al. [3]. The ICWY-MGS algorithm batches the projections together and computes one row of the strictly lower triangular matrix,

$$L^T_{k-1,1:k-2} = V^T_{k-2}\, \mathbf{v}_{k-1}.$$

The resulting inverse compact WY projector P is given by

$$P = I - V_{j-1}\, T_{j-1}\, V^T_{j-1}, \quad T_{j-1} = (\, I + L_{j-1}\,)^{-1}$$

The implied triangular solve requires an additional $(j-1)^2$ flops at iteration $j-1$ and thus leads to a slightly higher operation count compared to the original MGS algorithm, the above matrix-vector multiply increases ICWY MGS complexity by mn^2 ($3mn^2$ total) but reduces global reductions from $j-1$ at iteration j to only one when combined with the lagged normalization of a Krylov vector due to Kim and Chronopoulos [23].

An alternative form exists for the correction matrix T, corresponding to the application of the elementary projectors $I - v_j\, v_j^T$ in forward followed by backward order, and is expressed as

$$T_{j-1} = (\, I - L^T_{j-1}\,)\,(\, I - L_{j-1}\,)$$

The forward-backward MGS algorithm is discussed in Leon et al. [18] and a backward-forward ordering of projectors was employed by Thomas et al. [24] in the context of a mixed-precision implementation of MGS-GMRES. It was also presented in the review article by Antz et al. [25]. This can be viewed as a truncated form of the Neumann series associated with a symmetric form for the correction matrix T_{j-1}

$$P = I - V_{j-1}\, T_{j-1}\, V_{j-1}^T, \quad T_{j-1} = (\, I + L_{j-1} + L_{j-1}^T \,)^{-1}$$

where

$$T_{j-1} = I - L_{j-1} - L_{j-1}^T + L_{j-1}^T\, L_{j-1} + L_{j-1}\, L_{j-1}^T$$

This series is generated by the block inverse of the symmetric matrix given below (see Appendix)

$$Q_{j-1}^T\, Q_{j-1} = \begin{bmatrix} T_{j-2}^{-1} & Q_{j-2}^T\, \mathbf{q}_{j-2} \\ \mathbf{q}_{j-2}^T\, Q_{j-2} & 1 \end{bmatrix}, \quad T_{j-2}^{-1} = I + L_{j-2} + L_{j-2}^T$$

Block generalizations of the MGS and CGS2 algorithm are presented in Carson et al. [26]. The authors present block algorithms with the more favorable communication patterns described herein. An analysis of the backward stability of the these block Gram-Schmidt algorithms is also presented. There exist several ways to implement a block MGS algorithm. The correction matrix T can be formed recursively from a block triangular inverse, (see Appendix) as in the compact CWY MGS derived by Björck [27]. Barlow [22] employs the inverse compact ICWY form of Puglusi [19]. However, a lagged normalization is not applied and two reductions are required. These are summarized in [3].

3. Loss of Orthogonality. When the Krylov vectors are orthogonalized via the finite precision MGS process, the loss of orthogonality is related in a straightforward way to the convergence of GMRES. In particular, orthogonality among the Krylov vectors is effectively maintained until the norm-wise relative error approaches the machine precision as discussed in Paige and Strakoš [9]. In this section, we prove a corollary to Paige et al. [1] that allows us to establish that $T = (I+L_k)^{-1} = I - L_k + \mathcal{O}(\varepsilon^2)\kappa^2(B)$, where $B = [\, \mathbf{r}_0, AV_k \,]$ for the inverse compact WY MGS formulation of the Arnoldi expansion.

Let A be an $n \times n$ real-valued matrix, and consider the Arnoldi factorization of the matrix B. After k steps, in exact arithmetic, the algorithm produces the factorization

$$AV_k = V_{k+1}\, H_{k+1,k}, \quad V_{k+1}^T V_{k+1} = I_{k+1}$$

where $H_{k+1,k}$ is an upper Hessenberg matrix. When applied to the linear system $A\mathbf{x} = \mathbf{b}$, assume $\mathbf{x}_0 = 0$, $\mathbf{r}_0 = \mathbf{b}$, $\|\mathbf{b}\|_2 = \rho$ and $\mathbf{v}_1 = b/\rho$. The Arnoldi algorithm produces an orthogonal basis for the Krylov vectors as columns of the matrix V_k

Consider the computed matrix $\bar{V}_k$ with Krylov vectors as columns. The strictly lower triangular matrix $\bar{L}_k$ is obtained from the loss of orthogonality relation, where $\bar{V}_k$ denotes the computed matrix in fixed-precision arithmetic

$$\bar{V}_k^T \bar{V}_k = I + \bar{L}_k + \bar{L}_k^T$$

Let us first consider the lower triangular solution algorithm for T where we can clearly identify the elements of L_k appearing in the correction matrix T, along with higher powers of the inner products in the Neumann series. In order to bound the Frobenius norm of the matrix $\bar{L}_k$, the inverse compact WY form of the MGS projector is obtained from the Householder transformation for an augmented matrix.

3.1. Householder and MGS Equivalence. Based on an idea from Sheffield, Björck and Paige [10] demonstrated that the QR factorization produced by the product of Householder reflectors, represented by a transformation matrix P applied to an augmented matrix containing an $m \times n$ matrix A and a zero block, is equivalent to the factorization produced by MGS applied to the augmented matrix (backward error and loss of orthogonality)

$$P^T \begin{bmatrix} 0 \\ A \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

and here we demonstrate that the inverse compact WY projector for MGS can be obtained directly from the $(2,2)$ block of P^T. The proof follows from Paige [28]. We note that Barlow (2019) employs the

augmented form and P^T – but then Barlow does not employ the inverse, and instead uses the Schreiber and Van Loan [21] compact WY recursive form of the triangular matrix T for the MGS projector P.

From Paige et al. [1], the augmented and computed $\tilde{P}_k$ is

$$\tilde{P}_k = \begin{bmatrix} \tilde{S}_k & (I - \tilde{S}_k)\tilde{V}_k^T \\ \tilde{V}_k(I - \tilde{S}_k) & I - \tilde{V}_k(I - \tilde{S}_k)\tilde{V}_k^T \end{bmatrix}$$

By forming the matrix product $\tilde{P}_k\tilde{P}_k^T = I_k$, the authors prove the following identities in equations (5.4) and (5.5) of their seminal paper [1]

$$\begin{aligned} (\, I - \tilde{S}_k \,)\, \tilde{V}_k^T\tilde{V}_k (\, I - \tilde{S}_k \,)^T &= I - \tilde{S}_k\tilde{S}_k^T \\ &= (\, I - \tilde{S}_k \,)\,(\, I - \tilde{S}_k \,)^T + (\, I - \tilde{S}_k \,)\,\tilde{S}_k^T + \tilde{S}_k\,(\, I - \tilde{S}_k \,)^T \\ \tilde{V}_k^T\tilde{V}_k &= I_k + \tilde{S}_k^T(\, I - \tilde{S}_k \,)^{-T} + (\, I - \tilde{S}_k \,)^{-1}\,\tilde{S}_k \end{aligned}$$

Thus, the strictly upper triangular part of $\tilde{V}_k^T\tilde{V}_k$ is given by

$$\tilde{U}_k = (\, I - \tilde{S}_k \,)^{-1}\,\tilde{S}_k = \tilde{S}_k\,(\, I - \tilde{S}_k \,)^{-1}$$

For the augmented Householder matrix, $\tilde{P}^T$ is applied, and within the (2,2) block the MGS projector is identified as, $\tilde{P} = I - \tilde{V}_k(I - \tilde{S}_k)^T\tilde{V}_k^T$ and from Paige [28], eqn (2.2) and Corollary 5.1, it follows that $\tilde{L}_k = \tilde{U}_k^T$,

$$\begin{aligned} I - \tilde{S}_k^T &= I - \tilde{U}_k^T\,(\, I + \tilde{U}_k^T \,)^{-1} \\ &= I - \tilde{L}_k\,(\, I + \tilde{L}_k \,)^{-1} \\ &= I - \tilde{L}_k\,(\, I - \tilde{L}_k + \tilde{L}_k^2 - \tilde{L}_k^3 + \cdots \,) \\ &= I - \tilde{L}_k + \tilde{L}_k^2 - \tilde{L}_k^3 + \cdots \\ &= (I + \tilde{L}_k)^{-1} \end{aligned}$$

Therefore, the augmented Householder transformation matrix naturally produces an ICWY MGS projector, given by:

$$\tilde{P} = I - \tilde{V}_k(I + \tilde{L}_k)^{-1}\tilde{V}_k^T$$

with the lower triangular matrix $T = (I + \tilde{L}_k)^{-1}$. This MGS projector has previously appeared in the work of Paige and Wülling [29], on the bottom of their page 4, where they note it was first derived by Giraud et al. [5]. This represents a constructive proof for the existence of the ICWY MGS, and then from Björck [30], the lower triangular T is equivalent to the product of elementary projectors (applied in forward order, $j = 1, 2, \ldots k$) for the $A = QR$ factorization via modified Gram-Schmidt orthogonalization,

$$P = I - Q_k\,T\,Q_k^T = (I - \mathbf{q}_k\mathbf{q}_k^T)\cdots(I - \mathbf{q}_1\mathbf{q}_1^T)$$

For the block forms of MGS, Barlow [22] employs the augmented and upper triangular matrix T based on the compact WY representation of the projector with a backward ordering $j = k, \ldots, 1$ of the rank-1 projectors as presented in Malard and Paige [20]. The forward ordering is then obtained by using the transposed T^T as was derived by (Björck [27], section 7). This requires matrix-matrix multiplies, related to the recursive form of block triangular matrix inverses (see appendix). Thus, a triangular solve is avoided and the computation remains backward stable.

3.2. Truncated Neumann series. A corollary to the MGS-GMRES backward stability results from Paige and Strakos [9], and Paige et al. [1] is now established, namely that the Neumann series for T may be truncated according to

$$T = (\, I + \tilde{L}_k \,)^{-1} = I - \tilde{L}_k + \mathcal{O}(\varepsilon^2)\kappa^2(B)$$

The essential results will be based on the MGS factorization of the matrix

$$B = [\, \mathbf{r}_0,\, AV_k \,] = V_{k+1}\,[\, \mathbf{e}_1\rho_0,\, H_{k+1,k} \,]$$

From Paige and Strakos [9], they derive a bound on the loss-of-orthogonality given by

$$\| I - \tilde{V}_{k+1}^T \tilde{V}_{k+1} \|_F \leq \kappa([\mathbf{r}_0, \ AV_k]) \, \mathcal{O}(\varepsilon)$$

The derivation of this result should allow us to find an upper bound for $\|\tilde{L}_k\|_F^p$

From equation (5.5) of [6], the upper triangular structure of the loss of orthogonality relation is revealed to be

$$\tilde{U}_m = (I - \tilde{S}_m)^{-1} \tilde{S}_m = \tilde{S}_m (I - \tilde{S}_m)^{-1}$$

where $\tilde{U}_m = \tilde{L}_m^T$ is the strictly upper triangular (sut) part of $\tilde{V}^T\tilde{V}$. It then follows immediately, from equation (5.6) of [6] that a bound on the loss of orthogonality is given by

$$\sqrt{2}\|\tilde{V}_m^T \tilde{v}_m\|_2 \leq \|I - \tilde{V}_m^T \tilde{V}_m\|_2 = \sqrt{2}\|(I - \tilde{S}_m)^{-1} \tilde{S}_m \|_F \leq \frac{4}{3}(2m)^{1/2} \hat{\gamma}_n \tilde{\kappa}_F(B)$$

where

$$\hat{\gamma}_n = \frac{\tilde{c}_n \varepsilon}{1 - \tilde{c}_n \varepsilon}, \quad \tilde{\kappa}_F(B) = \min_{\mathrm{diag} D > 0} \|AD\|_F / \sigma_{\min}(AD)$$

The matrix D is defined in [1] to be *any* positive definite diagonal matrix. Therefore, it follows that

$$\|\tilde{U}_k\|_F = \|\tilde{L}_k^T\|_F \leq \mathcal{O}(\varepsilon)\tilde{\kappa}_F(B), \quad \|\tilde{L}_k\|_F^p \leq \mathcal{O}(\varepsilon^p)\tilde{\kappa}_F^p(B)$$

and thus the matrix inverse from the Neuman series, is expressed as

$$\begin{aligned} T = (I + \tilde{L}_k)^{-1} &= I - \tilde{L}_k + \tilde{L}_k^2 - \tilde{L}_k^3 + \cdots \\ &= I - \tilde{L}_k + \mathcal{O}(\varepsilon^2)\kappa^2(B) \end{aligned}$$

The growth of the condition number above is related to the norm-wise relative backward error

$$\beta(\mathbf{x}_k) = \frac{\|\mathbf{r}_k\|_2}{\|\mathbf{b}\|_2 + \|A\|\|\mathbf{x}_k\|_2}$$

and in particular, Paige and Strakos [9] prove that

$$\beta(\mathbf{x}_k)\kappa\left([\mathbf{r}_0, \ AV_k]\right) = \mathcal{O}(1)$$

This implies that all powers of $\|\tilde{L}_k\|_F^p$ remain $\mathcal{O}(\varepsilon)$ until convergence of MGS-GMRES when $\|\tilde{S}_k\|_2 \to 1$. This matrix 2-norm can be used to track the loss of orthogonality as presented by Paige [6].

4. Low-synchronization MGS-GMRES. The MGS–GMRES orthogonalization algorithm can be viewed as the Gram-Schmidt QR factorization of a matrix B formed by adding a new column to V_m in each iteration

$$\left[\ r_0, \quad AV_m \ \right] = V_{m+1} \left[\ \|r_0\|e_1, \quad H_{m+1,m} \ \right]$$

The algorithm was proven to be backward stable for the solution of linear systems $A\mathbf{x} = b$ in [1] and orthogonality is maintained to $\mathcal{O}(\varepsilon)\kappa(B)$, depending upon the condition number of the matrix $B = \kappa([\mathbf{r}_0, \ AV_m])$. For the inverse compact WY MGS-GMRES algorithm the normalization of the Krylov vector $\mathbf{v}_{i+1}$ at iteration $i+1$ represents the delayed scaling of the vector $\mathbf{v}_{i+2}$ in the matrix-vector product $\mathbf{v}_{i+2} = A\mathbf{v}_{i+1}$. Therefore, an additional Step 8 is required in the one-reduce algorithm, $\mathbf{r}_{1:i+1,i+2} = \mathbf{r}_{1:i+1,i+2}/\mathbf{r}_{i+1,i+1}$ and $\mathbf{v}_{i+1} = \mathbf{v}_{i+1}/\mathbf{r}_{i+1,i+1}$. The diagonal element of the R matrix in the Arnoldi QR factorization of the matrix B corresponding to H_i, is updated after the MGS projection in Step 12.

Given the form for the ICWY MGS projector derived above, the one-reduce MGS-GMRES algorithm presented in Swirydowicz etal. [3] can be modified, with the inverse matrix replaced by the correction matrix $T = (I - L_{i+i})$ in Step 11 of the algorithm.

Algorithm 4.1 Truncated Neumann Series MGS-GMRES

Input: Matrix A; right-hand side vector $\mathbf{b}$; initial guess vector $\mathbf{x}_0$
Output: Solution vector $\mathbf{x}$

$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, $\mathbf{v}_1 = \mathbf{r}_0$.
$\mathbf{v}_2 = A\mathbf{v}_1$
$(\, V_2,\, R,\, L_2\,) = \mathrm{mgs}\,(\, V_2,\, R,\, L_1\,)$
for $i = 1, 2, \ldots$ **do**
    $\mathbf{v}_{i+2} = A\mathbf{v}_{i+1}$ ▷ Matrix-vector product
    $[\, L^T_{:,i+1},\, \mathbf{r}_{i+2}\,] = V^T_{i+1}[\mathbf{v}_{i+1}\, \mathbf{v}_{i+2}]$ ▷ Global synchronization
    $\mathbf{r}_{i+1,i+1} = \|\mathbf{v}_{i+1}\|_2$
    $\mathbf{v}_{i+1} = \mathbf{v}_{i+1}/\mathbf{r}_{i+1,i+1}$ ▷ Lagged normalization
    $\mathbf{r}_{1:i+1,i+2} = \mathbf{r}_{1:i+1,i+2}/\mathbf{r}_{i+1,i+1}$ ▷ Scale for Arnoldi
    $L^T_{:,i+1} = L^T_{:,i+1}/\mathbf{r}_{i+1,i+1}$
    $\mathbf{r}_{1:i+1,i+2} = T_{i+1}\, \mathbf{r}_{1:i+1,i+2}$ ▷ Projection Step
    $\mathbf{v}_{i+2} = \mathbf{v}_{i+2} - V_{i+1}\, \mathbf{r}_{1:i+1,i+2}$
    $H_i = \mathbf{r}_{i+1}$
    Apply Givens rotations to H_i
end for
$\mathbf{y}_m = \mathrm{argmin}\|(\, H_m\mathbf{y}_m - \|\mathbf{r}_0\|_2\mathbf{e}_1\,)\|_2$
$x = \mathbf{x}_0 + V_m\mathbf{y}_m$

4.1. Backward Error and Stopping Criteria. The stopping criteria for MGS-GMRES is an important consideration and is related to backward error for solving linear systems $A\mathbf{x} = b$. The most common convergence criterion found in existing iterative solver frameworks is based upon the relative residual, defined by

$$\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{b}\|_2} = \frac{\|\mathbf{b} - A\mathbf{x}_k\|_2}{\|\mathbf{b}\|_2} < tol \tag{4.1}$$

However, when the columns of V_k become linearly dependent, as indicated by $\|S_k\|_2 = 1$, the orthogonality of the Krylov vectors is completely lost. Then the convergence of MGS-GMRES flattens or stalls at this iteration.

Due to the relationship with the backward error for solving linear systems $Ax = b$, elucidated by Prager and Oettli [31] and Rigal and Gaches [32]. The backward stability anlysis of Paige et al. [1] relies instead upon the norm-wise relative backwards error (NRBE) reaching machine precision as the orthogonality is lost

$$\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{b}\|_2 + \|A\|_\infty\|\mathbf{x}\|_2} \approx \mathcal{O}(\varepsilon) \tag{4.2}$$

which is achieved when $\|S\|_2 = 1$. This metric is also commonly applied to assess the backward error for direct solvers.

For a sufficiently non-singular matrix

$$\sigma_{\min}(A) \gg n^2\varepsilon\|A\|_F$$

one can employ MGS-GMRES to solve $A\mathbf{x} = \mathbf{b}$ with the NBRE stopping criterion.

4.2. Numerical Experiments. In order to demonstrate that the Neumann series MGS-GMRES algorithm maintains the same convergence history as the original MGS-GMRES algorithm of Saad and Schultz [2], two systems of equations are solved without using a preconditioner. In all experiments $\mathbf{b} = \mathbf{e} = (\,1, \ldots, 1)^T$. The first system of equations has the coefficient matrix A given by `impcol_e` from the matrix-market collection maintained by the national institute of standards (NIST). This problem was studied by Greenbaum et al. [8] in order to analyze the convergence of GMRES. Both the relative residual (4.1) and norm-wise relative backward error (4.2) are compared along with the loss of orthogonality as

measured by $\|S\|_2$ We note that relative residual and norm-wise backward error at convergence are lower than those achieved by the original MGS-GMRES algorithm.

The second linear system is based on the matrix `F_183_6`, as studied by Paige and Strakoš [9]. For the matrix `F_183_6`, the dimension is $n = 183$, $\|A\|_2 = 1.2 \times 10^9$, and $\kappa(A) \approx 1.5 \times 10^{11}$. The convergence history is plotted in Figure 2 and achieves a lower NRBE when the symmetric form of the correction matrix T is employed. In addition, the loss of orthogonality is more gradual. This form of the T matrix leads to the lowest NRBE for highly non-normal and extremely ill-conditioned coefficient matrices A such as `F_183_6`. Therefore, these empirical results suggest that the truncated Neumann series and backward-forward T variants of MGS-GMRES can improve upon the loss of orthogonality and norm-wise backward error of the original algorithm.

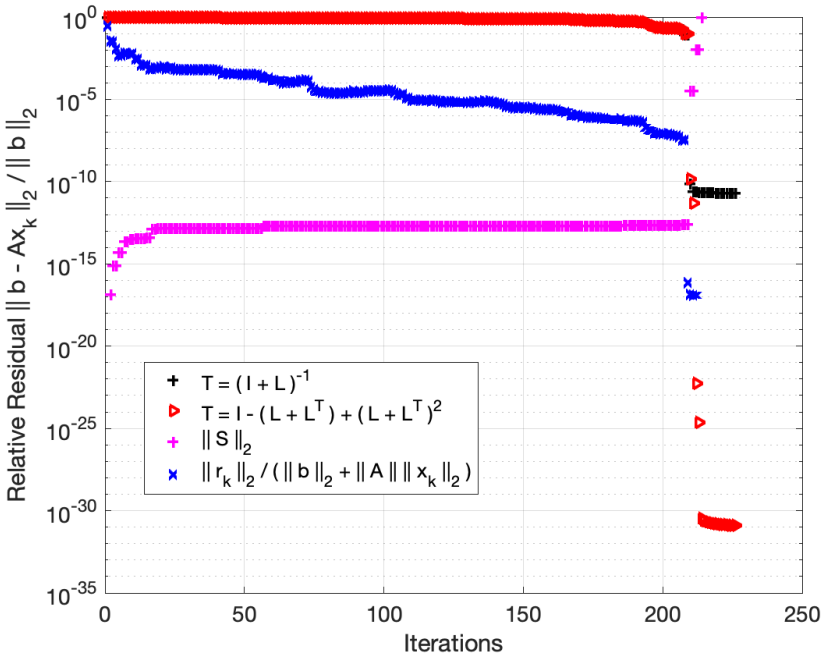

Fig. 1: Greenbaum, Rozložnik and Strakoš (1997). `impcol_e` matrix. Convergence history.

5. Algebraic Multigrid Preconditioner. An overview of algebraic multigrid (AMG) is now provided using the BoomerAMG library in Hypre. As discussed in [33], this is a particularly powerful method for solving challenging systems of linear equations. AMG solvers [4, 34, 35] are also efficient preconditioners for Krylov iterations applied to large-scale linear systems arising in physics-based simulations due to their optimal complexity, rapid convergence and robustness [36].

AMG can in theory solve a linear system with n unknowns in $\mathcal{O}(n)$ operations. However, it is now common practice to instead use AMG as a preconditioner to a Krylov method, even though it was originally developed as a solver. An AMG method accelerates the solution of a linear system

$$Ax = b \tag{5.1}$$

through error reduction by using a sequence of coarser matrices called a *hierarchy*. We will refer to the sequence of matrices as A_k, where $k = 0 \ldots m$, and A_0 is the matrix from (5.1). Each A_k has dimensions $m_k \times m_k$ where $m_k > m_{k+1}$ for $k < m$. For the purposes of this paper, we will assume that

$$A_k = R_k A_{k-1} P_k \,, \tag{5.2}$$

for $k > 0$, where P_k is a rectangular matrix with dimensions $m_{k-1} \times m_k$. P_k is referred to as a *prolongation matrix* or *prolongator*. R_k is the *restriction matrix* and $R_k = P_k^T$ in the Galerkin formulation of AMG.

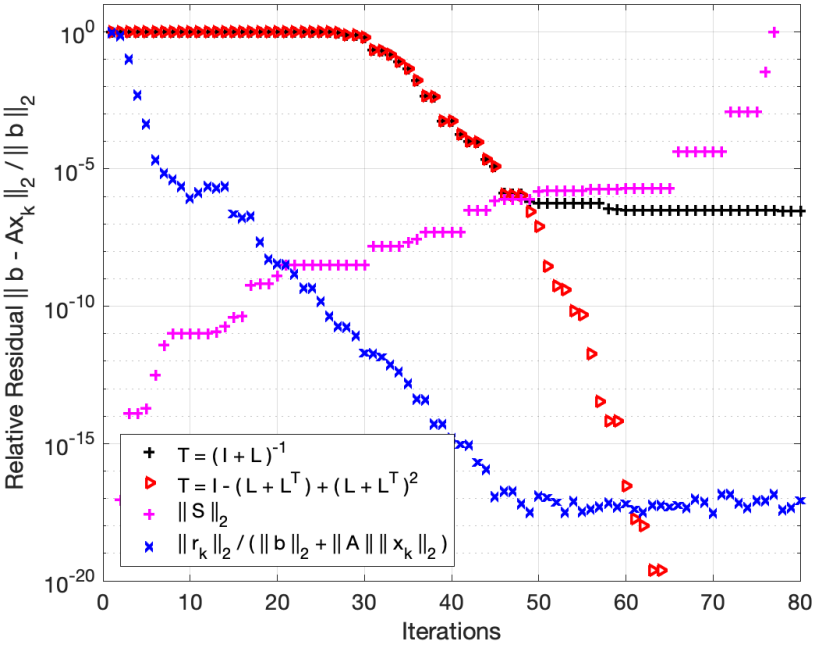

Fig. 2: Paige and Strakoš (2002). `f_183_6` matrix. Convergence history.

Associated with each A_k, $k < m$, is a solver called a *smoother*, which is usually an inexpensive iterative method, e.g., Gauss-Seidel, polynomial, or incomplete factorization. The *coarse solver* used for A_m (the lowest level in the hierarchy) is often a direct solver, although it may be an iterative method if A_m is singular.

The setup phase of AMG is nontrivial for several reasons. Each prolongator P_k is developed algebraically from A_{k-1} (hence the name of the method). Once the transfer matrices are determined, the coarse-matrix representations are recursively computed from A through sparse matrix-matrix multiplication.

5.1. Ruge–Stüben Classical AMG. We now give a brief overview of classic Ruge-Stüben AMG, starting with some notation that will be used in the subsequent discussions. Point j is a neighbor of i if and only if there is a non-zero element a_{ij} of the matrix A. Point j strongly influences i if and only if

$$|a_{ij}| \geq \theta \max_{k \neq i} |a_{ik}|, \tag{5.3}$$

where θ is the strength of connection threshold, $0 < \theta \leq 1$. This strong influence relation is used to select coarse points. The selected coarse points are retained in the next coarser level, and the remaining fine points are dropped. Let C_k and F_k be the coarse and fine points selected at level k, and let m_k be the number of grid points at level k ($m_0 = n$). Then, $m_k = |C_k| + |F_k|$, $m_{k+1} = |C_k|$, A_k is a $m_k \times m_k$ matrix, and P_k is a $m_{k-1} \times m_k$ matrix. Here, the coarsening is performed row-wise by interpolating between coarse and fine points. The coarsening generally attempts to fulfill two contradictory criteria. In order to ensure that a chosen interpolation scheme is well-defined and of good quality, some close neighborhood of each fine point must contain a sufficient amount of coarse points to interpolate from. Hence the set of coarse points must be rich enough. However, the set of coarse points should be sufficiently small in order to achieve a reasonable coarsening rate. The interpolation should lead to a reduction of roughly five times the number of non-zeros at each level of the V-cycle.

There has been extensive research on different variants of AMG since the development of the first AMG algorithms [4]. The original Ruge-Stüben interpolation based approach is now referred to as

classical AMG. One of the drawbacks with the C-AMG method is that, despite rapid convergence, it often generates excessive operator complexities, in particular for three-dimensional problems. This issue is exacerbated for parallel implementations of AMG when the coarsening algorithm is applied within an MPI rank (sub-domain) and the smoother is local. At the interface between MPI ranks the smoother is a point-wise Jacobi iteration. Consequently, efforts were made to coarsen more aggressively to reduce operator complexities [37]. More aggressive coarsening leads to often considerably reduced convergence rates because it violates conditions required for classical interpolation. Convergence can be improved again by combining more aggressive coarsening with long-distance interpolation [38]. Non-Galerkin coarse grids, for which the coarse level operator does not satisfy the relationship $A_k = P_k^T A_{k-1} P_k$ for each level k, were introduced in [39].

5.2. AMG Setup. During the setup phase of AMG methods, a multilevel V-cycle hierarchy is constructed that consists of linear systems with exponentially decreasing sizes on coarser levels. A strength-of-connection matrix S, is typically first computed to indicate directions of algebraic smoothness and applied in the coarsening algorithms. The construction of S may be performed efficiently on GPUs, because each row of S is computed independently by selecting entries in the corresponding row of A with a prescribed threshold value θ. BoomerAMG currently provides the parallel maximal independent set (PMIS) coarsening [40] on GPUs, which is a modified from Luby's algorithm [41] for finding maximal independent sets using random numbers. The process of selecting coarse points in this algorithm is massively parallel, which makes it appropriate for GPUs.

Interpolation operators in AMG transfer residual errors between adjacent levels. There are a variety of interpolation schemes available in BoomerAMG on CPUs. Direct interpolation [35] is straightforward to implement on GPUs because the interpolatory set of a fine point i is just a subset of the neighbors of i, and thus the interpolation weights can be determined solely by the i-th equation. A bootstrap AMG (BAMG) [42] variant of direct interpolation is generally found to be better than the original formula. The weights $\mathbf{w}_{ij}$ are computed by solving the local optimization problem

$$\min \|a_{ii}\mathbf{w}_i^T + a_{i,C_i^s}\|_2 \quad \text{s.t.} \quad \mathbf{w}_i^T \mathbf{f}_{C_i^s} = \mathbf{f}_i,$$

where $\mathbf{w}_i$ is a vector that contains $\mathbf{w}_{ij}$, C_i^s and denotes strong C-neighbors of i and $\mathbf{f}$ is a target vector that needs to be interpolated exactly. For elliptic problems where the near null-space is spanned by constant vectors, i.e., $\mathbf{f} = \mathbf{1}$, the closed-form solution of (5.2) is given by

$$\mathbf{w}_{ij} = -\frac{a_{ij} + \beta_i / n_{C_i^s}}{a_{ii} + \sum_{k \in N_i^w} a_{ik}}, \quad \beta_i = \sum_{k \in \{\mathbf{f}_i \cup C_i^w\}} a_{ik} \ , \tag{5.4}$$

where $n_{C_i^s}$ denotes the number of points in C_i^s, C_i^w the weak C-neighbors of i, $\mathbf{f}_i$ the F-neighbors, and N_i^w the weak neighbors. A known issue of PMIS coarsening is that it can result in F-points without C-neighbors [43]. In such situations, distance-one interpolation algorithms often do work well, whereas interpolation operators that can reach C-points at a larger range, such as the extended interpolation [43], can generally yield much better convergence. However, implementing extended interpolation is much more complicated mainly due to the fact that the sparsity pattern of the interpolation operator cannot be determined a priori, which requires dynamically combining C-points in a distance-2 neighborhood, and furthermore, efficient implementation can be even more difficult on GPUs. With minor modifications to the original form, it turns out that the extended interpolation operator can be rewritten by using standard sparse matrix computations such as matrix-matrix (M-M) multiplication and diagonal scaling with certain FF- and FC-sub-matrices. The coarse-fine C-F splitting of the coarse matrix A is given by

$$A = \begin{bmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{bmatrix}$$

where A is assumed to be decomposed into $A = D + A^s + A^w$, the diagonal, the strong part and weak part respectively, and A_{FF}^w, A_{FC}^w, A_{FF}^s and A_{FC}^s are the corresponding submatrices of A^w and A^s.

The full prolongation operator is given by

$$P = \begin{bmatrix} W \\ I \end{bmatrix}$$

The extended "MM-ext" interpolation takes the form

$$W = -\left[(D_{FF} + D_\gamma)^{-1}(A^s_{FF} + D_\beta)\right]\left[D_\beta^{-1} A^s_{FC}\right]$$

with

$$D_\beta = \mathrm{diag}(A^s_{FC}\mathbf{1}_C) \quad D_\gamma = \mathrm{diag}(A^{\mathbf{w}}_{FF}\mathbf{1}_F + A^{\mathbf{w}}_{FC}\mathbf{1}_C),$$

This formulation allows simple and efficient implementations that can utilize available optimized sparse kernels on GPUs. Similar approaches that are referred to as "MM-ext+i" modified from the original extended+i algorithm [43] and "MM-ext+e" are also available in BoomerAMG. See [44] for details on the class of M-M based interpolation operators.

Aggressive coarsening reduces the grid and operator complexities of the AMG hierarchy, where a second coarsening is applied to the C-points obtained from the first coarsening to produce a smaller set of final C-points. The A-1 aggressive coarsening strategy described in [35] is employed in our studies. The second PMIS coarsening is performed with the CC block of $S^{(A)} = S^2 + S$ that has nonzero entry $S^{(A)}_{ij}$ if i is connected to j with at least a path of length less than or equal to two. Aggressive coarsening is usually used with two-stage interpolation [45] which computes a second-stage interpolation matrix P_2 and combined with the first-stage P_1 as $P = P_1 P_2$. The aforementioned MM-based interpolation is also available for the second stage.

Finally, Galerkin triple-matrix products are used to build coarse-level matrices $A_c = P^T A P$ involving the prolongation P and restriction P^T operators. This computation is performed using parallel primitives from Thrust and routines from cuSPARSE or hypre's own sparse kernels. We refer to [46] for the details omitted here on the algorithms employed in hypre for computing distributed sparse M-M multiplications on GPUs.

5.3. Two–Stage Inner–Outer Gauss–Seidel Iteration. To solve a linear system $A\mathbf{x} = \mathbf{b}$, the Gauss-Seidel (GS) iteration is based on the matrix splitting $A = L + D + U$, where L and U are the strictly lower and upper triangular parts of the matrix A, respectively. Then, the traditional GS updates the solution based on the following recurrence,

$$\mathbf{x}_{k+1} := \mathbf{x}_k + M^{-1}\mathbf{r}_k, \quad k = 0, 1, 2, \ldots \tag{5.5}$$

where $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, $A = M - N$, and $M = L + D$, $N = -U$ or $M = U + D$, $N = -L$ for the forward or backward sweeps, respectively. In the following, $\mathbf{x}_k$ is the k–th iterate. To avoid explicitly forming the matrix inverse M^{-1} in (5.5), a sparse-triangular solve is used to apply M^{-1} to the current residual vector $\mathbf{r}_k$.

To improve the solver scalability, hypre implements a hybrid variant of Gauss-Seidel [47], where the neighboring processes first exchange the elements of the solution vector on the boundary, but then each process independently applies the local relaxation. Furthermore, in hypre, each process may apply multiple local GS sweeps for each round of the neighborhood communication. With this approach, each local relaxation updates only the local part of the vector $\mathbf{x}_{k+1}$ (during the local relaxation, the non-local solution elements on the boundary are not kept consistent among the neighboring processes). This hybrid algorithm is shown to be effective for many problems [47].

A two-stage Gauss-Seidel relaxation employs a fixed number of "inner" stationary iterations for approximately solving the triangular system with M,

$$\widehat{\mathbf{x}}_{k+1} := \widehat{\mathbf{x}}_k + \widehat{M}^{-1}(\mathbf{b} - A\widehat{\mathbf{x}}_k), \quad k = 0, 1, 2, \ldots \tag{5.6}$$

where $\widehat{M}^{-1}$ represents the approximate triangular system solution, i.e., $\widehat{M}^{-1} \approx M^{-1}$. A Jacobi-Richardson (or Jacobi) inner iteration (further abbreviated as JR) is employed. In particular, if $\mathbf{g}^{(j)}_k$ denotes the approximate solution from the j-th inner iteration at the k-th outer GS iteration, then the initial solution is chosen to be the diagonally scaled residual vector,

$$\mathbf{g}^{(0)}_k = D^{-1}\mathbf{r}_k, \tag{5.7}$$

and the $(j+1)$–st JR iteration computes the approximate solution by the recurrence

$$\mathbf{g}^{(j+1)}_k := \mathbf{g}^{(j)}_k + D^{-1}(\mathbf{r}_k - (L + D)\mathbf{g}^{(j)}_k) \tag{5.8}$$

$$= D^{-1}(\mathbf{r}_k - L\mathbf{g}^{(j)}_k). \tag{5.9}$$

When "zero" inner sweeps are performed, the two-stage GS recurrence becomes

$$\widehat{\mathbf{x}}_{k+1} := \widehat{\mathbf{x}}_k + \mathbf{g}_k^{(0)} = \widehat{\mathbf{x}}_k + D^{-1}(\mathbf{b} - A\widehat{\mathbf{x}}_k),$$

and this special case corresponds to Jacobi-Richardson for the global system, or local system on each process. When s inner iterations are performed, it follows that

$$\begin{aligned}\widehat{\mathbf{x}}_{k+1} := \widehat{\mathbf{x}}_k + \mathbf{g}_k^{(s)} &= \widehat{\mathbf{x}}_k + \sum_{j=0}^{s}(-D^{-1}L)^j D^{-1}\widehat{\mathbf{r}}_k \\ &\approx \widehat{\mathbf{x}}_k + (I + D^{-1}L)^{-1}D^{-1}\widehat{\mathbf{r}}_k = \widehat{\mathbf{x}}_k + M^{-1}\widehat{\mathbf{r}}_k,\end{aligned}$$

where M^{-1} is approximated by the degree-s Neumann expansion. Note that $D^{-1}L$ is strictly lower triangular so that the Neumann series converges in a finite number of steps. The two-stage GS recurrence (5.6) can be also written as

$$\widehat{\mathbf{x}}_{k+1} := \widehat{\mathbf{x}}_k + \widehat{M}^{-1}(\mathbf{b} - (M - N)\widehat{\mathbf{x}}_k) \tag{5.10}$$

$$= (I - \widehat{M}^{-1}M)\widehat{\mathbf{x}}_k + \widehat{M}^{-1}(\mathbf{b} + N\widehat{\mathbf{x}}_k). \tag{5.11}$$

In the classical one-stage recurrence (5.5), the preconditioner matrix is taken as $\widehat{M}^{-1} = M^{-1}$, and only the second term remains in the recurrence (5.11), leading to the following "compact" form,

$$\mathbf{x}_{k+1} := M^{-1}(\mathbf{b} + N\mathbf{x}_k). \tag{5.12}$$

Hence, the recurrences (5.5) and (5.12) are mathematically equivalent, while the recurrence (5.12) has a lower computation cost.

A similar "compact" recurrence for the two-stage algorithm can be derived as

$$\widetilde{\mathbf{x}}_{k+1} := \widehat{M}^{-1}(\mathbf{b} + N\widetilde{\mathbf{x}}_k). \tag{5.13}$$

However, with the approximate solve using $\widehat{M}^{-1}$, the recurrences (5.6) and (5.11) are no longer equivalent. For example, even if it is assumed that $\widetilde{\mathbf{x}}_k = \widehat{\mathbf{x}}_k$, comparing (5.5) and (5.6), the difference in the residual norms using the classical and the standard two-stage iterations is given by

$$\begin{aligned}\|\widetilde{\mathbf{r}}_{k+1} - \mathbf{r}_{k+1}\| &= \|A(I - \widehat{M}^{-1}M)M^{-1}\mathbf{r}_k\| \\ &\le \|A(I - \widehat{M}^{-1}M)\|\|M^{-1}\mathbf{r}_k\|,\end{aligned} \tag{5.14}$$

while comparing (5.5) and (5.11), the difference between the classical and the compact two-stage iterations is

$$\|\widehat{\mathbf{r}}_{k+1} - \mathbf{r}_{k+1}\| = \|A(I - \widehat{M}^{-1}M)(M^{-1}\mathbf{r}_k + \mathbf{x}_k)\|$$

$$\le \|A(I - \widehat{M}^{-1}M)\|\|M^{-1}\mathbf{r}_k\| \tag{5.15}$$

$$+ \|A(I - \widehat{M}^{-1}M)\|\|\mathbf{x}_k\| \tag{5.16}$$

and the compact form has the extra term with $\|\mathbf{x}_k\|$ in the bound. For the recurrence (5.13) to be as effective as the recurrence (5.11), we found that additional inner iterations are often required (to make $\|I - \widehat{M}^{-1}M\|$ small).

Two outer and two inner inner iterations often lead to rapid convergence in less than five preconditioned GMRES iterations for the momentum solver of the incompressible Navier-Stokes equations [33].

5.4. Coarse Matrix Splitting and Dropping Strategy. There are two important measures that determine the quality of an AMG algorithm. The first is the convergence factor, which indicates how fast the method converges. The second is the operator complexity, which affects the number of operations and the memory usage. Operator complexity C is defined as

$$C = \sum_{k=0}^{m} \mathrm{nnz}(A_k)/\mathrm{nnz}(A)\,. \tag{5.17}$$

where $nnz(A)$ is the number of non-zero elements of A. The complexity measure indicates the amount of memory required. To reduce memory utilization, the complexity C should remain small. The complexity is also an indicator of the number of operations per V-cycle in the solver, assuming an iterative smoother. Small operator complexities lead to small V-cycle times. The average stencil size $S(A_k)$ is the average number of non-zero elements per row of A_l. Even when the stencil sizes $S(A_k)$ of the original matrix are small, very large stencil sizes can occur on coarser levels. Large stencil sizes can lead to large setup times, even if the operator complexity is small. This is because the coarsening algorithms, and to some degree interpolation, visit neighbors of neighbors, resulting in super-linear or even quadratic growth in the number of operations when evaluating the coarse system or the interpolation matrix. Large stencil sizes can also increase parallel-communication cost, because they may require the exchange of larger sets of data. Both convergence factors and complexities need to be considered when evaluating coarsening and interpolation algorithms, as they often influence each other. Higher complexities can improve convergence, and lower complexities lead to degradation in convergence rates. A degradation in convergence rate due to lower complexity often can be overcome by Krylov methods such as GMRES.

In this study, a dropping strategy is proposed to further reduce the amount of computation within the two-stage inner-outer Gauss-Seidel smoothers, without leading to a degradation of the GMRES-AMG solver convergence rate. Given the matrix splitting $A = D + L + U$, at every level of the V-cycle, dropping is applied according to the Matlab statement `L = (abs(L) > tol) .* L`. Consequently, we expect that the operator complexity C, for the L matrix, can be reduced substantially with the drop in $nnz(L)$.

5.5. Hypre-BoomerAMG. To study the performance of the two-stage Gauss-Seidel preconditioner and smoother in a practical setting, incompressible fluid flow simulations were performed with Nalu-Wind [33]. This is the primary fluid mechanics code for the ExaWind project, one of the application projects chosen for the DOE Exascale Computing Project (ECP) and is used for high-fidelity simulations of air flow dynamics around wind turbines. Nalu-Wind solves the acoustically incompressible Navier-Stokes equations, where mass continuity is maintained by an approximate pressure projection scheme. The governing physics equations for momentum, pressure, and scalar quantities are discretized in time with a second-order BDF-2 integrator, where an outer Picard fixed-point iteration is employed to reduce the nonlinear system residual at each time step. Within each time step, the Nalu-Wind simulation time is often dominated by the time required to setup and solve the linearized governing equations, using hypre-BoomerAMG. To solve the momentum equations, Nalu-Wind typically employs Gauss-Seidel (GS) or symmetric Gauss-Seidel (SGS) iteration as a preconditioner to accelerate GMRES convergence. The pressure systems are solved using GMRES with an algebraic multigrid (AMG) preconditioner, where a Gauss-Seidel smoother is applied to relax or remove high energy components of the solution error (e.g. those associated with the large eigenvalues of the system), which the coarse-grid solver fails to eliminate. Hence, Gauss-Seidel iteration is a compute time intensive component, employed either as a stand-alone preconditioner for the Krylov solver or as a smoother within an AMG V-cycle.

The Nalu-Wind time integrator employs the one-reduce MGS-GMRES linear solver for the momentum and pressure-Poisson solvers for the incompressible Navier-Stokes governing equations as described in [33] and [3]. The one-reduce solver described herein has been implemented as a part of hypre. The momentum solver is preconditioned with a two-stage Gauss-Seidel relaxation scheme as described previously and in the recent work by Thomas et al. [15]. The pressure-Poisson preconditioner is based on an AMG algorithm using aggressive PMIS coarsening at the first two levels combined with the matrix-based approach for the second-stage interpolation.

The McAlister wind tunnel experiment for wind-turbine blades is an unsteady RANS simulation of a fixed-wing, with a NACA0015 cross section, operating in uniform inflow and was chosen as a problem to evaluate the performance of our hypre linear solver. Resolving the high-Reynolds number boundary layer over the wing surface requires resolutions of $\mathcal{O}(10^{-5})$ normal to the surface resulting in cell aspect ratios of $\mathcal{O}(40,000)$. These high aspect ratios, coupled with the loss of diagonal dominance in the momentum system, present a significant challenge for the iterative solvers. Overset meshes were employed to generate body-fitted meshes for the wing and the wind tunnel geometry. The simulations were performed for a wing at 12 degree angle of attack, a 1 m chord length, denoted c, 3.3 aspect ratio, i.e., $s = 3.3c$, and a square wing tip. The inflow velocity is $u_\infty = 46$ m/s, the density is $\rho_\infty = 1.225$ kg/m^3, and the dynamic

viscosity is $\mu = 3.756 \times 10^{-5}$ kg/(m s), leading to a Reynolds number, $Re = 1.5 \times 10^6$. Wall normal resolutions were chosen to adequately represent the boundary layers on both the wing and tunnel walls. The $k - \omega$ SST RANS turbulence model was employed for the simulations. Due to the complexity of mesh generation, only one mesh with approximately 3 million grid points was generated.

Coarsening for Hypre-BoomerAMG is based on the parallel maximal independent set (PMIS) algorithm of Luby [40, 41, 43] allowing for a parallel setup phase. A transposed prolongation operator is retained for triple-matrix RAP products. The strength of connection threshold is set to $\theta = 0.25$. Aggressive coarsening is applied on the first two V-cycle levels with multi-pass interpolation and a stencil width of two elements per row. The remaining levels employ M-M extended+i interpolation. with truncation level 0.25 together with a maximum stencil width of two matrix elements per row. Sparsification techniques with non-Galerkin operators and drop tolerances applied to specific levels were introduced in [48] to reduce the complexity C. The truncation is applied on the first three levels with drop tolerances $\gamma = [\,0.0,\ 0.01,\ 0.01\,]$. The smoother is two sweeps of the inner-outer Gauss-Seidel iteration with two inner sweeps. The Hypre-BoomerAMG smoother is hybrid with the two-stage Gauss-Seidel applied locally and then Jacobi smoothing for globally shared degrees of freedom. The coarsening rate for the wing simulation is roughly $4\times$ with eight levels in the V-cycle for Hypre. Operator complexity C is close to 1.1 indicating more efficient V-cycles with aggressive coarsening, however, an increased number of (restarted) GMRES iterations are required compared to standard coarsening. The comparison among l_1-Jacobi, hybrid Gauss-Seidel and the proposed two-stage hybrid Gauss-Seidel smoothers is shown in Figure 3.

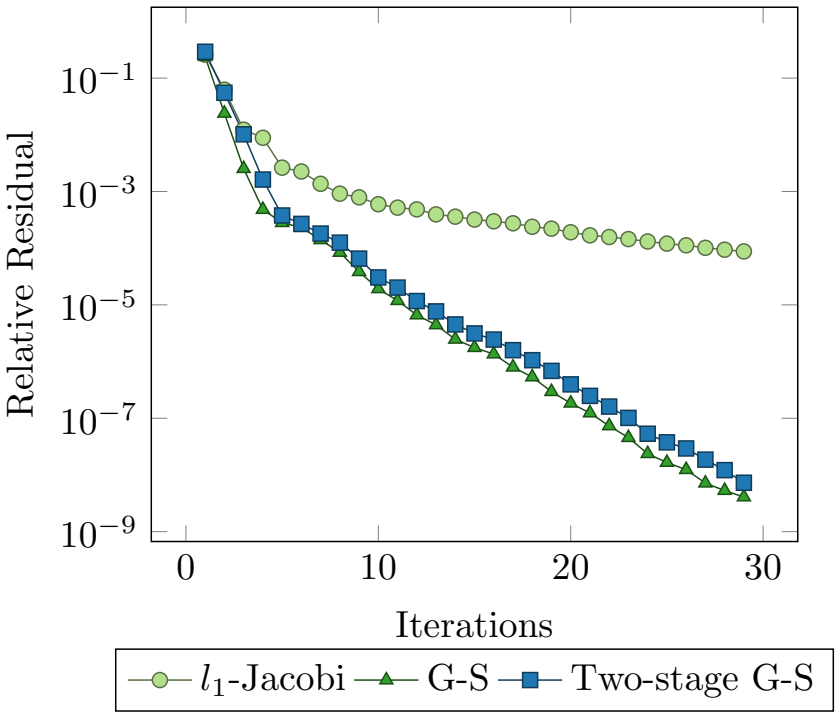

Fig. 3: Convergence history of C-AMG V-cycle with l_1-Jacobi, hybrid Gauss-Seidel and two-stage hybrid Gauss-Seidel smoothers

5.6. Compatible Relaxation in LAMG. In this section, our proposed dropping strategy is applied within the context of the Los Alamos algebraic multigrid (LAMG) framework created by Cullum and Joubert [49]. These authors have written the AMGToolBox in Matlab for the purposes of prototyping algorithms. The block lower triangular system arising from the coarse-fine ordering of the matrix at the k-th AMG V-cycle level with compatible relaxation is given below and is discussed in Brannick and Falgout [50]. This results in a recursive polynomial type smoother based upon a Gauss-Seidel iteration

and which generates a Neumann series.

Now consider the C-F splitting of the current AMG coarse level matrix, The full prolongation operator is given by

$$P = \begin{bmatrix} -A_{FF}^{-1} A_{FC} \\ I \end{bmatrix}, \quad W = -A_{FF}^{-1} A_{FC}$$

However, A_{FF}^{-1} is usually approximated by lumping or a diagonal scaling. We then form the matrix L, the lower triangular part of A above

$$L = \begin{bmatrix} M_{FF} & 0 \\ A_{CF} & M_{CC} \end{bmatrix}$$

The Gauss-Seidel smoother is based on $M_{FF} = D_{FF} + L_{FF}$ and successive over-relaxation (SOR) employs damping factor ω and forms the matrix $M_{FF} = D_{FF} + \omega\, L_{FF}$. This also applies to the matrix $M_{CC} = D_{CC} + L_{CC}$.

A polynomial type smoother based on a Gauss-Seidel iteration applied to the linear system, $Ax = b$, where D is the diagonal of A, is given by

$$\mathbf{x}_k = (I + D^{-1}L)^{-1} D^{-1}\, \mathbf{b}$$

The matrix inverse can be replaced by a truncated Neumann series, where the inverse approximation is given by,

$$(\, I + D^{-1}L \,)^{-1} = I - D^{-1}L + (D^{-1}L)^2 - (D^{-1}L)^3 + \ldots$$

Convergence of the series depends on the eigenvalues of the iteration matrix $\rho(G) < 1$, where

$$G = (\, I - D^{-1}(\, I + L \,)\,)$$

We also note that the Neumann series smoother converges rapidly for close to normal matrices where the off-diagonal elements of L decay rapidly to zero. Because the matrix L is once again strictly lower triangular, it is idempotent and the Neumann series is a finite sum. A further improvement in the convergence rate of the GMRES+AMG solver is obtained in combination with the residual correction that was introduced by Verbeek and Cullum (2001). At each level in the V-cycle, the basic AMG correction vector is augmented with an additional correction to the fine points.

$$\mathbf{x}_{FF} = \mathbf{x}_{FF} + A_{FF}^{-1}\, \mathbf{r}_{FF}$$

Scaling results in maximum diagonal elements $\max \mathrm{diag}(\, A_{FF} \,) < 1$ and thus significantly improves the accuracy of the correction.

The above polynomial algorithm is cache-optimal for CPU architectures. Instead of triangular solves with backwards or forwards sweeps (with cache fetch ordering conflicts), the algorithm relies on matrix-vector products in natural (fastest stride-1 memory reference) order. The algorithm is also ideally suited to GPU implementation because matrix-vector multiplications become the dominant cost. In order to demonstrate that a dropping strategy can be an effective technique for reducing the compute time and yet maintain the solver convergence rate, we consider the 2-D Laplace problem as studied in Joubert and Cullum [49].

The drop tolerance is set to $tol = 1.0$ and thus the smoother becomes more like a standard Jacobi iteration for the finest levels of the V–cycle. The number of non-zeros $nnz(L)$ of the lower triangular matrix in the splitting $A = D + L + U$ are plotted in Figure 4. The convergence history of the GMRES-AMG solver is plotted in Figure 5, where it can be seem that the slope of the relative residual error remains the same as without dropping. The convergence rate for the solver with a Jaocbi smoother is also plotted and exhibits a different slope. We postulate that the dropping acts like an adaptive smoother in the V-cycle, where the iterations switch from Jacobi on the fine levels to Gauss-Seidel on the coarser levels. The adaptive smoothers developed by Magri et al. [51] are based upon sparse approximate inverses (SPAI) and their efficacy depends on the amount of fill-in as measured by the number of non-zeros in the resulting matrix at each level. In our case, this is determined by the drop tolerance and diagonal dominance of the coarse matrices.

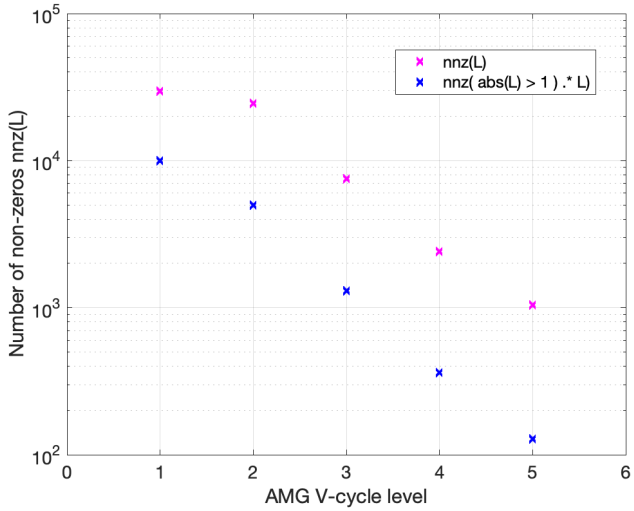

Fig. 4: 2-D Laplace problem $N = N_x \times N_y = 10,000$. Number of non-zeros.

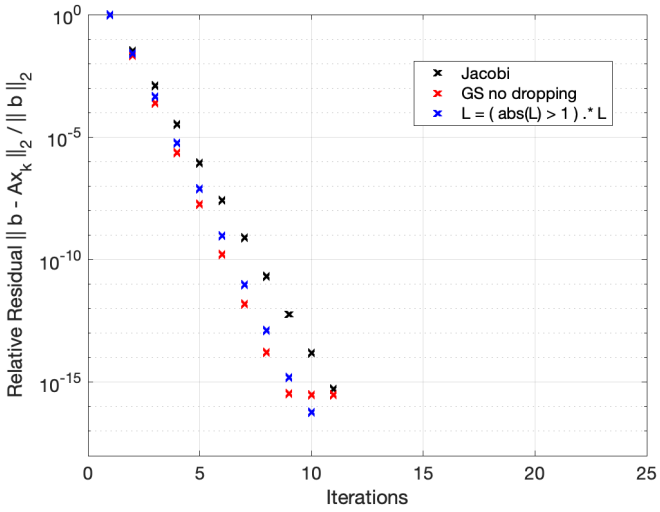

Fig. 5: 2-D Laplace problem $N = N_x \times N_y = 10,000$

Next consider the McAlister blade simulation described earlier. The problem is run again but now using the LAMG V-cycle as the preconditioner for the one-reduce MGS-GMRES solver. With dropping set to `L = (abs(L) > 1e-8) .* L`, the number of non-zeros $nnz(L)$ with and without dropping are plotted in Figure 6. The convergence history of the GMRES-AMG solver remains essentially unchanged with the dropping strategy as can be observed in Figure 7. The compute time for the solver with dropping is reduced by at least 10%.

6. Conclusions. In this paper we have shown how truncated Neumann series play an important role in both a new formulation of the MGS-GMRES algorithm and inner-outer Gauss-Seidel smoother iterations for an AMG preconditioner. The resulting implementations of these algorithms are well-suited to multi-core computer architectures such as GPUs because they rely on matrix-vector products. To a large extent, the loss of orthogonality of the Krylov vectors determines when the MGS-GMRES solver will converge to a backward stable solution. Our corollary to the seminal backward stability proof of Paige

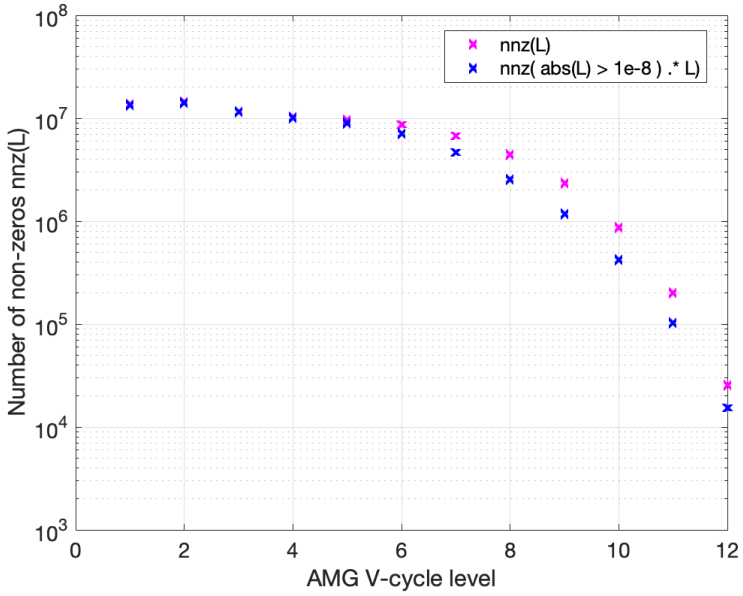

Fig. 6: C-AMG V-cycle number of non-zeros $nnz(L)$

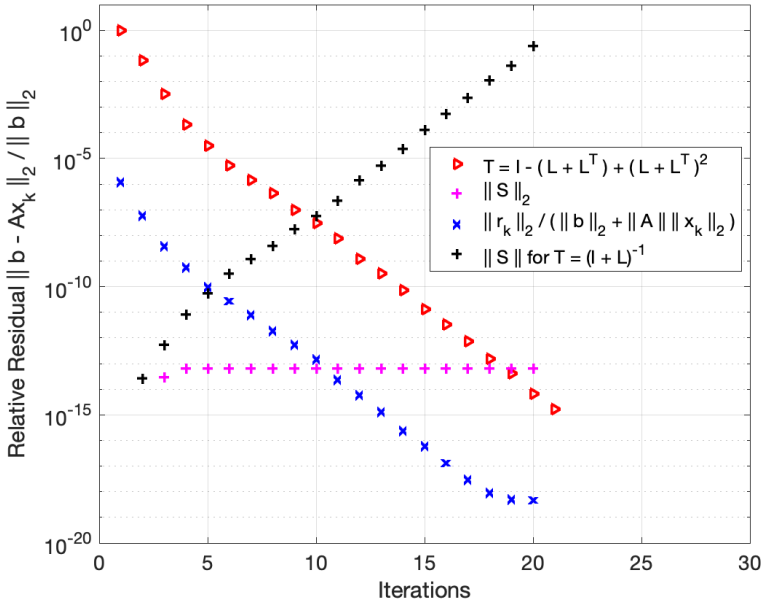

Fig. 7: Nalu-Wind pressure solver convergence history

et al. [1] demonstrates that the inverse compact WY MGS projector appears as the $(2,2)$ block of the augmented Householder transformation matrix introduced in Björck and Paige [10]. Furthermore, the loss of orthogonality results elucidated by Paige and Strakoš [9] imply that the strictly lower triangular matrix L_k, appearing in the correction matrix T, remains small and thus $T = I - L_k$ is sufficient for convergence of MGS-GMRES when $\|L_k\|_F^p = \mathcal{O}(\varepsilon^p)\kappa_F^p(B)$, and $p > 1$. Furthermore, the matrix L_k

is idempotent because it is strictly lower triangular. Therefore the Neumann series is a finite sum. Numerical experiments demonstrate that the truncated Neumann series formulation of MGS-GMRES produces the same loss of orthogonality and norm-wise relative backward error as the original algorithm due to Saad and Schultz [2]. In particular, the norm-wise relative backward error reaches $\mathcal{O}(\varepsilon)$ when the Krylov vectors lose linear independence and the loss of orthogonality as measured by $\|S\|_2$ increases to one, at which point the relative residual error stalls.

In addition to these results, we have demonstrated that the inner-outer two-stage Gauss-Seidel iteration may be expressed as a truncated Neumann series. When higher-order terms are included in the sum, rapid convergence is achieved for GMRES-AMG solvers applied to the incompressible Navier-Stokes equations in computational fluid dynamics. An effective strategy for reducing the computation time is to apply a dropping strategy to the matrix L from the splitting $A = D + L + U$ at each level in the V-cycle. The dropping strategy was applied in two different implementations of the C-AMG solver, namely a Matlab prototype for the LAMG toolbox from LANL, written by Joubert and Cullum [49] and the Hypre-BoomerAMG framework from LLNL. The former includes compatible relaxation as described by Brannick and Falgout [50] and dropping was evaluated in this context. For hypre the dropping strategy was applied within each parallel subdomain by hybrid smoothers for the local block diagonal matrices on each MPI rank. Pressure continuity problems from the Nalu-Wind incompressible Navier-Stokes CFD models were examined. GMRES-AMG solver convergence was not adversely affected by the dropping. The number of non-zeros per row is reduced and the associated solver computation time is lowered. Indeed the dropping strategy results in an adaptive smoother that changes form depending on the level in the V–cycle hierarchy. The recent work of Magri et al. [51] employs a sparse approximate inverse at each level to construct adaptive smoothers. The amount of fill-in or number of non-zeros determines the quality of the smoother and is analogous to our dropping strategy, where small elements of diagonally dominant matrices at each coarse level are removed. Further study is certainly merited to determine if this approach applies to a broader class of problems than fluid mechanics.

Acknowledgment. Funding was provided by the Exascale Computing Project (17-SC-20-SC). The National Renewable Energy Laboratory is operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. A portion of this research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725 and using computational resources at NREL, sponsored by the DOE Office of Energy Efficiency and Renewable Energy.

Appendix. Block Matrix Inverses and WY Modified Gram-Schmidt . For the block upper triangular matrix inverse, we can write

$$\begin{bmatrix} A & X \\ 0 & B \end{bmatrix} \begin{bmatrix} A^{-1} & Y \\ 0 & B^{-1} \end{bmatrix} = \begin{bmatrix} I & AY + XB^{-1} \\ 0 & I \end{bmatrix} \tag{6.1}$$

where $Y = -A^{-1}\, X\, B^{-1}$ In the case of the block lower triangular matrix inverse, we have

$$\begin{bmatrix} A & 0 \\ X & B \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ Y & B^{-1} \end{bmatrix} = \begin{bmatrix} I & 0 \\ XA^{-1} + BY & I \end{bmatrix} \tag{6.2}$$

where $Y = -B^{-1}\, X\, A^{-1}$ Finally, for the inverse compact WY MGS, given the recursive T matrix generated by

$$T^{-1} = (\, I + L_{j-1}\,) = \begin{bmatrix} T_{j-2}^{-1} & 0 \\ \mathbf{q}_{j-1}^T Q_{j-2} & 1 \end{bmatrix} \tag{6.3}$$

it is possible to prove that

$$\begin{bmatrix} T_{j-2}^{-1} & 0 \\ \mathbf{q}_{j-1}^T Q_{j-2} & 1 \end{bmatrix} \begin{bmatrix} T_{j-2} & 0 \\ -\mathbf{q}_{j-1}^T Q_{j-2} T_{j-2} & 1 \end{bmatrix} = \begin{bmatrix} I_{j-2} & 0 \\ 0 & 1 \end{bmatrix} \tag{6.4}$$

Therefore, lower triangular correction matrix $T = (\, I + L\,)^{-1}$ from Björck [30] and Paige et al. [1] is equivalent to the compact WY matrix from Björck [27]. For another example, consult equations (2.8)

and (3.10) of Barlow [22]. The inverse for a general block matrix is given by Lu and Shiou [52] or For the matrix R given by

$$R = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

The block inverse takes the form

$$R^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(\,D - CA^{-1}B\,)^{-1}CA^{-1} & -A^{-1}B(\,D - CA^{-1}B\,)^{-1} \\ -(\,D - CA^{-1}B\,)^{-1}CA^{-1} & (\,D - CA^{-1}B\,)^{-1} \end{bmatrix}$$

Therefore, in the case of a symmetric correction matrix T_{j-1}, it follows that

$$\begin{bmatrix} T_{j-2}^{-1} & Q_{j-2}^T\,\mathbf{q}_{j-2} \\ \mathbf{q}_{j-2}^T\,Q_{j-2} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} T_{j-2} & -T_{j-2}\,L_{j-1,1:j-2}^T \\ -L_{j-1,1:j-2}\,T_{j-2} & 1 \end{bmatrix}$$

where $\alpha^{-1} = (\,1 - L_{j-1,1:j-2}\,T_{j-2}\,L_{j-1,1:j-2}^T\,)$ is the inverse Schur complement and $\mathcal{O}(\varepsilon^2)\|L\|_2^2$ terms are dropped from the X matrix derived using the Sherman-Morrison-Woodbury formula for a symmetric X in the projection matrix $P = I - Q\,X\,Q^T$ given in Bielich [53].

REFERENCES

[1] C. C. Paige, M. Rozložník, Z. Strakoš, Modified Gram-Schmidt (MGS), least squares, and backward stability of MGS-GMRES, SIAM Journal on Matrix Analysis and Applications 28 (1) (2006) 264–284.

[2] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on scientific and statistical computing 7 (3) (1986) 856–869.

[3] K. Swirydowicz, J. Langou, S. Ananthan, U. Yang, S. Thomas, Low synchronization Gram-Schmidt and generalized minimal residual algorithms, Numerical Linear Algebra with Applications 28 (2020) 1–20.

[4] J. W. Ruge, K. Stüben, Algebraic multigrid, in: Multigrid methods, SIAM, 1987, pp. 73–130.

[5] L. Giraud, J. Langou, M. Rozložník, On the loss of orthogonality in the Gram-Schmidt orthognalization process, Computers and Mathematics with Applications 50 (2005) 1069–1075. doi:doi:10.1016/j.camwa.2005.08.009.

[6] C. C. Paige, The effects of loss of orthogonality on large scale numerical computations, in: Proceedings of the International Conference on Computational Science and Its Applications, Springer-Verlag, 2018, pp. 429–439.

[7] A. Ruhe, Numerical aspects of Gram-Schmidt orthogonalization of vectors, Linear Algebra and its Applications 52 (1993) 591–601.

[8] A. Greenbaum, M. Rozložník, Z. Strakoš, Numerical behaviour of the modifed Gram-Schmidt GMRES implementation, BIT 37 (3) (1997) 706–719.

[9] C. C. Paige, Z. Strakoš, Residual and backward error bounds in minimum residual krylov subspace methods, SIAM Journal on Scientific and Statistical Computing 23 (6) (2002) 1899–1924.

[10] Å. Björck, C. C. Paige, Loss and recapture of orthogonality in the modfied gram-schmidt algorithm, SIAM Journal on Matrix Analysis and Applications 13 (1992) 176–190.

[11] L. Giraud, S. Gratton, J. Langou, A rank-k update procedure for reorthogonalizing the orthogonal factor from modified Gram-Schmidt, SIAM J. Matrix Analysis and Applications 25 (4) (2004) 1163–1177.

[12] P. Lanzkron, D. Rose, D. Szyld, Convergence of nested iterative methods for linear systems, Numer. Math. 58 (1991) 685—702.

[13] D. Szyld, T. Jones, The two stage and multi-splitting methods for the parallel solution of linear systems, SIAM J. Matrix Anal. Appl. 13 (1992) 671—679.

[14] A. Frommer, D. Szyld, H-splittings and two-stage iterative methods, Numer. Math. 63 (1992) 345—356.

[15] S. Thomas, I. Yamazaki, L. Berger-Vergiat, B. Kelley, S. Rajamanickam, J. Hu, K. Świrydowicz, P. Mullowney, Two-stage Gauss–Seidel preconditioners and smoothers for Krylov solvers on a GPU cluster, SIAM J. Sci. Comput.Submitted (2021).

[16] P. Mullowney, R. Li, S. Thomas, S. Ananthan, A. Sharma, A. Williams, J. Rood, M. A. Sprague, Preparing an incompressible-flow fluid dynamics code for exascale-class wind energy simulations, in: Proceedings of the ACM/IEEE Supercomputing 2021 Conference, ACM, 2021.

[17] A. Bienz, W. Gropp, L. Olson, Node-aware improvements to allreduce, in: Proceedings of the 2019 IEEE/ACM Workshop on Exascale MPI (ExaMPI), Association for Computing Machinery, 2019, pp. 1–10.

[18] S. J. Leon, Å. Björck, W. Gander, Gram-schmidt orthogonalization: 100 years and more, Numerical Linear Algebra with Applications 20 (3) (2013) 492–532.

[19] C. Puglisi, Modification of the Householder method based on the compact WY representation, SIAM Journal on Scientific and Statistical Computing 13 (3) (1992) 723–726.

[20] J. Malard, C. C. Paige, Efficiency and scalability of two parallel QR factorization algorithms, in: Proceedings of IEEE Scalable High Performance Computing Conference, 1994, pp. 615–622.

[21] R. Schreiber, C. V. Loan, A storage-efficient WY representation for products of householder transformations, SIAM Journal on Scientific and Statistical Computing 10 (1) (1989) 53–57.

[22] J. L. Barlow, Block modified Gram–Schmidt algorithms and their analysis, SIAM Journal on Matrix Analysis and Applications 40 (4) (2019) 1257–1290.

[23] S. K. Kim, A. T. Chronopoulos, An efficient parallel algorithm for extreme eigenvalues of sparse nonsymmetric matrices, International Journal of Supercomputer Applications 6 (1) (1992) 98–111.
[24] S. J. Thomas, E. Carson, K. Lund, K. Świrydowicz, Improving the backward stability of mixed-precision MGS-GMRES, in: SIAM Conference on Computational Science and Engineering, SIAM, 2021.
URL https://meetings.siam.org/sess/dsp_talk.cfm?p=108409
[25] H. Anzt, A survey of numerical linear algebra methods utilizing mixed precision arithmetic, International Journal of High-Performance Computing and Applications (2021).
[26] E. Carson, K. Lund, M. Rozložník, S. Thomas, An overview of block Gram-Schmidt methods and their stability properties (2020). arXiv:2010.12058.
[27] Å. Björck, Numerics of Gram-Schmidt orthogonalization, Linear Algebra and Its Applications 197 (1994) 297–316.
[28] C. C. Paige, A useful form of unitary matrix obtained from any sequence of unit 2-norm n-vectors, SIAM Journal on Matrix Analysis and Applications 31 (2) (2009) 565–583.
[29] C. C. Paige, W. Wülling, Properties of a unitary matrix obtained from a sequence of normalized vectors, SIAM Journal on Matrix Analysis and Applications 35 (2) (2014) 526–545.
[30] A. Björck, Solving least squares problems by Gram–Schmidt orthogonalization, BIT 7 (1967) 1–21.
[31] W. Oettli, W. Prager, Compatibility of approximate solutions of linear equations with given error bounds for coefficients and right hand sides, Numeriche Mathematik 6 (1964) 405—409.
[32] J. L. R. andJ. Gaches, On the compatibility of a given solution with the data of a linear system, Journal of the ACM 14 (1967) 543–548.
[33] S. J. Thomas, S. Ananthan, S. Yellapantula, J. J. Hu, M. Lawson, M. A. Sprague, A comparison of classical and aggregation-based algebraic multigrid preconditioners for high-fidelity simulation of wind turbine incompressible flows, SIAM Journal on Scientific Computing 41 (5) (2019) S196–S219.
[34] A. Brandt, S. McCormick, J. W. Ruge, Algebraic multigrid (AMG) for sparse matrix equations, in: Evans (Ed.), Sparsity and Its Applications, Cambridge University Press, Cambridge, 1984.
[35] K. Stüben, Algebraic multigrid (AMG): an introduction with applications, in: U. Trottenberg, A. Schuller (Eds.), Multigrid, Academic Press, Inc., USA, 2000.
[36] A. H. Baker, R. D. Falgout, T. V. Kolev, U. Meier-Yang, Scaling Hypre's Multigrid Solvers to 100,000 Cores, Springer London, London, 2012, pp. 261–279.
[37] U. M. Yang, Parallel algebraic multigrid methods — high performance preconditioners, in: B. A.M., T. A. (Eds.), Numerical Solution of Partial Differential Equations on Parallel Computers. Lecture Notes in Computational Science and Engineering, Vol. 51, Springer, Berlin, Heidelberg, 2006.
[38] U. M. Yang, On long range interpolation operators for aggressive coarsening, Numerical Linear Algebra with Applications 17 (2010) 453–472.
[39] R. D.Falgout, J. B. Schroder, Non-galerkin coarse grids for algebraic multigrid, SIAM Journal on Scientific Computing 36 (2014) 309–334.
[40] H. De Sterck, U. Meier-Yang, J. J. Heys, Reducing complexity in parallel algebraic multigrid preconditioners, SIAM Journal on Matrix Analysis and Applications 27 (4) (2006) 1019–1039.
[41] M. Luby, A simple parallel algorithm for the maximal independent set problem, SIAM Journal on Computing 15 (1986) 1036–1053.
[42] T. Manteuffel, S. McCormick, M. Park, J. Ruge, Operator-based interpolation for bootstrap algebraic multigrid, Numerical Linear Algebra with Applications 17 (2-3) (2010) 519–537.
[43] H. De Sterck, R. D. Falgout, J. W. Nolting, U. Meier-Yang, Distance-two interpolation for parallel algebraic multigrid, Numerical Linear Algebra with Applications 15 (2-3) (2008) 115–139.
[44] R. Li, B. Sjogreen, U. Meier-Yang, A new class of AMG interpolation operators based on matrix matrix multiplications, To appear SIAM Journal on Scientific Computing (2020).
[45] U. Meier-Yang, On long-range interpolation operators for aggressive coarsening, Numerical Linear Algebra with Applications 17 (2-3) (2010) 453–472.
[46] R. D. Falgout, R. Li, B. Sjogreen, U. Meier-Yang, Porting *hypre* to heterogeneous computer architectures: Strategies and experiences, submitted to Parallel Computing (2020).
[47] A. H. Baker, R. D. Falgout, T. V. Kolev, U. M. Yang, Multigrid smoothers for ultraparallel computing, SIAM J. Sci. Comput. 33 (2011) 2864–2887.
[48] A. Bienz, R. D. Falgout, W. Gropp, J. B. S. L. N. Olson, Reducing parallel communication in algebraic multigrid through sparsification, SIAM Journal on Scientific Computing 38 (2016) 332–357.
[49] W. Joubert, J. Cullum, Scalable algebraic multigrid on 3500 processors, Electronic Transactions on Numerical Analysis 23 (2006) 105–128.
[50] J. J. Brannick, R. D. Falgout, Compatible relaxation and coarsening in algebraic multigrid, SIAM Journal on Scientific Computing 32 (3) (2010) 1393–1416.
[51] V. A. Paludetto Magri, A. Franceschini, C. Janna, A novel algebraic multigrid approach based on adaptive smoothing and prolongation for ill-conditioned systems, SIAM Journal on Scientific Computing 41 (1) (2019) A190–A219.
[52] T.-T. Lu, S.-H. Shiou, Inverses of 2×2 block matrices, Computers and Mathematics with Applications 44 (2002) 119–129.
[53] D. Bielich, Orthogonalization algorithms in the context of modern high performance computation and their stability, Ph.D. thesis, University of Colorado, Denver, CO (2021).