MIXED PRECISION AUGMENTED GMRES*

YONGSEOK JANG†, PIERRE JOLIVET†, AND THEO MARY†

Abstract. We aim to accelerate the restarted generalized minimal residual (GMRES) method for the solution of linear systems by combining two types of techniques. On the one hand, mixed precision GMRES algorithms, which use lower precision in certain steps of the inner cycles, offer significant reductions to computational and memory costs. On the other hand, augmented GMRES algorithms, which recycle information on the eigenvalues of the matrix between restarts by incorporating additional vectors into the Krylov basis, can significantly speed up the convergence. In this work, we investigate how to combine mixed precision and augmentation, in order to cumulate the reduced per-iteration cost of the former with the reduced number of iterations of the latter. We first explore the GMRES with deflated restarting (GMRES-DR) variant, which we show to present limited mixed precision opportunities. Indeed, GMRES-DR can exploit a preconditioner constructed in low precision, but requires a flexible paradigm to also apply it in low precision; moreover, the matrix–vector product and orthonormalization steps must both be kept in high precision as otherwise the method stagnates at low accuracy. We explain that this is because GMRES-DR is based on some algebraic simplifications that are only valid in exact arithmetic, but fail to hold in finite precision. This observation leads us to investigate a more general augmented GMRES framework (AugGMRES) that avoids making these simplifications. AugGMRES is much more resilient to the use of low precision, does not require a flexible paradigm, and successfully converges to high accuracy even when low precision is used for all inner operations. Our numerical experiments illustrate the robustness and fast convergence of AugGMRES on a range of sparse matrices, including ill-conditioned real-life ones.

Key words. Generalized minimal residual, GMRES, mixed precision, reduced precision, augmentation, deflation, sparse matrices

MSC codes. 65F10, 65F25, 65Y04, 68Q25

1. Introduction.

The generalized minimal residual (GMRES) method is a powerful iterative solver widely used for large sparse linear systems arising in various scientific computing and engineering applications. While a unified convergence analysis of GMRES is not fully developed, convergence bounds can be estimated based on the eigenvalues of the matrix A, influenced by its normality (see [36, 21]). In some cases, removing or deflating small eigenvalues can significantly accelerate the convergence. This motivates the idea of *augmenting* the Krylov subspace with eigenvectors associated with small eigenvalues of A, which amounts to deflate the eigenvalue from A. In the context of restarted GMRES, the Krylov subspaces built in the previous cycles can be recycled to approximate these eigenvectors. Augmented GMRES approaches are very general, and can incorporate in the Krylov basis any kind of vectors, not only eigenvectors; see [18] for a more extensive discussion of augmented GMRES, and its connection with deflated approaches.

While traditional numerical linear algebra has predominantly relied on double precision floating-point arithmetic (fp64), in the recent years, mixed precision algorithms have emerged as a promising solution to reduce computational and memory costs without sacrificing accuracy [28]. These algorithms allow for the use of multiple precision levels within a single computation. In particular, in the context of restarted GMRES, not too ill-conditioned systems can be solved to high accuracy with a significant usage of lower precision arithmetic. This is because restarted GMRES is a form of iterative refinement, which allows the operations performed in the inner cy-

*Version of July 15, 2025.

†Sorbonne Université, CNRS, LIP6, 4 place Jussieu, 75005 Paris, France
(yongseok.jang@lip6.fr, pierre.jolivet@lip6.fr, theo.mary@lip6.fr).

cles (preconditioning, matrix–vector product, orthonormalization) to be carried out in lower precision; it suffices to keep the outer operations (solution update and residual computation) in high precision [16, 4]. This observation has been leveraged to obtain significant speedups on modern hardware [3, 31, 1, 28]. For a more comprehensive overview of mixed precision algorithms in GMRES, we refer the reader to [28, sect. 8] and the references therein.

The main contribution of this paper is to show that (and how) mixed precision can be combined with the aforementioned augmented GMRES variants. We are only aware of one existing work on the topic: the paper by Oktay and Carson [37], which proposes a GCRO-DR [38] based iterative refinement that constructs the preconditioner (an LU factorization) in lower precision. However, all other operations, including the preconditioner application, are kept in high precision, and so the potential acceleration is limited. In contrast, in this paper, we seek to develop mixed precision augmented GMRES variants in a much more general framework, allowing lower precision to be used not only in the preconditioner construction, but also in the preconditioner application, the matrix–vector products, the orthonormalization, and the eigenvalue problems that arise in augmented GMRES. Moreover, each of these operations is parametrized by its own precision parameter, so that the choice to switch them to low precision can be made independently of the other steps, and so that different levels of low precision (such as fp32 and fp16) can be simultaneously used within a given variant.

While we propose a general mixed precision augmented GMRES framework, we focus our study and experiments on two specific augmented GMRES variants. First, we explore GMRES-DR [35], one of the most popular variants, and show it presents significant limitations in the use of mixed precision. While lower precision arithmetic can be used to construct the preconditioner, applying it in low precision requires the use of a flexible paradigm. More importantly, low precision cannot be used at all for other critical operations like the matrix–vector products and the orthonormalization, as its use leads GMRES-DR to stagnate at a correspondingly low accuracy. We explain that this is because GMRES-DR is based on some algebraic simplifications that only hold in exact arithmetic, but not in finite precision. To address this issue, we turn to a second mixed precision augmented GMRES (AugGMRES) variant, that is more general and remains valid even in finite precision. We show that this AugGMRES variant can converge to high accuracy when using low precision for all inner operations, including the matrix–vector products and the orthonormalization; moreover, it does not require the use of a flexible paradigm.

We perform extensive numerical experiments, using fp64 as the high precision and both fp32 and fp16 as lower precisions, with a range of matrices, including real-life ones. Overall, our proposed mixed precision AugGMRES method demonstrates significant improvements in GMRES convergence rates while successfully incorporating lower precision arithmetic in most of the operations.

The rest of this paper is structured as follows: section 2 covers the necessary preliminaries on mixed precision and augmentation techniques for GMRES. Section 3 introduces the general mixed precision AugGMRES framework. Section 4 explores the specialization of this framework to the mixed precision GMRES-DR method and discusses its limitations. Section 5 presents numerical experiments illustrating the success of AugGMRES. Section 6 concludes the paper and discusses future research directions.

Notations. This work adopts a notation system suitable for complex-valued systems. Vectors are denoted by bold lowercase letters, such as $\boldsymbol{x} \in \mathbb{C}^n$ for $n \in \mathbb{N}$, while matrices are represented by uppercase letters. The ℓ_2 norm of a vector $\boldsymbol{x}$ is denoted as $\|\boldsymbol{x}\|$. A matrix transpose is indicated by X^T, the Hermitian transpose by X^H, and the pseudo-inverse by $X^\dagger$. The smallest and largest singular values of X are denoted as $\sigma_{\min}(X)$ and $\sigma_{\max}(X)$, while the condition number is $\kappa(X) = \sigma_{\max}(X)/\sigma_{\min}(X)$. The identity matrix of order n is denoted as I_n and $\mathbf{0}_n$ is a zero column vector of length n. To make algorithm descriptions more comprehensible, MATLAB-style notation is used, such as $X(1:i, 1:j)$ to refer to the submatrix containing the first i rows and j columns.

2. Preliminaries.

Krylov subspace methods play a crucial role in solving large, sparse linear systems, with GMRES standing out as a key algorithm for dealing with nonsymmetric problems. In practice, GMRES is usually restarted to limit the size of the Krylov basis, which improves efficiency and manages memory usage, but can slow down the convergence. To enhance the convergence of restarted GMRES, augmented techniques have been subsequently developed. At the same time, restarted GMRES provides opportunities for mixed precision arithmetic, which can reduce computational costs while preserving accuracy. This section provides a brief overview of these concepts, serving as the basis for the discussions and methods presented in this paper.

2.1. GMRES.

In the GMRES method, we want to minimize the residual norm over a Krylov subspace. Let $\mathcal{K}_m(A, \boldsymbol{r}_0)$ be an m-dimensional Krylov subspace defined by

$$\mathcal{K}_m(A, \boldsymbol{r}_0) = \operatorname{span}\left\{\boldsymbol{r}_0, A\boldsymbol{r}_0, \ldots, A^{m-1}\boldsymbol{r}_0\right\},$$

where $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$ is an initial residual vector with an initial guess $\boldsymbol{x}_0$. Based on the Gram–Schmidt algorithm, the Arnoldi process generates a set of orthonormal Krylov basis vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$. Denoting as V_m the matrix formed by $[\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m]$, the Arnoldi procedure yields the so-called *Arnoldi identity*,

$$AV_m = V_{m+1}H_m,$$

where $V_{m+1}^H V_{m+1} = I_{m+1}$ and H_m is an $(m+1) \times m$ upper Hessenberg matrix. Using this identity and the orthonormality of V_{m+1}, the problem of minimizing the residual norm can be rewritten as

$$\min \|\boldsymbol{r}_m\| = \min \|\boldsymbol{b} - A\boldsymbol{x}_m\| = \min \|\boldsymbol{c} - H_m\boldsymbol{y}_m\|,$$

where $\boldsymbol{x}_m \in \boldsymbol{x}_0 + \mathcal{K}_m(A, \boldsymbol{r}_0)$, $\boldsymbol{c} = V_{m+1}^H\boldsymbol{r}_0$, and $\boldsymbol{y}_m \in \mathbb{C}^{m+1}$. Since $\boldsymbol{v}_1 = \boldsymbol{r}_0/\|\boldsymbol{r}_0\|$, we can simplify $\boldsymbol{c}$ as $\boldsymbol{c} = \|\boldsymbol{r}_0\|\,\boldsymbol{e}_1$ with the first canonical basis vector $\boldsymbol{e}_1$ of $\mathbb{R}^{m+1}$.

2.2. Augmented GMRES.

In the following, we first explain the principle behind augmented GMRES for a generic augmentation subspace. Then we focus on the special case where this subspace consists of the approximate eigenvectors computed via the harmonic Ritz formulation, and we explain the algebraic simplifications that can be operated in this case to obtain the GMRES-DR method.

Augmented GMRES framework. In the augmented GMRES framework, we aim to find the approximate solution $\boldsymbol{x}_m$ in a *search space* $\mathcal{S}_m$ to minimize the residual norm such that

$$\boldsymbol{x}_m \in \boldsymbol{x}_0 + \mathcal{S}_m \qquad \text{and} \qquad \boldsymbol{r}_m \in \boldsymbol{r}_0 + A\mathcal{S}_m,$$

with the Galerkin orthogonal condition $\boldsymbol{r}_m \perp A\mathcal{S}_m$ (also known as the *minimal residual principle*). In GMRES, the search space $\mathcal{S}_m$ is given by the Krylov subspace $\mathcal{K}_m(A, \boldsymbol{r}_0)$.

In this paper, we construct the augmented subspace such that

$$\mathcal{S}_m := \mathcal{K}_{m-k}(A, \boldsymbol{r}_0) + \mathcal{P}_k,$$

where $\mathcal{P}_k$ is an arbitrary k-dimensional subspace whose basis vectors form a matrix P_k. Using Arnoldi iterations, we can generate the Krylov basis vectors and impose the Galerkin orthogonal condition. Hence, we derive the Arnoldi-like identity such that

$$AW_m = V_{m+1}H_m, \qquad \text{where } W_m = [V_{m-k}, P_k].$$

Here, the column vectors of V_{m-k} and W_m are the basis vectors of $\mathcal{K}_{m-k}(A, \boldsymbol{r}_0)$ and $\mathcal{S}_m$, respectively. Note that while V_{m-k} is orthonormal, W_m is not.

A common choice in defining P_k is the use of eigenvectors. In [33], Morgan proposed the augmented GMRES with eigenvectors, called GMRES-E, using Rayleigh–Ritz method. The Rayleigh–Ritz method computes the approximate eigenvectors and eigenvalues as follows: for $B \in \mathbb{C}^{n\times n}$ and an m-dimensional subspace $\mathcal{S}$ of $\mathbb{C}^n$, find an eigenpair $(\boldsymbol{p}, \lambda) \in \mathbb{C}^n \times \mathbb{C}$ of B with respect to $\mathcal{S}$ satisfying

$$\boldsymbol{p} \in \mathcal{S} \qquad \text{and} \qquad B\boldsymbol{p} - \lambda\boldsymbol{p} \perp \mathcal{S}. \tag{2.1}$$

In our work, we consider $B = A^{-1}$ and $\mathcal{S} = A\mathcal{S}_m$, which gives the harmonic Ritz pairs of A. Therefore, using the basis vectors of $\mathcal{S}$ generated by the Arnoldi process, (2.1) yields

$$\begin{aligned} A^{-1}\left(AW_m\boldsymbol{g}\right) - \lambda AW_m\boldsymbol{g} \perp \mathcal{S} \;&\Leftrightarrow W_m^H A^H \left(W_m\boldsymbol{g} - \lambda AW_m\boldsymbol{g}\right) = 0 \\ &\Leftrightarrow H_m^H V_{m+1}^H W_m\boldsymbol{g} - \lambda H_m^H H_m\boldsymbol{g} = 0, \end{aligned} \tag{2.2}$$

by the Arnoldi-like identity and the orthonormality of V_{m+1}. Here, λ is a harmonic Ritz value and the associated harmonic Ritz vector $\boldsymbol{p}$ is $W_m\boldsymbol{g}$.

Remark 2.1. In general, the search space $\mathcal{S}_m$ is not a Krylov subspace but in case of employing the harmonic Ritz pairs, we have

$$\mathcal{S}_m = \operatorname{span}(\boldsymbol{r}_0, \ldots, A^{m-k}\boldsymbol{r}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_k) = \mathcal{K}_m(A, \tilde{\boldsymbol{r}}_0),$$

for some starting vector $\tilde{\boldsymbol{r}}_0$. For the existence of $\tilde{\boldsymbol{r}}_0$ and more details, please see [34].

Remark 2.2. One of the advantages of augmented GMRES is the flexibility in selecting vectors to define the augmentation subspace P_k. For instance, by defining $B = A^HA$ and $\mathcal{S} = \operatorname{range}(W_m)$, we can obtain approximate singular vectors; see [19, 30] for SVD-based methods. In [10, 39], $\boldsymbol{p}_k$ is chosen as the error vector $\boldsymbol{x}_k - \boldsymbol{x}_{k-1}$. Additional strategies for defining P_k can be found in [9]. While augmented GMRES allows for various choices of P_k, this paper primarily focuses on augmentation as formulated in (2.2).

Deriving GMRES-DR. In the specific case where the augmented subspace corresponds to the harmonic Ritz vectors, Morgan [35] has shown with his method GMRES-DR that we can operate some simplifications in order to recover an actual Krylov subspace with the familiar Arnoldi identity; see [35, Theorem 3.3]. The key simplifications that are operated by GMRES-DR are based on the fact that the harmonic residual vectors $\boldsymbol{t}_j$, defined by

$$\boldsymbol{t}_j = AV_m\boldsymbol{g}_j - \lambda_j V_m\boldsymbol{g}_j \in \operatorname{span}(V_{m+1}), \qquad \text{for } j = 1, \ldots, k,$$

are collinear with the GMRES residual vector, denoted $\boldsymbol{r}_m$. That is, there exists α_j such that $\boldsymbol{t}_j = \alpha_j \boldsymbol{r}_m$ [34, Theorem 5.5]. Morgan exploits this observation to derive the GMRES-DR method [35] as follows. Since $\boldsymbol{r}_m = V_{m+1}\boldsymbol{\rho}$ with $\boldsymbol{\rho} = \boldsymbol{c} - H_m \boldsymbol{y}_m$, we have

$$\boldsymbol{t}_j = \alpha_j \boldsymbol{r}_m = \alpha_j V_{m+1}\boldsymbol{\rho}.$$

Denoting $G_k = [\boldsymbol{g}_1, \dots, \boldsymbol{g}_k]$, we obtain

$$AV_m G_k = V_m G_k \Lambda_k + V_{m+1}\boldsymbol{\rho}\, \boldsymbol{a}_k^T = V_{m+1}\begin{bmatrix} G_k & \boldsymbol{\rho} \\ \boldsymbol{0}_k^T & \end{bmatrix}\begin{bmatrix} \Lambda_k \\ \boldsymbol{a}_k^T \end{bmatrix} = V_{m+1} G_{k+1}\begin{bmatrix} \Lambda_k \\ \boldsymbol{a}_k^T \end{bmatrix}, \tag{2.3}$$

where $\Lambda_k = \operatorname{diag}(\lambda_1, \dots, \lambda_k)$ and $\boldsymbol{a}_k^T = [\alpha_1, \dots, \alpha_k]$.

We next orthonormalize the harmonic Ritz vectors and the residual vector to form V_{k+1}. Denoting $G_{k+1} = Q_{k+1}R_{k+1}$ given by the QR decomposition, we can rewrite it as

$$G_{k+1} = \begin{bmatrix} Q_k & \boldsymbol{q}_{k+1} \\ \boldsymbol{0}_k^T & \end{bmatrix}\begin{bmatrix} R_k & \tilde{\boldsymbol{r}}_{k+1} \\ \boldsymbol{0}_k^T & \end{bmatrix}.$$

It leads to

$$AV_m Q_k = V_{m+1} Q_{k+1} R_{k+1} \left[\begin{array}{c} \Lambda_k \\ \boldsymbol{a}_k^T \end{array} \right] R_k^{-1} \tag{2.4}$$

using (2.3). Then, the Arnoldi identity into (2.4) implies that

$$V_{m+1} H_m Q_k = V_{m+1} Q_{k+1} R_{k+1} \left[\begin{array}{c} \Lambda_k \\ \boldsymbol{a}_k^T \end{array} \right] R_k^{-1}.$$

Hence, using the orthonormality of V_{m+1} and Q_{k+1}, we can define V_{k+1} and H_k by

$$V_{k+1} = V_{m+1} Q_{k+1} \qquad \text{and} \qquad H_k = Q_{k+1}^H H_m Q_k,$$

to obtain

$$AV_k = V_{k+1} H_k.$$

Then, performing $(m-k)$ Arnoldi iterations with V_{k+1} generates the remaining $(m-k)$ basis vectors to yield the familiar Arnoldi identity

$$AV_m^{\text{new}} = V_{m+1}^{\text{new}} H_m^{\text{new}}.$$

Remark 2.3. Morgan's GMRES-DR [35] is algebraically equivalent to GCRO-DR [38]. But it is not true for flexible GMRES-DR (FGMRES-DR) and flexible GCRO-DR (FGCRO-DR). Only if a certain collinearity condition is satisfied, FGMRES-DR and FGCRO-DR are equivalent [17]. We also refer to [23] for the analysis of FGMRES-DR.

2.3. Mixed precision GMRES. Mixed precision algorithms utilize different floating-point arithmetic within a given computation to improve performance while maintaining acceptable accuracy. These techniques have become increasingly relevant due to advancements in modern hardware that efficiently handle lower precision arithmetic. Double precision (fp64) arithmetic does not always fully exploit modern hardware capabilities, whereas lower precision arithmetic, such as single (fp32) or half (fp16) precisions, can offer substantial speedups and reduced memory usage. For a more comprehensive overview of mixed precision methods, we refer the reader to [28].

In the specific case of GMRES, various mixed precision methods have been proposed. Several studies have shown that lower (or mixed) precision can be exploited in specific operations, such as the matrix–vector products [22, 25], the orthonormalization [26, 2], or the preconditioner [7, 24, 6]. In the case of the preconditioner, we can distinguish simply constructing the preconditioner in lower precision, or also applying it in lower precision. The former reduces the time and memory costs of constructing the preconditioner, but not always the cost of applying it during the iterations. While memory accessor approaches [27, 5] have been proposed to translate the low storage precision into time reductions, directly performing the application in low precision may sometimes be more efficient. In this case, the rounding errors incurred in the application depend on the vectors that the preconditioner is applied to, which makes the preconditioner behave like a variable preconditioner. To address this, the flexible variant becomes essential; see also [13, 14]. Please refer to [13], which is based on the modular framework of [12], for a comprehensive analysis of a single cycle of GMRES in mixed precision.

Importantly, across multiple cycles, GMRES presents even more opportunities to use lower/mixed precision. Indeed, (potentially all) the operations in the Arnoldi process can often be switched to low precision while preserving a convergence to high accuracy. This is because restarted GMRES is a form of iterative refinement. Such GMRES-based iterative refinement approaches have been extensively analyzed [15, 16, 4] and practical implementations have been shown to significantly reduce the time and memory costs [3, 32, 31, 40]

Finally, we mention the work of Oktay and Carson [37] that, to our knowledge, is the first and so far only attempt at combining mixed precision with augmentation techniques. They propose a mixed precision GCRO-DR based iterative refinement, which however only exploits low precision for the preconditioner construction. We will discuss in detail how it compares with our proposed methods in the next section; the main point is that we aim to develop a more general method that considers the use of low precision in each (potentially all) inner operations.

3. Mixed precision augmented GMRES framework. In this section, we present a general mixed precision augmented GMRES framework, outlined in Algorithm 3.1, where the Arnoldi process is based on the modified Gram–Schmidt algorithm. After completing one cycle of GMRES, we define the augmented Krylov subspace using an arbitrary set of vectors P_k and compute an approximate solution by minimizing the residual norm. This process is repeated until the convergence criterion is satisfied.

In our general framework, every main operation of the algorithm is parametrized by its own specific precision, namely u_f for setting up the preconditioner (for example, this can be an approximate factorization), u_p for the preconditioner application, u_a for the matrix–vector product, u_o for the orthonormalization, u_e for the eigenvalue problem if necessary, and u_r for computing the residual; all other operations are performed in the working precision u, which is also the precision used for storing the matrix and the right-hand side and solution vectors. Table 3.1 summarizes these precision parameters. The interest in using such a general framework is that the choice to switch one type of operation to low precision can be made independently of the other operations, and moreover different levels of low precision can be simultaneously used within a given variant.

As mentioned earlier, Oktay and Carson [37] have proposed a mixed precision GCRO-DR based iterative refinement. While their method shares some similari-

Algorithm 3.1 Mixed precision augmented GMRES.

```
Input: matrix $A$, preconditioner $M$ in precision $u_f$, non-zero vector $\boldsymbol{b}$, size of search
    subspace $m$, size of augmentation $k$, tolerance $\varepsilon > 0$, maximum number of itera-
    tions max_it, initial vector $\boldsymbol{x}_0$, precisions $(u, u_p, u_a, u_o, u_e, u_r)$.
Output: approximate solution $\boldsymbol{x}$ for $A\boldsymbol{x} = \boldsymbol{b}$.
$\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$ in precision $u_r$.
$\beta = \|\boldsymbol{r}_0\|$ in precision $u$; $\boldsymbol{c} = [\beta, \boldsymbol{0}_m^T]^T$; it $= 1$.
// Perform Arnoldi process with the starting vector $\boldsymbol{r}_0/\beta$:
$\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$ in precision $u_o$.
for $j = 1 : m$ do
    $\boldsymbol{z} = M^{-1}\boldsymbol{v}_j$ in precision $u_p$.
    $\boldsymbol{q} = A\boldsymbol{z}$ in precision $u_a$.
    for $i = 1 : j$ do
        $H_m(i, j) = \boldsymbol{v}_i^H \boldsymbol{q}$ in precision $u_o$.
        $\boldsymbol{q} = \boldsymbol{q} - H(i, j)\boldsymbol{v}_i$ in precision $u_o$.
    end for
    $H_m(j + 1, j) = \|\boldsymbol{q}\|$ in precision $u_o$.
    $\boldsymbol{v}_{j+1} = \boldsymbol{q}/H(j + 1, j)$ in precision $u_o$.
end for
Set $V_{m+1} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_{m+1}]$.
Solve $\boldsymbol{y}^* = \arg\min \|\boldsymbol{c} - H_m\boldsymbol{y}\|$ in precision $u$.
$\boldsymbol{d} = M^{-1}(V_m\boldsymbol{y}^*)$ in precision $\min(u_o, u_p)$.
Update $\boldsymbol{x}_0 = \boldsymbol{x}_0 + \boldsymbol{d}$ in precision $u$.
$\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$ in precision $u_r$.
$\beta = \|\boldsymbol{r}_0\|$ in precision $u$; $W_m = V_m$.
while $\beta/\|\boldsymbol{b}\| > \varepsilon$ or it < max_it do
    Construct $P_k = [\boldsymbol{p}_1, \dots, \boldsymbol{p}_k]$ in precision $u_o$.
    (If required, compute $G_k = [\boldsymbol{g}_1, \dots, \boldsymbol{g}_k]$ in precision $u_e$ for $P_k = W_m G_k$.)
// Perform Arnoldi process with $\boldsymbol{v}_1 = \boldsymbol{r}_0/\|\boldsymbol{r}_0\|$ and $P_k$:
    for $j = 1 : m$ do
        if $j \le m - k$ then
            $\boldsymbol{z} = M^{-1}\boldsymbol{v}_j$ in precision $u_p$.
        else
            $\boldsymbol{z} = M^{-1}\boldsymbol{p}_{j-m+k}$ in precision $u_p$.
        end if
        $\boldsymbol{q} = A\boldsymbol{z}$ in precision $u_a$.
        for $i = 1 : j$ do
            $H_m(i, j) = \boldsymbol{v}_i^H \boldsymbol{q}$ in precision $u_o$.
            $\boldsymbol{q} = \boldsymbol{q} - H_m(i, j)\boldsymbol{v}_i$ in precision $u_o$.
        end for
        $H_m(j + 1, j) = \|\boldsymbol{q}\|$ in precision $u_o$.
        $\boldsymbol{v}_{j+1} = \boldsymbol{q}/H_m(j + 1, j)$ in precision $u_o$.
    end for
    Set $V_{m+1} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_{m+1}]$, $W_m = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_{m-k}, P_k]$, and $\boldsymbol{c} = [\beta, \boldsymbol{0}_m^T]^T$.
    Solve $\boldsymbol{y}^* = \arg\min \|\boldsymbol{c} - H_m\boldsymbol{y}\|$ in precision $u$.
    $\boldsymbol{d} = M^{-1}(V_m\boldsymbol{y}^*)$ in precision $\min(u_o, u_p)$.
    Update $\boldsymbol{x}_0 = \boldsymbol{x}_0 + \boldsymbol{d}$ in precision $u$.
    $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$ in precision $u_r$.
    $\beta = \|\boldsymbol{r}_0\|$ in precision $u$.
    it = it + 1.
end while
$\boldsymbol{x} = \boldsymbol{x}_0$.
```

Table 3.1
List of precision parameters of our mixed precision framework.

Precision	Description
u	working precision
u_f	precision to set up the preconditioner M
u_p	precision to apply M^{-1}
u_a	precision of the matrix–vector product
u_o	precision of the orthonormalization
u_e	precision to solve the eigenvalue problem
u_r	precision to compute the residual vector

ties with our framework, it also presents significant differences. Oktay and Carson's method is an iterative refinement where the inner system is solved with GCRO-DR; it therefore consists of three loops. GCRO-DR is especially suited for solving sequences of linear systems where the right-hand side changes, as is the case in Oktay and Carson's three-loop method. In contrast, Algorithm 3.1 only consists of two loops, and can be seen as an iterative refinement method where the inner system is solved with an augmented but single GMRES iteration. Thus, in our case, the right-hand side is fixed so that GCRO-DR is less natural. Moreover, our Algorithm 3.1 considers an unspecified augmentation subspace, and so is more general. Finally, and most importantly, Oktay and Carson's method only considers the use of low precision by computing low precision LU factors of the matrix, that are then used as preconditioner. All other operations performed during the actual iterations are kept in high precision. In contrast, Algorithm 3.1 considers a general framework, which allows for the use of multiple, independent (potentially low) precisions for each main operation. Moreover, we consider a general preconditioner M.

Building on the general augmented GMRES framework presented above, we now turn to a specific variant, GMRES-DR. While our framework allows for general augmentation and variable precision choices, GMRES-DR focuses on augmentation using harmonic Ritz vectors based on the collinear relation. In the following section, we examine how GMRES-DR can be extended to mixed precision settings and highlight the limitations that arise when low precision is used for critical components.

4. Mixed precision GMRES-DR and its limitations.

In GMRES-DR, as introduced in section 2, the orthonormalization of the residual vector and harmonic Ritz vectors is performed using the collinearity property (e.g., the existence of the coefficient vector $\boldsymbol{a}_k^T$), along with an additional QR decomposition on G_{k+1}. Other components, such as the Arnoldi process and the minimization step, follow the standard GMRES framework. We refer the reader to [35] for the complete algorithm and to [23] for its flexible variant.

It is worth noting that GMRES-DR and AugGMRES using harmonic Ritz vectors are mathematically equivalent [35]. As described in Algorithm 3.1, GMRES-DR can be extended to a mixed precision setting by assigning various precision levels as listed in Table 3.1. In this paper, we first focus on low precision computations in GMRES-DR, with particular attention to two key components: preconditioning and the Rayleigh–Ritz formulation, both of which are examined through numerical experiments.

Experimental setting. In this section, we will illustrate the behavior of specific instances of GMRES-DR with some numerical experiments. Our algorithm is implemented in MATLAB and uses the built-in operations, such as `eig()` for solving eigenvalue problems, "\" (backslash) for applying preconditioners and solving least squares problems, and `ilu` for constructing the incomplete LU (ILU(0), in our case) preconditioner. Computations in fp64 and fp32 arithmetic use MATLAB's `double` and `single` types, while fp16 operations are simulated using the `chop` library [29]. When we apply `chop()` to complex variables, the real and imaginary parts are processed separately.

For consistency, these illustrative experiments all use the same matrix fv3 taken from the SuiteSparse Matrix Collection[1]. This matrix has a condition number $\kappa(A) = 2.03 \times 10^3$. For the right-hand side vector, we use $\boldsymbol{b} = A\boldsymbol{e}/\|A\boldsymbol{e}\|$, where $\boldsymbol{e} = [1, \ldots, 1]^T$. We then solve the system using GMRES with $m = 10$ and GMRES-DR with $m = 10$ and $k = 2$. For this matrix, GMRES-DR significantly enhances the convergence in a uniform precision context (see, for example, Figure 4.1 below). Indeed, using deflated restarting reduces the number of cycles from 19 to 11, for cycle lengths of 10 in both cases and thus of comparable cost.

4.1. Lower precisions u_f and u_p.

While preconditioning plays a crucial role in accelerating the convergence, constructing and applying the preconditioner are often some of the most computationally expensive operations. This motivates the interest of performing them in low precision. In the following illustrative experiments, we consider an incomplete LU factorization without fill-in, ILU(0), to construct the preconditioner M.

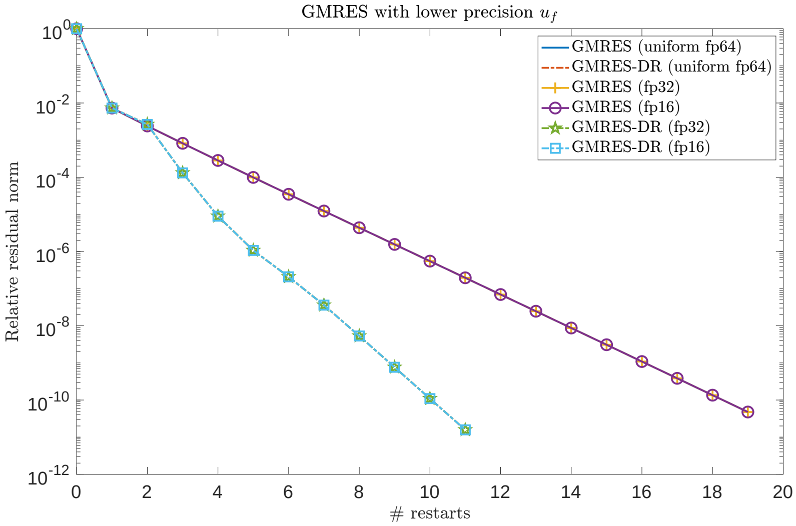

FIG. 4.1. *Lower precision preconditioning (u_f): solve fv3 with $m = 10$ and $k = 2$.*

In Figure 4.1, we first investigate constructing the preconditioner in a lower precision u_f, set to either fp32 or fp16. All other operations, including the precondition application, are kept in fp64. The figure illustrates that constructing the preconditioner M in lower precision performs comparably to the uniform precision GMRES(-DR). Remarkably, even when M is constructed in fp16, the GMRES solvers demonstrate identical convergence rates to those using higher precision preconditioners. Clearly,

[1] https://sparse.tamu.edu/

this observation is matrix-dependent, but we will provide additional experiments in section 5 that confirm this behavior on various matrices. Thus, in this first mixed precision setting, we are able to successfully combine the use of lower precision with the faster convergence of GMRES-DR.

As mentioned, in some contexts, it can be beneficial to not only construct but also apply the preconditioner in lower precision. We investigate this variant in Figure 4.2, by setting u_f to fp16, u_p to either fp32 or fp16, and all other precisions to fp64. While reducing the precision u_p does not degrade the convergence of standard GMRES, GMRES-DR stagnates at an accuracy of order u_p.

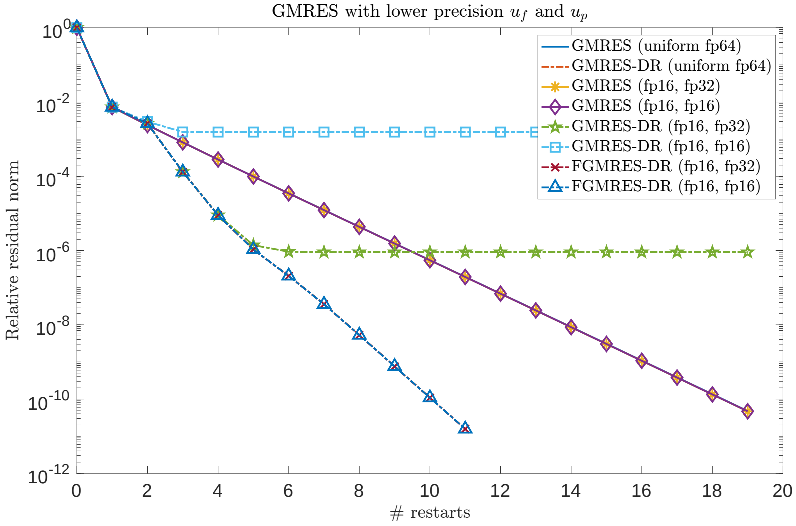

FIG. 4.2. *Mixed precision preconditioning* (u_f, u_p)*: solve fv3 with* $m = 10$ *and* $k = 2$*.*

We will explain the reason behind this stagnation in subsection 4.3, but let us first immediately mention that this issue can be readily fixed by using the flexible variant FGMRES-DR. Indeed, as illustrated Figure 4.2, FMGRES-DR preserves the convergence of GMRES-DR while allowing the preconditioner to be applied in either fp32 or fp16. Therefore, for this illustrative matrix, the preconditioner can be both constructed and applied in a precision as low as fp16 while converging as fast as the uniform precision GMRES-DR in fp64, but this comes at the expense of using the flexible variant, which requires storing two Krylov bases of size m.

4.2. Lower precisions u_e, u_o, and u_a. Next, we attempt to reduce the precision of the other inner operations, namely, u_a for the matrix–vector product, u_o for the orthonormalization, and u_e for the eigenvalue problem. Unfortunately, Figure 4.3 shows that lowering any of these precisions leads to a stagnation of FGMRES-DR to an accuracy of corresponding order. The figure illustrates this by setting each of these precisions individually to fp32 while keeping all other precisions in fp64; the same conclusion is achieved when using fp16 instead of fp32, or when reducing more than one precision simultaneously. Note that, unlike for the preconditioner application precision u_p, using the flexible variant here does not resolve this issue. Thus, while GMRES-DR may improve the convergence initially, using lower precision for u_a, u_o, or u_e eventually leads to stagnation and limits the attainable accuracy of the solver.

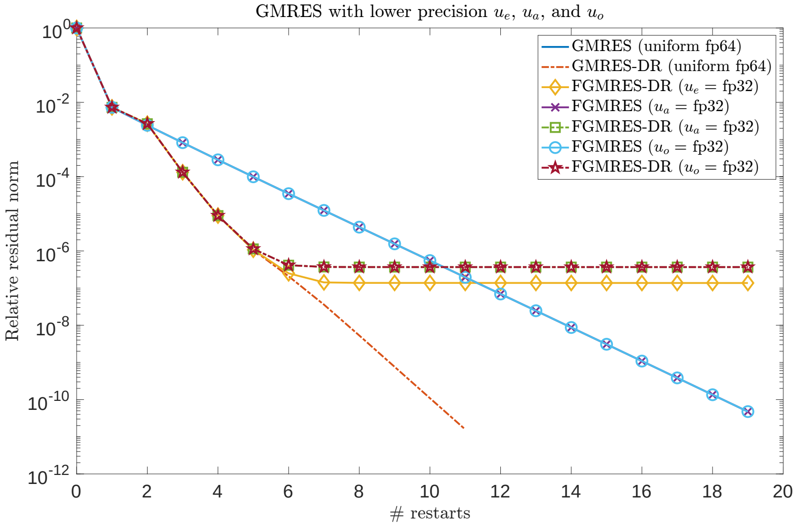

FIG. 4.3. *Lower precision eigenvalue computation and Arnoldi iterations* (u_e, u_a, u_o)*: solve fv3 with* $m = 10$ *and* $k = 2$*.*

Importantly, this is a limitation that is specific to GMRES-DR and, in particular, that is not shared by standard GMRES. Figure 4.3 shows indeed that standard GMRES continues to converge at the same speed even with these reduced precisions. We provide an explanation for this stagnation behavior in the next section.

4.3. Why GMRES-DR stagnates in low precision. As seen in subsection 2.2, the derivation of GMRES-DR is based on the fact that, in exact arithmetic, using the Arnoldi identity and the orthonormality of V_{m+1}, the residual vector $\boldsymbol{r}_m$ is written as

$$\boldsymbol{r}_m = \boldsymbol{b} - A\boldsymbol{x}_m = V_{m+1}\boldsymbol{\rho}, \tag{4.1}$$

where $\boldsymbol{\rho} = \boldsymbol{c} - H_m\boldsymbol{y}_m$ and $\boldsymbol{c} = V_{m+1}^H\boldsymbol{r}_0$.

In finite precision, however, (4.1) no longer holds exactly. Specifically, the Arnoldi process with modified Gram–Schmidt yields

$$AV_m = V_{m+1}H_m + E_m \qquad \text{and} \qquad V_{m+1}^H V_{m+1} = I_{m+1} + F_{m+1},$$

where $\|E_m\| \le p_1(n,m)\,\|A\|_F \max(u_a, u_o)$ and $\|F_{m+1}\| \le p_2(n,m)\kappa(A)u_o$ for some low-degree polynomials p_1 and p_2 in n and m; see [11] for further details.

As a result, we obtain

$$\begin{aligned}
\boldsymbol{r}_m &= \boldsymbol{b} - A\boldsymbol{x}_m \\
&= \boldsymbol{r}_0 - AV_m\boldsymbol{y}_m \\
&= \boldsymbol{r}_0 - V_{m+1}H_m\boldsymbol{y}_m - E_m\boldsymbol{y}_m \\
&= V_{m+1}V_{m+1}^H\boldsymbol{r}_0 - V_{m+1}H_m\boldsymbol{y}_m - E_m\boldsymbol{y}_m + (I_n - V_{m+1}V_{m+1}^H)\boldsymbol{r}_0 \\
&= V_{m+1}\boldsymbol{\rho} + \boldsymbol{\delta}_r,
\end{aligned}$$

with

$$\boldsymbol{\delta}_r = -E_m\boldsymbol{y}_m + (I_n - V_{m+1}V_{m+1}^H)\boldsymbol{r}_0.$$

Note that, in exact arithmetic, $\boldsymbol{r}_0 \in \operatorname{span}(V_{m+1})$ and thus $(I_n - V_{m+1}V_{m+1}^H)\boldsymbol{r}_0 = \boldsymbol{0}_n$. However, this does not hold in inexact arithmetic due to the loss of orthogonality of V_{m+1}. If $\boldsymbol{r}_0 = V_{m+1}\boldsymbol{c}_0$ for some coefficient vector $\boldsymbol{c}_0$, then

$$\left\|(I_n - V_{m+1}V_{m+1}^H)\boldsymbol{r}_0\right\| = \|V_{m+1}F_{m+1}\boldsymbol{c}_0\| \lesssim \|F_{m+1}\|\,\|\boldsymbol{r}_0\|\,,$$

neglecting second-order terms. We thus have

$$\|\boldsymbol{\delta}_r\| \lesssim p_3(n,m)\big(\kappa(A)u_o\,\|\boldsymbol{r}_0\| + \max(u_o,u_a)\,\|A\|\,\|\boldsymbol{y}_m\|\,\big).$$

This shows that the approximation $\boldsymbol{r}_m \approx V_{m+1}\boldsymbol{\rho}$ operated by GMRES-DR only holds up to an error term $\boldsymbol{\delta}_r$ proportional to $\max(u_o,u_a)$. Since this term is not taken into account by GMRES-DR, it will cause stagnation once the residual becomes smaller than $\boldsymbol{\delta}_r$.

Indeed, let $\widetilde{\boldsymbol{r}}_m = V_{m+1}\boldsymbol{\rho}$ and let

$$\widetilde{\boldsymbol{r}}_m^{\text{new}} = \widetilde{\boldsymbol{r}}_m - AV_{m+1}^{\text{new}}y_m$$

be the residual obtained by a new cycle of GMRES-DR. The true residual is

$$\boldsymbol{r}_m^{\text{new}} = \boldsymbol{r}_m - AV_{m+1}^{\text{new}}y_m = \widetilde{\boldsymbol{r}}_m^{\text{new}} - \boldsymbol{\delta}_r$$

Thus, even if $\|\widetilde{\boldsymbol{r}}_m^{\text{new}}\|$ becomes small, $\|\boldsymbol{r}_m^{\text{new}}\|$ may stagnate at the level of $\|\boldsymbol{\delta}_r\|$.

A lower precision u_e will also lead to the stagnation of GMRES-DR due to its influence in the computation of G_k. Specifically, inaccuracies in eigenvectors imply that the subspace spanned by $\{\boldsymbol{p}_1,\dots,\boldsymbol{p}_k,\boldsymbol{r}_0\}$ no longer exactly aligns with the original Krylov subspace spanned by V_{k+1}, that is,

$$\operatorname{range}([P_k,\boldsymbol{r}_0]) \neq \operatorname{range}(V_{k+1}).$$

In the next section, we show that this stagnation issue can be overcome by going back to the general case of augmented GMRES.

5. Numerical experiments with AugGMRES.

In this section, we go back to the general case of Algorithm 3.1. We illustrate experimentally that mixed precision can successfully be employed if we do not operate the GMRES-DR simplifications. We select the same augmented vectors as in GMRES-DR: the eigenvectors corresponding to the k largest eigenvalues λ in (2.2), that is, the smallest approximate eigenvalues of A. Hence, we use the same harmonic Ritz formulation for augmentation. We then compare the convergence rates of our AugGMRES in mixed precision with GMRES and GMRES-DR in uniform fp64.

Experimental setting. We consider various sparse matrices: random synthetic ones, matrices taken from the SuiteSparse collection [20], and a matrix arising in a computational fluid dynamics (CFD) simulation from ONERA (see subsection 5.3 for details).

We explore various precision combinations. As a natural choice for the working precision, we set u to fp64. Moreover, we take $u_r = u$ and $u_f \leq u_p$. We will investigate various combinations for u_p, u_o, u_a, u_e chosen among fp64, fp32, and fp16. In our experiments, using fp16 for u_o lead to stagnation for both standard and augmented GMRES, even for well-conditioned matrices, so we will only use fp32 or higher for u_o. This may be due to the use of right-preconditioning, which the recent analysis of Buttari et al. [13] shows to be more sensitive to the orthonormalization precision

Table 5.1
Configuration of linear solvers for each matrix with condition number

Matrix	Origin	$\kappa(A)$	n	m	k	Preconditioner
Random1	Synthetic	10^3	1000	20	4	Jacobi
Random2	Synthetic	10^5	1000	20	4	Jacobi
fv3	SuiteSparse	$O(10^3)$	9801	10	2	ILU(0)
SiO2	SuiteSparse	$O(10^4)$	155 331	40	8	ILU(0)
LS89	CFD (ONERA)	$O(10^{14})$	115 368	60	5	Block diagonal LU

Table 5.2
Configuration of mixed precision levels

Matrix	u_f	u_p	u_e	u_a	u_o
Random1	fp16,32	u_f	fp16,32	fp16,32	fp32
Random2	fp16,32	u_f	fp16,32	fp32	fp32
fv3	fp16	u_f	fp16,32	fp16,32	fp32
SiO2	fp32	u_f	fp16,32	fp16,32	fp32
LS89	fp32	fp16,32	fp16,32	fp32	fp32

than left-preconditioning. The use and study of left-preconditioning is outside of the scope of this paper.

Depending on A, we employ different types of preconditioning and set appropriate values for m and k, but we maintain a fixed tolerance of $\varepsilon = 10^{-10}$ for all experiments. The solver and precision configurations are indicated in Tables 5.1 and 5.2, respectively.

5.1. Random synthetic matrices. We begin by testing our algorithm on random synthetic sparse matrices generated using the MATLAB command `sprand`. To ensure A is invertible, we impose diagonal dominance and adjust the singular value distribution. This allows us to generate random sparse matrices with specific condition numbers. As indicated in Tables 5.1 and 5.2, we solve two linear systems with Jacobi preconditioning in mixed precision.

In Figure 5.1, we plot the convergence history of GMRES, GMRES-DR, and AugGMRES in uniform fp64 precision, as well as various mixed precision configurations of AugGMRES. Comparing the uniform fp64 variants first confirms that AugGMRES significantly enhances the convergence of GMRES and matches that of GMRES-DR (the curves for uniform fp64 AugGMRES and GMRES-DR are not visible because they perfectly overlap with the other curves in the fastest convergence group). Moreover, AugGMRES remains successful even when lower precision arithmetic is employed. In particular, using fp32 arithmetic for all precisions except u and u_r (purple upward triangles) preserves the same convergence as the uniform fp64 variant. Furthermore, even fp16 arithmetic can be used for selected operations; for this example, fp16 preconditioning ($u_f = u_p =$ fp16, red circles) and fp16 matrix–vector products ($u_a =$ fp16, cyan leftward triangles) both perform comparably to uniform fp64. Using fp16 for computing the harmonic Ritz vectors ($u_e =$ fp16, green downward triangles) causes a minor degradation in convergence. Using fp16 for all precisions except u_o (blue diamonds) leads to a more noticeable, but still small degradation, which suggests that the errors have a cumulative effect on the convergence.

Figure 5.2 shows a similar experiment but with a more ill conditioned matrix. For this matrix, uniform precision fp64 GMRES stagnates at about 10^{-6} relative

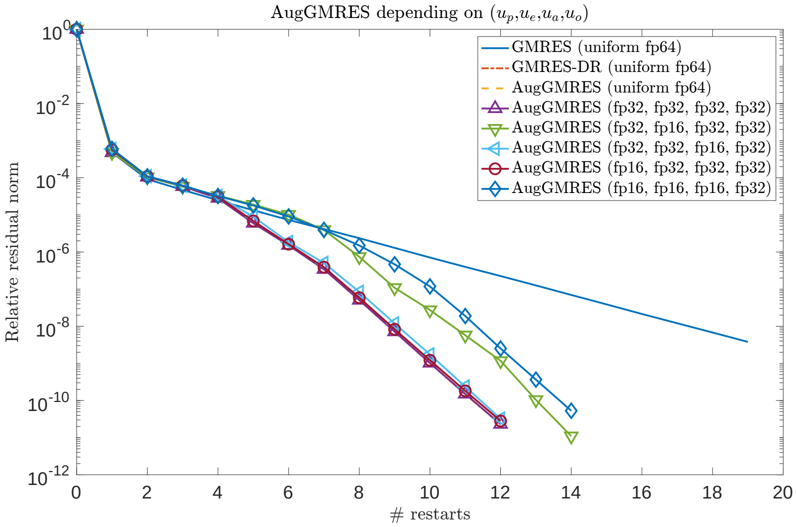

FIG. 5.1. *Random1 matrix: relative residual norm for the case of $\kappa(A) = 10^3$ depending on precision configurations (u_p, u_e, u_a, u_o) combinations.*

residual norm, whereas AugGMRES successfully converges to the target accuracy. The figure further illustrates that there is still potential for mixed precision despite the ill conditioning of the problem: using fp32 for all operations except u and u_r still leads to successful convergence thanks to augmentation, although slightly later than in the uniform fp64 case (15 restarts instead of 12). Here, however, the use of fp16 is not recommendable: using fp16 for u_e or u_a loses the advantage of augmentation and leads to the same stagnation as standard GMRES. Using fp16 for preconditioning (u_f and u_p), the algorithm fails to converge due to the numerical singularity in the Jacobi preconditioner. These results highlight the robustness of our proposed mixed precision approach on ill-conditioned problems, provided that appropriate precision levels are selected for critical steps.

5.2. SuiteSparse Matrix Collection. We complement the previous experiments on synthetic matrices with some additional experiments on two matrices from the SuiteSparse Matrix Collection: fv3 and SiO2. For each matrix, the solver parameter settings are indicated in Tables 5.1 and 5.2.

As previously observed in subsections 4.1 and 4.2, employing lower precision computations does not degrade the convergence of GMRES when solving the fv3 system. Figure 5.3 shows that this observation extends to AugGMRES, which exhibits significantly improved convergence compared with standard GMRES, even with an intensive usage of lower precision. Specifically, fp16 can be used for u_f, u_p, u_e, and u_a; only u_o requires fp32. Thus, unlike (F)GMRES-DR, AugGMRES can use lower precision u_p without requiring the flexible variant and, more importantly, can also use lower precision u_e, u_a, and u_o while avoiding stagnation and successfully improving the convergence of standard GMRES.

Figure 5.4 illustrates similar numerical behaviors for mixed precision AugGMRES with matrix SiO2, which is slightly more ill-conditioned and much larger than fv3. A notable difference here is that employing lower precision computations degrades the convergence, but this is the case for both GMRES and AugGMRES, so the issue is independent of augmentation. In fact, the use of augmentation always improves the

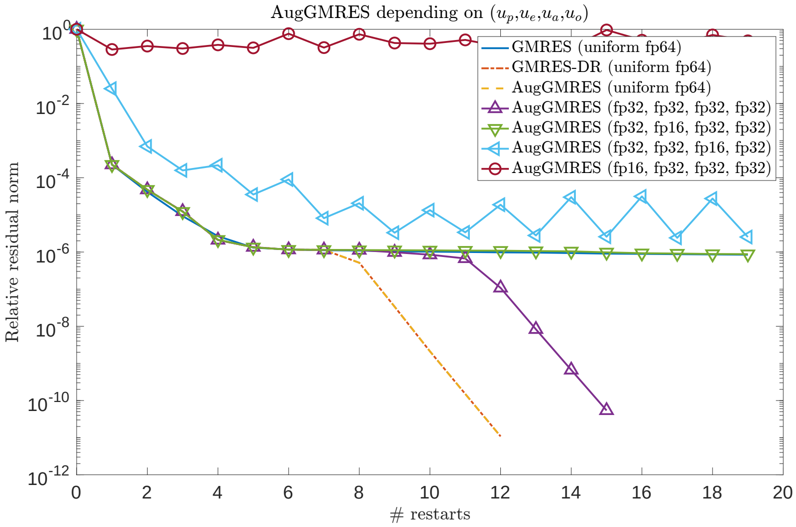

FIG. 5.2. *Random2 matrix: relative residual norm for the case of* $\kappa(A) = 10^5$ *depending on precision configurations* (u_p, u_e, u_a, u_o) *combinations.*

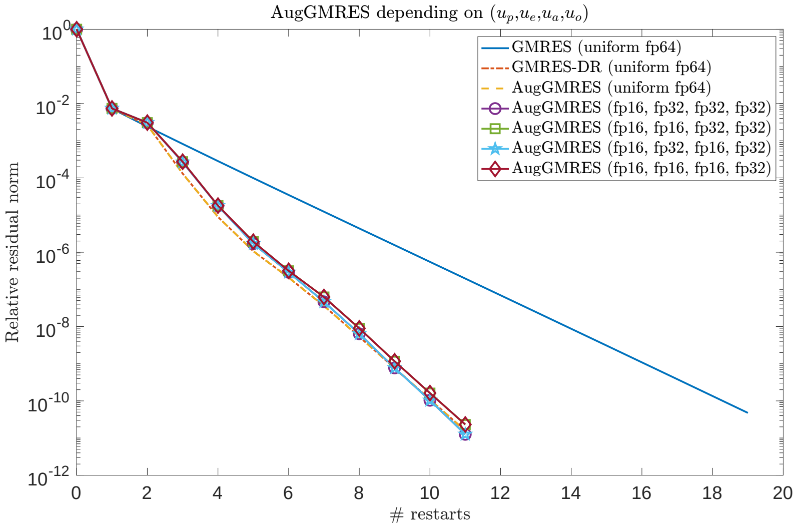

FIG. 5.3. *fv3 matrix: relative residual norm depending on precision configurations* (u_p, u_e, u_a, u_o) *combinations.*

convergence rate, regardless of the mixed precision configuration. Moreover, mixed precision augmented GMRES converges faster than uniform fp64 GMRES, which shows that the loss of convergence induced by the use of mixed precision can be more than compensated by augmentation.

5.3. CFD simulation. As our final numerical experiment, we consider the LS89 test case [8], a well-known benchmark in computational fluid dynamics (CFD) for analyzing transonic flow in high-pressure turbine blades. The LS89 turbine cascade, originally studied in experimental settings, serves as a standard test case for evaluating numerical solvers applied to compressible Navier–Stokes equations. This test case

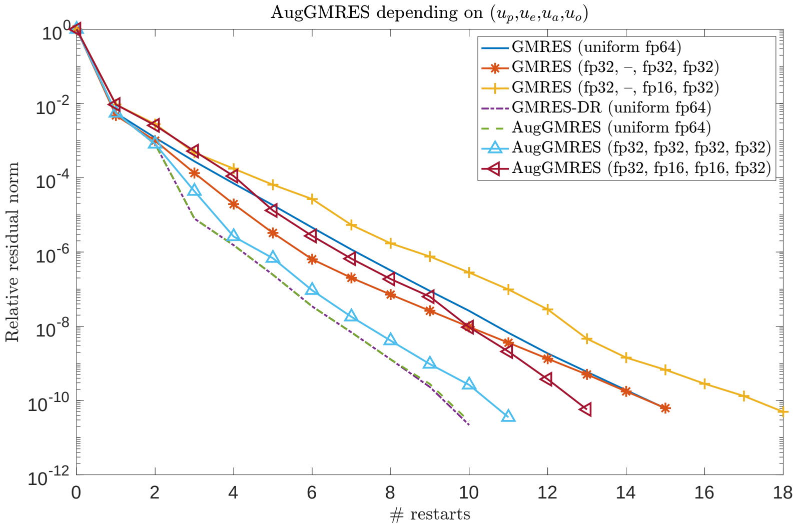

FIG. 5.4. *SiO2 matrix: relative residual norm depending on precision configurations* (u_p, u_e, u_a, u_o) *combinations.*

presents a challenging scenario due to strong pressure gradients, shock waves, and boundary layer effects, making it a suitable candidate for assessing the performance of mixed precision iterative solvers.

The linear system arising from the LS89 problem is characterized by a large sparse nonsymmetric matrix with $n = 115\,368$ and $\kappa(A) = O(10^{14})$, which is thus extremely ill conditioned. We employ a block diagonal LU preconditioner with a six blocks of equal size $n/6$, computed in u_f = fp32. However, we experiment with various precision levels u_p for applying this preconditioner.

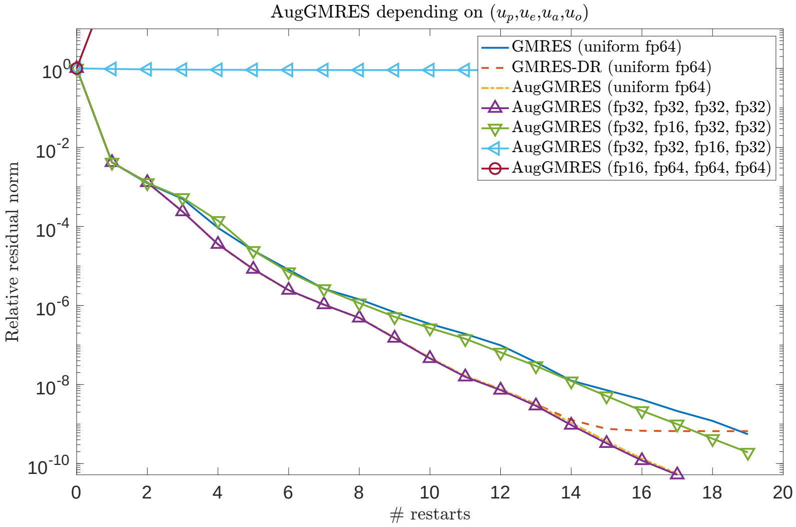

FIG. 5.5. *LS89 matrix: relative residual norm depending on precision configurations* (u_p, u_e, u_a, u_o) *combinations.*

Even for this ill-conditioned problem, Figure 5.5 demonstrates that lower precision arithmetic can be used. Here, the use of fp16 is again not recommendable, since it prevents convergence when used for preconditioning (u_p) or matrix–vector products (u_a); if fp16 is used for solving the eigenvalue problem (u_e), AugGMRES converges but the benefit of augmentation is almost entirely lost. However, while fp16 should not be used, fp32 can be used for all these operations and does not degrade the convergence of AugGMRES. Interestingly, for this problem, AugGMRES (in uniform or mixed precision) exhibits better convergence than GMRES-DR in uniform fp64, which eventually stagnates at around 10^{-9}. This stagnation is likely caused by the same loss of collinearity discussed in subsection 4.3, which illustrates that this issue is related to the use of finite (not just mixed) precision and can occur even in a uniform fp64 setting. In any case, this experiment shows that, despite the extremely ill-conditioned nature of the system, appropriately chosen mixed precision configurations can be combined with augmentation.

6. Conclusion.

We have proposed a mixed precision augmented GMRES algorithm that successfully combines the benefits of augmentation and mixed precision.

We have presented a general framework that uses independent precision parameters for each of the main kernels, and that considers a generic augmented subspace. For the latter, in practice, we have focused on using approximate eigenvectors estimated via the harmonic Ritz formulation. One important point is that the simplifications operated to derive the popular GMRES-DR method should be avoided in mixed precision, because they rely on a collinearity property that does not hold in finite precision. While this specific GMRES-DR variant thus presents limited mixed precision opportunities, we have shown that the more general augmented GMRES method can make use of lower precisions, such as fp32 or even fp16, for computationally intensive kernels like preconditioning, matrix–vector product, eigenvalue decomposition, and orthonormalization, and still successfully converge, often at the same speed as uniform precision fp64 augmented GMRES (or GMRES-DR). We have experimentally illustrated the benefits of our mixed precision augmented GMRES approach for various sparse matrices, including ill-conditioned real-life ones.

Given the promising numerical results obtained by this study, developing a high-performance parallel implementation of this mixed precision augmented GMRES approach seems an interesting perspective for future work.

Acknowledgments. We thank Emeric Martin for several discussions on the limitations of mixed precision GMRES-DR.

Funding. This work was partially supported by projects of the French National Agency for Research (ANR): InterFLOP (ANR-20-CE46-0009), NumPEx Exa-MA (ANR-22-EXNU-0002), and MixHPC (ANR-23-CE46-0005-01).

REFERENCES

[1] A. Abdelfattah, H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, A. Fox, M. Gates, N. J. Higham, X. S. Li, J. Loe, P. Luszczek, S. Pranesh, S. Rajamanickam, T. Ribizel, B. F. Smith, K. Swirydowicz, S. Thomas, S. Tomov, Y. M. Tsai, and U. M. Yang, *A survey of numerical linear algebra methods utilizing mixed-precision arithmetic*, Int. J. High Perform. Comput. Appl., 35 (2021), pp. 344–369, https://doi.org/10.1177/10943420211003313.

[2] J. I. Aliaga, H. Anzt, T. Grützmacher, E. S. Quintana-Ortí, and A. E. Tomás, *Compressed basis GMRES on high-performance graphics processing units*, Int. J. High Perform. Comput. Appl., 37 (2023), pp. 82–100.

[3] P. R. Amestoy, A. Buttari, N. J. Higham, J.-Y. L'Excellent, T. Mary, and B. Vieublé, *Combining sparse approximate factorizations with mixed precision iterative refinement*, ACM Trans. Math. Software, 49 (2023), https://doi.org/10.1145/3582493.

[4] P. R. Amestoy, A. Buttari, N. J. Higham, J.-Y. L'Excellent, T. Mary, and B. Vieublé, *Five-precision GMRES-based iterative refinement*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 529–552, https://doi.org/10.1137/23M1549079.

[5] P. R. Amestoy, A. Jego, J.-Y. L'Excellent, T. Mary, and G. Pichon, *BLAS-based block memory accessor with applications to mixed precision sparse direct solvers*, https://hal.science/hal-05019106. HAL EPrint hal-05019106.

[6] H. Anzt, J. Dongarra, G. Flegar, N. J. Higham, and E. S. Quintana-Ortí, *Adaptive precision in block-Jacobi preconditioning for iterative sparse linear system solvers*, Concurrency Computat. Pract. Exper., 31 (2019), p. e4460, https://doi.org/10.1002/cpe.4460.

[7] M. Arioli and I. S. Duff, *Using FGMRES to obtain backward stability in mixed precision*, Electron. Trans. Numer. Anal, 33 (2009), pp. 31–44.

[8] T. Arts and M. L. De Rouvroit, *Aero-thermal performance of a two dimensional highly loaded transonic turbine nozzle guide vane: A test case for inviscid and viscous flow computations*, vol. 79047, American Society of Mechanical Engineers, 1990.

[9] J. Baglama and L. Reichel, *Augmented GMRES-type methods*, Numer. Linear Algebra Appl., 14 (2007), pp. 337–350.

[10] A. H. Baker, E. R. Jessup, and T. Manteuffel, *A technique for accelerating the convergence of restarted GMRES*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 962–984.

[11] Å. Björck and C. C. Paige, *Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 176–190.

[12] A. Buttari, N. J. Higham, T. Mary, and B. Vieublé, *A modular framework for the backward error analysis of GMRES*, https://hal.science/hal-04525918. HAL EPrint hal-04525918; to appear in *IMA J. Numer. Anal.*.

[13] A. Buttari, X. Liu, T. Mary, and B. Vieublé, *Mixed precision strategies for preconditioned GMRES: a comprehensive analysis*, https://hal.science/hal-05071696. HAL EPrint hal-05071696.

[14] E. Carson and I. Daužickaite, *The stability of split-preconditioned FGMRES in four precisions*, Electron. Trans. Numer. Anal., 60 (2024), p. 40–58, https://doi.org/10.1553/etna_vol60s40.

[15] E. Carson and N. J. Higham, *A new analysis of iterative refinement and its application to accurate solution of ill-conditioned sparse linear systems*, SIAM J. Sci. Comput., 39 (2017), pp. A2834–A2856, https://doi.org/10.1137/17M1122918.

[16] E. Carson and N. J. Higham, *Accelerating the solution of linear systems by iterative refinement in three precisions*, SIAM J. Sci. Comput., 40 (2018), pp. A817–A847, https://doi.org/10.1137/17M1140819.

[17] L. M. Carvalho, S. Gratton, R. Lago, and X. Vasseur, *A flexible generalized conjugate residual method with inner orthogonalization and deflated restarting*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1212–1235.

[18] O. Coulaud, L. Giraud, P. Ramet, and X. Vasseur, *Deflation and augmentation techniques in Krylov linear solvers*, Research Report RR-8265, INRIA, Feb. 2013, https://inria.hal.science/hal-00803225. Preliminary version of the book chapter entitled ”Deflation and augmentation techniques in Krylov linear solvers” published in ”Developments in Parallel, Distributed, Grid and Cloud Computing for Engineering”, ed. Topping, B.H.V and Ivanyi, P., Saxe-Coburg Publications, Kippen, Stirlingshire, United Kingdom, ISBN 978-1-874672-62-3, p. 249-275, 2013.

[19] H. A. Daas, L. Grigori, P. Hénon, and P. Ricoux, *Recycling Krylov subspaces and truncating deflation subspaces for solving sequence of linear systems*, ACM Trans. Math. Software, 47 (2021), pp. 1–30.

[20] T. A. Davis and Y. Hu, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), pp. 1:1–1:25, https://doi.org/10.1145/2049662.2049663.

[21] M. Embree, *How descriptive are GMRES convergence bounds?*, arXiv preprint arXiv:2209.01231, (2022).

[22] L. Giraud, S. Gratton, and J. Langou, *Convergence in backward error of relaxed GMRES*, SIAM J. Sci. Comput., 29 (2007), pp. 710–728, https://doi.org/10.1137/040608416.

[23] L. Giraud, S. Gratton, X. Pinel, and X. Vasseur, *Flexible GMRES with deflated restarting*, SIAM J. Sci. Comput., 32 (2010), pp. 1858–1878.

[24] L. Giraud, A. Haidar, and L. T. Watson, *Mixed-precision preconditioners in parallel domain decomposition solvers*, in Domain decomposition methods in science and engineering XVII, Springer, 2008, pp. 357–364.

[25] S. Graillat, F. Jézéquel, T. Mary, and R. Molina, *Adaptive precision sparse matrix-vector product and its application to Krylov solvers*, SIAM J. Sci. Comput., 46 (2024), pp. C30–C56, https://doi.org/10.1137/22M1522619.

[26] S. Gratton, E. Simon, D. Titley-Peloquin, and P. L. Toint, *A note on inexact inner products in GMRES*, SIAM J. Matrix Anal. Appl., 43 (2022), pp. 1406–1422, https://doi.org/10.1137/20M1320018.

[27] T. Grützmacher, H. Anzt, and E. S. Quintana-Ortí, *Using Ginkgo's memory accessor for improving the accuracy of memory-bound low precision BLAS*, Software—Practice and Experience, (2021), https://doi.org/10.1002/spe.3041.

[28] N. J. Higham and T. Mary, *Mixed precision algorithms in numerical linear algebra*, Acta Numerica, 31 (2022), pp. 347–414, https://doi.org/10.1017/s0962492922000022.

[29] N. J. Higham and S. Pranesh, *Simulating low precision floating-point arithmetic*, SIAM J. Sci. Comput., 41 (2019), pp. C585–C602, https://doi.org/10.1137/19M1251308.

[30] Y. Jang, L. Grigori, E. Martin, and C. Content, *Randomized flexible GMRES with deflated restarting*, Numer. Algorithms, 98 (2025), pp. 431–465, https://doi.org/10.1007/s11075-024-01801-3.

[31] N. Lindquist, P. Luszczek, and J. Dongarra, *Accelerating restarted GMRES with mixed precision arithmetic*, IEEE Trans. Parallel Distrib. Syst., 33 (2022), pp. 1027–1037, https://doi.org/10.1109/tpds.2021.3090757.

[32] J. A. Loe, C. A. Glusa, I. Yamazaki, E. G. Boman, and S. Rajamanickam, *Experimental evaluation of multiprecision strategies for GMRES on GPUs*, in 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2021, pp. 469–478, https://doi.org/10.1109/IPDPSW52791.2021.00078.

[33] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.

[34] R. B. Morgan, *Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1112–1135, https://doi.org/10.1137/S0895479897321362, https://doi.org/10.1137/S0895479897321362, https://arxiv.org/abs/https://doi.org/10.1137/S0895479897321362.

[35] R. B. Morgan, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.

[36] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen, *How fast are nonsymmetric matrix iterations?*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 778–795.

[37] E. Oktay and E. Carson, *Mixed precision GMRES-based iterative refinement with recycling*, Programs and Algorithms of Numerical Mathematics, (2023), pp. 149–162.

[38] M. L. Parks, E. De Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674.

[39] R.-R. Wang, Q. Niu, X.-B. Tang, and X. Wang, *Solving shifted linear systems with restarted GMRES augmented with error approximations*, Computers & Mathematics with Applications, 78 (2019), pp. 1910–1918.

[40] Y. Zhao, T. Fukaya, L. Zhang, and T. Iwashita, *Numerical investigation into the mixed precision GMRES (m) method using fp64 and fp32*, Journal of Information Processing, 30 (2022), pp. 525–537, https://doi.org/https://doi.org/10.2197/ipsjjip.30.525.