

Adaptive Improvements to the LGMRES(m, l) Method for Efficient Resolution of Sparse Linear Systems in Computational Fluid Dynamics

Juan C. Cabral¹, Gustavo E. Espínola¹, Christian E. Schaerer¹, Jhabriel Varela^{1,2}.

¹ *Grupo de Investigación en Computación Científica y Matemática Aplicada - CCyMA*

Polytechnic School, National University of Asuncion

Campus Universitario, 2111 SL, San Lorenzo, Paraguay

jccabral@pol.una.py, gustavoespínola@fpuna.edu.py, cschaer@pol.una.py, jhabriel@pol.una.py

² *Polytechnic University Taiwan-Paraguay*

Sebastian Gaboto, 1018 AS, Asunción, Paraguay

Abstract. Accurate and efficient solution of sparse linear systems arising from computational fluid dynamics (CFD) simulations is a computationally intensive task. Traditional iterative methods such as GMRES(m) often suffer from stagnation when applied to large, nonsymmetric matrices derived from discretized fluid-dynamics models. To address this limitation, a modified version of GMRES(m), known as LGMRES(m, l), was developed to incorporate additional error vectors into the search space. However, previous studies have focused mainly on increasing the subspace dimension m while keeping the number of error vectors l fixed. This paper proposes PD-full LGMRES, which adaptively updates the restart size m and retains error-approximation vectors that contribute to residual reduction. At each restart, m is set by a discrete proportional-derivative controller. The proposed adaptive LGMRES is tested on matrices arising from CFD applications. Benchmark matrices from the SuiteSparse Matrix Collection specific to these applications are used to assess the robustness and efficiency of the adaptive LGMRES method in comparison with its static counterpart and other iterative solvers. Numerical results obtained on benchmark matrices from environmental fluid-dynamics applications confirm the effectiveness of the proposed adaptive LGMRES.

Keywords: Computational Fluid Dynamics, LGMRES, Adaptive Methods, Sparse Linear Systems, Computational Efficiency.

1 Introduction

Computational Fluid Dynamics (CFD) is crucial for modeling complex environmental phenomena, such as contaminant transport in shallow water [1–3]. These simulations often give rise to large-scale, sparse, and non-symmetric linear systems that are computationally expensive to solve. Iterative methods based on Krylov subspaces, such as the Generalized Minimal Residual (GMRES) method [4], are essential for this task. However, the substantial memory requirements of full GMRES have led to restarted variants like GMRES(m), which limit the subspace dimension. While effective, the performance of GMRES(m) can degrade if the restart parameter m is poorly chosen, motivating the development of strategies to adaptively select it [5–7].

To further improve convergence, the search subspace can be augmented with information from previous restarts. The LGMRES(m, l) method, introduced by Baker et al. [8], enhances GMRES(m) by augmenting the standard Krylov subspace with a set of historical correction vectors. Nonetheless, selecting static values for both m and l can still be suboptimal for ill-conditioned CFD problems. To address this challenge, this work proposes PD-full LGMRES, an adaptive variant that dynamically adjusts the restart parameter m using a proportional-derivative controller while retaining all previous error-approximation vectors to enrich the search space.

This paper is organized as follows. In §2, we introduce the proposed method. In §3, the numerical results and analysis and discussion of convergence figures. The conclusions are presented at §4 showing that the adaptive strategy improves the convergence of standard restart versions of GMRES.

2 Proposed Method: PD–full LGMRES—An Adaptive Augmented Variant of GMRES(m)

The restarted Generalized Minimal Residual Method, popularly known as GMRES(m) proposed by Saad [4] is a Krylov subspace method for solving large sparse non-symmetric linear systems. Let $A \in \mathbb{R}^{n \times n}$ be a nonsingular matrix, $b \in \mathbb{R}^n$ the right-hand side vector, and x_j the approximation to the solution at the j -th restart cycle. The residual at this cycle is given by $r_j = b - Ax_j$. GMRES(m) approximates the solution to linear system at the $j + 1$ -th restart cycle using the residual at j -th cycle r_j for constructing a Krylov subspace of $\mathcal{K}_m(A, r_j)$ of dimension m . From this, it follows that the $j + 1$ -th approximation satisfies

$$x_{j+1} \in x_j + \mathcal{K}_m(A, r_j), \quad (1)$$

with m denoting the (usually constant) value of the restarting parameter. Let $r_j := b - Ax_j$ be the residual vector of the j -th approximation. Then, GMRES(m) seeks for an approximate solution that minimizes the 2-norm of the residual

$$\min_{x_{j+1} \in x_j + \mathcal{K}_m(A, r_j)} \|b - Ax_{j+1}\|_2. \quad (2)$$

In the *full* LGMRES method, all previously computed approximation error vectors are retained and used to enrich the search space at each restart cycle. The approximation error vector at cycle j is defined as $z_j := x_j - x_{j-1}$. The resulting search space at restart cycle $j + 1$ is then given by

$$\mathcal{M}_{j+1} = \mathcal{K}_m(A, r_j) \cup \text{span}\{z_1, z_2, \dots, z_j\},$$

where $\mathcal{K}_m(A, r_j)$ denotes the m -dimensional Krylov subspace generated by A and r_j . The *full* LGMRES scheme can significantly enhance convergence compared to standard restarted GMRES; it entails growing memory and orthogonalization costs as the number of cycles increases. For ill-conditioned problems, highly nonnormal, or stagnating problems, full LGMRES can help uncover convergence behaviors that may be masked in shorter-memory methods. For instance, it can reveal whether past approximation vectors retain valuable directions or whether redundancy dominates as iterations grow. Insights gained from analyzing the contribution of all z_j vectors can inform the development of adaptive truncation strategies. For example, one might retain only those z_j vectors with the greatest orthogonality or variance, or apply weighting schemes to prioritize their inclusion in the augmented subspace.

In summary, the practical analysis of full LGMRES is not only relevant for benchmarking purposes but also provides valuable insights into Krylov subspace augmentation strategies. It can guide the design of more effective iterative solvers by highlighting the impact of memory retention on convergence behavior.

2.1 PD-GMRES(m)

Classical GMRES(m) often suffers from stagnation when the chosen restart parameter m is inadequate for certain stages of the iterative process. PD-GMRES(m), by Cuevas et al. [5], overcomes this drawback by adaptively modifying m in response to residual trends, thereby enhancing convergence while preserving the original structure of the GMRES scheme. The update formula for the restart parameter becomes:

$$m_j = m_{j-1} + \left\lceil \alpha_P \frac{\|r_j\|}{\|r_{j-1}\|} + \alpha_D \frac{\|r_j\| - \|r_{j-2}\|}{2\|r_{j-1}\|} \right\rceil, \quad \alpha_P, \alpha_D \in \mathbb{R}, \quad (3)$$

where r_j is the residual at the end of the j -th restart cycle, α_P is the proportional gain, α_D is the derivative gain, and $\lceil \cdot \rceil : \mathbb{R} \rightarrow \mathbb{N}$, denotes rounding to the nearest integer.

The proportional term measures the immediate change in residual norm, while the derivative term captures the trend (acceleration or deceleration) of residual reduction. This results in a more responsive adaptation of m , especially effective when the method exhibits oscillatory or slowly decreasing convergence patterns.

2.2 PD - full LGMRES(m_j)

In this section, we propose a combined adaptive scheme that integrates the strengths of full LGMRES and PD-GMRES in a unified framework, denoted as PD - full LGMRES(m_j). The method simultaneously adjusts the restart parameter m_j based on residual behavior, and retains informative error approximation vectors to enrich the Krylov subspace.

Let $z_j = x_j - x_{j-1}$ denote the error approximation vector at restart cycle j , and $r_j = b - Ax_j$ the residual at cycle j . The next approximation is defined as

$$x_{j+1} = x_j + q_{m_j-1}(A)r_j + \sum_{k=1}^j \alpha_{ik} z_k,$$

where $q_{m_j-1}(A)$ is a degree- $m_j - 1$ GMRES polynomial generated from r_j and $\alpha_{ik} \in \mathbb{R}$ are scalar coefficients obtained by solving a least-squares problem.

The restart parameter m_j is updated dynamically using a discrete proportional-derivative (PD) controller according to Equation (3). The values used for the proportional and derivative gains in the PD controller were chosen to match those reported in Cabral et al.[9], where $\alpha_P = 2$ and $\alpha_D = 0.8$ were empirically found to provide a good balance between responsiveness and stability across a variety of test cases. These same values are adopted in this work to ensure consistency and comparability with the results presented in that previous study. Note, however, that these choices might not be optimal. Finding optimal values for the PD controller is a non-trivial task that might benefit from the use of data-driven techniques [7].

3 Numerical Results

We tested the proposed PD - *full* LGMRES(m_j) method on six benchmark matrices from the SuiteSparse Matrix Collection [10], all arising from CFD and related applications. All matrices are non-symmetric and large-scale. We used a zero initial guess and terminated iterations when the relative residual satisfied

$$\frac{\|r_j\|}{\|r_0\|} \leq 10^{-9},$$

or when the number of cycles exceeded 1000. Simulations were executed using MATLAB® R2022a for Windows 11. To solve the sparse linear systems, we employed the implemented solvers from the KrySBAS library [11]. Numerical experiments are available for reproduction at <https://github.com/jccf19/PD-fLGMRES>.

Table 1. Selected Matrices from the SuiteSparse Matrix Collection [10].

Id	Problem	n	nnz	Condition Number	Application Area
1	sherman5	3 312	20 793	1.88×10^5	Computational fluid dynamics
2	sherman2	1 080	23 094	9.60×10^{11}	Computational fluid dynamics
3	Raefsky1	3 242	293 409	1.29×10^4	Computational fluid dynamics problem sequence
4	Raefsky2	3 242	293 409	1.29×10^4	Subsequent computational fluid dynamics problem
5	Cavity05	1182	32 632	5.77×10^5	Computational fluid dynamics problem sequence
6	Cavity10	2597	76 171	1.29×10^4	Computational fluid dynamics

Table 1 presents a selection of six sparse matrices extracted from the SuiteSparse Matrix Collection [10]. These matrices were chosen to evaluate the robustness and scalability of the proposed adaptive LGMRES algorithm in the context of CFD and related fields. All matrices are non-singular and represent challenging test cases due to their high condition numbers, non-symmetric structure, or both. The matrices vary in size, sparsity, and physical origin, providing a comprehensive testbed for assessing convergence behavior and computational performance.

The convergence results presented in Figures 1–3 illustrate the performance differences between the compared methods: GMRES(30), LGMRES(28,2), PD-GMRES, and PD-fLGMRES. The analysis focuses on three representative matrices: sherman2, cavity05, and raefsky2, each exhibiting distinct numerical challenges.

In Figure 1, both GMRES(30) and LGMRES(28,2) fail to converge within the maximum allowed 1000 restart cycles, reflecting the extreme numerical difficulty associated with the matrix’s very high condition number ($\kappa \approx 9.6 \times 10^{11}$). In contrast, PD-GMRES and PD-fLGMRES achieve convergence in 260 cycles, with nearly overlapping convergence curves. This behavior suggests that, for this particular matrix, the dynamic adjustment of the restart parameter m is sufficient to overcome stagnation, and the additional augmentation of the search space with historical error vectors in PD-fLGMRES provides no noticeable improvement. A plausible explanation is that the adaptive control of m rapidly captures the most effective search direction, thereby reducing the relative benefit of retaining an extended set of historical vectors.

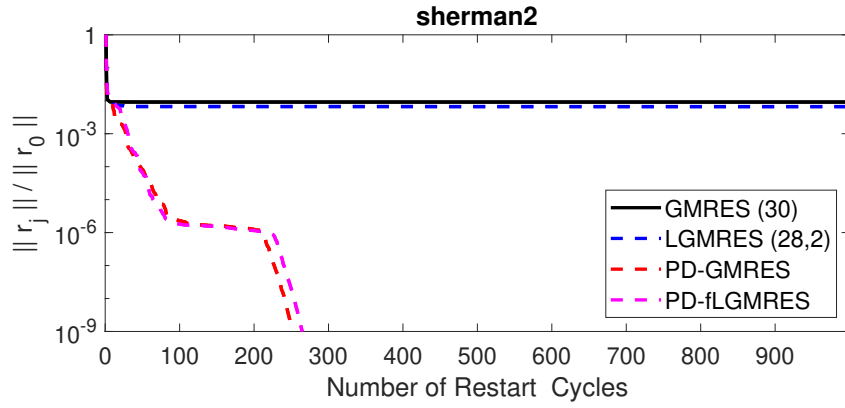


Figure 1. The convergence curves of matrix sherman2.

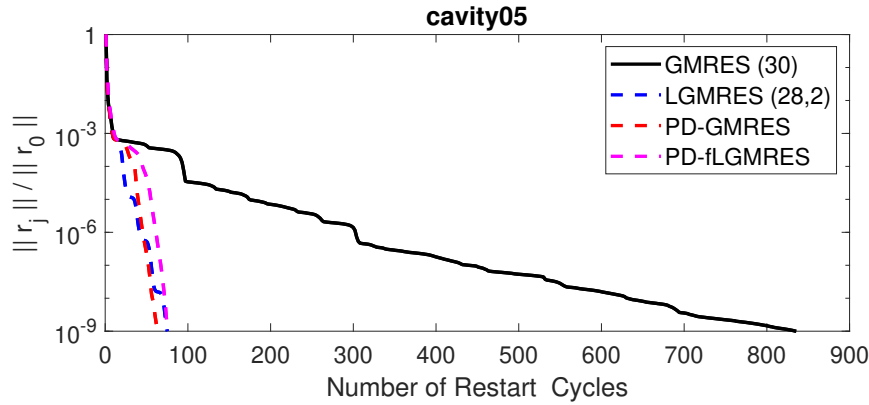


Figure 2. The convergence curves of matrix cavity05.

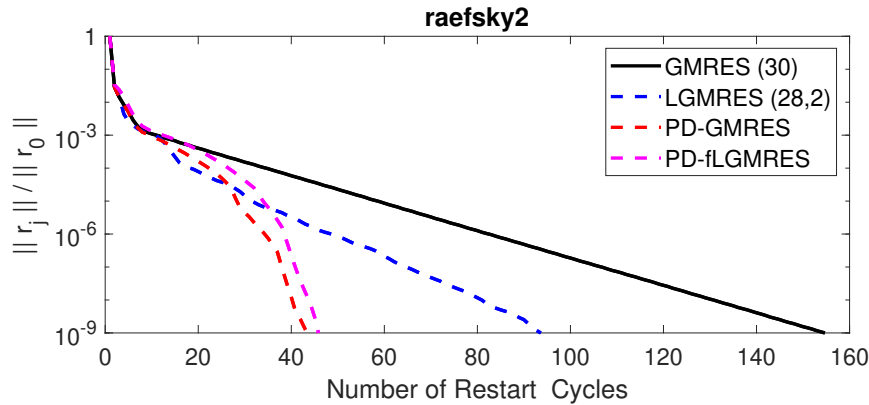


Figure 3. The convergence curves of matrix raefsky2.

In Figure 2, GMRES(30) requires approximately 400 cycles to converge, while LGMRES(28,2) dramatically reduces this to 80 cycles. Both PD-GMRES and PD-fLGMRES maintain similar performance (80–85 cycles) with rapid and stable convergence. Here, the classical LGMRES already achieves near-optimal performance, which limits the potential for further improvement by adaptive strategies. The slightly higher cycle count of PD-fLGMRES compared to LGMRES(28,2) may be attributed to adjustments in m that do not offer an advantage when the static parameter is already well chosen. This indicates that, for moderately difficult matrices, adaptivity may not outperform a well-tuned static configuration.

In Figure 3, GMRES(30) requires around 160 cycles to converge, while LGMRES(28,2) improves this to 90. PD-GMRES and PD-fLGMRES further reduce the iteration count to approximately 45–50 cycles. The adaptive methods exhibit steeper residual reductions and fewer plateau phases. In this case, the combination of error vector retention and dynamic m adaptation yields a performance gain, with PD-fLGMRES producing a slightly smoother convergence curve than PD-GMRES. Although the iteration counts are similar, the increased stability

and predictability of convergence can be advantageous, particularly in sequential problem settings where consistent behavior is desirable.

Table 2. Number of restart cycles until convergence (excluding PD-ALGMRES)

Matrix	GMRES(30)	LGMRES(28,2)	PD-GMRES	PD-fLGMRES
Sherman5	1000	1000	90	110
Sherman2	1000	1000	260	260
Raefsky1	140	30	40	48
Raefsky2	160	90	45	50
Cavity05	400	80	80	85
Cavity10	1000	100	100	110

The numerical data in Table 2 reinforces the trends observed in the convergence plots and provide a quantitative measure of performance improvements. For highly ill-conditioned problems such as Sherman5 and Sherman2, both GMRES(30) and LGMRES(28,2) fail to converge within the iteration limit, while the adaptive variants achieve substantial reductions: PD-GMRES converges in 90 cycles for Sherman5, representing a 91% reduction compared to the maximum limit, and in 260 cycles for Sherman2 (74% reduction). In Sherman5, PD-GMRES outperforms PD-fLGMRES (90 vs. 110 cycles), suggesting that in some cases the computational overhead of extended memory may offset its benefits.

For moderately difficult problems such as Raefsky1 and Raefsky2, LGMRES(28,2) already achieves significant gains (78.6% and 43.8% reductions, respectively), but the adaptive methods further improve convergence. PD-GMRES reduces the cycle count for Raefsky1 by 71.4% compared to GMRES(30), while PD-fLGMRES achieves a 65.7% reduction. In Raefsky2, PD-GMRES and PD-fLGMRES yield reductions of 71.9% and 68.8% respectively. Finally, for Cavity05 and Cavity10, LGMRES(28,2) already offers large reductions (80% and 90%), leaving little scope for additional gains. The adaptive methods produce similar results, with differences below 10 cycles. Overall, the table confirms that PD-fLGMRES provides the most pronounced benefits when the problem exhibits alternating phases of fast and slow convergence, while its advantage over PD-GMRES may diminish in extremely ill-conditioned cases or when the static configuration is already near-optimal.

4 Conclusion

Generally, PD-fLGMRES offers selective advantages, especially for problems where classical LGMRES already improves over GMRES and where convergence patterns alternate between fast and slow phases. For extremely ill-conditioned systems such as sherman2, its performance aligns closely with PD-GMRES. The adaptive approaches reduce the dependency on prior parameter tuning, delivering robust performance without extensive calibration. From a practical perspective, the choice between PD-GMRES and PD-fLGMRES should weigh the additional storage and orthogonalization cost against the potential benefits in convergence smoothness. In CFD applications involving problem sequences, PD-fLGMRES may be preferred to maintain consistent performance across simulations. Future work should examine the time and memory requirements under preconditioning and explore criteria for selectively retaining error vectors.

Acknowledgements. The authors acknowledge funding from the Polytechnic Faculty (FPUNA) through its Research Department. This work was also supported by the Paraguayan National Council of Science and Technology (CONACYT) through the ARASY Strategic Project "Strengthening Computational Mathematical Modeling Capacities for Earth and Life" (ESTR01-23). G.E.E. was funded by CONACYT through the Incentive for the Training of Researchers in National Postgraduate Programs (BCAS06-15), and J.V. was funded by CONACYT through the program for Advanced Human Capital Insertion in Academia (PRIA01-8).

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

- [1] L. C. Oporto, D. A. Ramírez, J. D. Varela, and C. E. Schaerer. Analysis of contaminant transport under wind conditions on the surface of a shallow lake. *Mecánica Computacional*, vol. 34, n. 31, pp. 2155–2163, 2016.
- [2] D. Bareiro, E. O’Durnin, L. Oporto, and C. Schaerer. Wind effects on shallow water contaminant transport simulation. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, vol. 6, n. 1, 2018.
- [3] D. Bareiro, E. O’Durnin, L. Oporto, and C. Schaerer. Shallow water model for pollutant distribution in the ypacarai lake. *Revista de la Sociedad Científica del Paraguay*, vol. 26, n. 2, pp. 54–76, 2021.
- [4] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, 2 edition, 2003.
- [5] R. Cuevas, C. E. Schaerer, and A. Bhaya. A proportional-derivative control strategy for restarting the gmres (m) algorithm. *Journal of Computational and Applied Mathematics*, vol. 337, pp. 209–224, 2018.
- [6] G. Espínola, J. C. Cabral, and C. Schaerer. Adaptive GMRES(m) for the electromagnetic scattering problem. *Trends in Computational and Applied Mathematics*, vol. 21, n. 1, pp. 191, 2020.
- [7] L. Duvenbeck, C. Riethmüller, and C. Rohde. Data-driven geometric parameter optimization for PD-GMRES. *arXiv preprint arXiv:2503.09728*, vol. 1, 2025.
- [8] A. H. Baker, E. R. Jessup, and T. Manteuffel. A technique for accelerating the convergence of restarted gmres. *SIAM Journal on Matrix Analysis and Applications*, vol. 26, n. 4, pp. 962–984, 2005.
- [9] J. C. Cabral, C. E. Schaerer, and A. Bhaya. Improving GMRES(m) using an adaptive switching controller. *Numerical Linear Algebra with Applications*, vol. 27, n. 5, pp. e2305, 2020.
- [10] S. P. Kolodziej, M. Aznaveh, M. Bullock, J. David, T. A. Davis, M. Henderson, Y. Hu, and R. Sandstrom. The SuiteSparse matrix collection website interface. *Journal of Open Source Software*, vol. 4, n. 35, pp. 1244, 2019.
- [11] J. Varela, G. Espínola, and J. Cabral. KrySBAS: Krylov Subspace-Based Adaptive Solvers. <https://github.com/nidtec-una/krysbas-dev>. Versión v0.4.0; MATLAB toolbox; Licencia GPL-3.0, 2025.