XLIII CILAMCE
21–25, November, 2022
FOZ DO IGUAÇU – PR – BRAZIL

A Hybrid Algorithm based on Stationary and Krylov methods for Nonsymmetric Linear Systems

Sebastian Marin[1], Carlos M. Vera[1], Juan C. Cabralr[2], Christian E. Schaerer[2]

[1]*School of Engineering,National University of Asunción*
Campus Universitario, 2111 SL, San Lorenzo, Paraguay
emarin@fiuna.edu.py, cmvera@fiuna.edu.py
[2]*Polytechnic School, National University of Asunción*
Campus Universitario, 2111 SL, San Lorenzo, Paraguay
jccabral@pol.una.py, cschaer@pol.una.py

Abstract. Iterative Krylov methods, like Generalized Minimal Residual (GMRES) and Full Orthogonalization Method (FOM), are normally used for the solution of sparse and nonsymmetric linear systems from Computational Mechanics problems. In practice, restarted versions, are used to reduce storage and orthogonalization costs. However, numerical experience shows that these methods may present stagnation or slow convergence. The Stationary method is older, simpler to understand and implement, but usually not completely effective. Contrarily, the Krylov method has a more recent development and is more effective than the former, but the analysis is usually harder to understand with difficulties in selecting its parameters. A cycle of a proposed hybrid method consists of n Stationary iterations of Richardson followed by $m \times k$ iterations of the restarted GMRES, where n, m and k are values much smaller than the dimension of the non-symmetric matrix. Such cycles can be repeated until convergence is achieved. The advantage of this approach is in the opportunity to allow better performance of its individual properties. This combination of methods is competitive from the point of view of helping to accelerate convergence with respect to the number of iterations for some linear problems. We are going to present computational experiments to show the advantages and the main problems raised from the perspective of the proposed hybrid method.

Keywords: Iterative method, Nonsymmetric Linear Systems, Stationary methods, Krylov methods.

1 Introduction

In this work, we focus on iterative methods to solve linear systems of equations of the form

$$Ax = b \tag{1}$$

where $A \in \mathbb{C}^{n\times n}$ is non-singular and $b, x \in \mathbb{C}^n$. Iterative methods refer to a wide range of techniques that use a successive approximation to obtain more accurate solutions to a linear system at each iteration. Some ways to improve them are discussed by introducing static and dynamic parameters.

A cycle of Richardson-GMRES(n, m, k) consists of n Richardson iterations followed by k cycles of GMRES(m). Such cycles can be repeated until convergence is achieved. The result of any one cycle serves as the initial guess for the next cycle. The proposed method may also be considered a preprocessing, in which GMRES(m) is preceded by n Richardson iterations. The advantage in this approach is in the opportunity to use moderate m, which results in time and memory saving. Because the number of inner products among the vectors of iteration is about $O(m^2)$, using a moderate m is particularly attractive on message-passing parallel architectures, where inner products require expensive global communication (Sidi, Shapira, 1998).

This paper is organized as follows. In §2, we introduce the formulation for Stationary methods to solve linear systems. In §3, the non-stationary methods it is presented, specifically focusing on GMRES. Numerical results are presented at §4 and the conclusions are presented at §5 showing that the hybrid strategy improves the convergence of Richardson and standard GMRES(m).

2 Iterative Methods for Linear Systems

2.1 Stationary Methods

A simple iteration to solve 1 can be described by a recurrence of the form

$$x_{k+1} = x_k + Kr_k, \tag{2}$$

where K is a real $n \times n$ matrix and r_k is the residual vector defined by $r_k = b - Ax_k$ in each iteration. When the vector r_k reaches the zero vector, the desired solution of the linear system $Ax = b$ is reached.

Matrix K is often referred to as the feedback gain matrix and r_k represents the residual vector between the input b and the output Ax_k (Schaerer, Kaszkurewicz, 2001). The linear problem is equivalent to a regulator problem of forcing the output to regulate the constant input b, by a suitable choice of the controller, in this case, the matrix K (Bhaya, Kaszkurewicz, 2006). In the procedure of building the matrix K, each method differs one from another.

Two types of iterative methods are very popular: Stationary and Non-stationary methods. The Stationary method is older, simpler to understand and implement, but usually not completely effective. The Non-stationary methods have a more recent development and more effective, but the analysis is usually harder to understand. To use a control perspective for these iterative methods, the equation (2) is modified. Therefore, the approximate solution takes the form

$$x_{k+1} = x_k + K(b - Ax_k) = (I - KA)x_k + Kb. \tag{3}$$

The choice of constant matrix K and the relation $(I - AK)$ characterize a stationary method. It can be observed that the particular selection of $K = A^{-1}$ produces that the iterative method converges in only one iteration but this is impractical for large matrices. Conversely, Non-stationary methods differ from stationary methods because the matrix K and other parameters change at each iteration. Generally, these parameters are computed by taking the inner product of residuals, or other vectors with information about previous iterations.

Observe that, the residual r_k is obtained from equation (2) as (Schaerer, Kaszkurewicz, 2001):

$$r_{k+1} = (I - AK)r_k. \tag{4}$$

An improvement of iterative methods can be made by updating dynamically the feedback matrix K_k, i.e., $K = K_k$. Some non-stationary methods use the control Lyapunov function approach to obtain the appropriate parameters for every iteration (Bhaya, Kaszkurewicz, 2006). A first example of a non-stationary method is $K_k = \alpha_k I$, where α_k is a scalar sequence and I is an identity matrix of an appropriate dimension. If α_k is computed in terms of the residual vector r_k, the resulting method is referred to as Adaptive Richardson (Greenbaum, 1997).

For this example, the equation (4) can be written as

$$r_{k+1} = r_k - \alpha_k A r_k. \tag{5}$$

In control jargon, the value α_k is the control parameter and r_k the state. A control Lyapunov function can be used to design an asymptotically stabilizing state feedback control for equation (5) for driving the state r_k to the origin (Bhaya, Kaszkurewicz, 2006). When the linear system has an arbitrary nonsingular matrix A, the candidate Lyapunov function can be defined as:

$$V(r_k) := \langle r_k, r_k \rangle, \tag{6}$$

from which the α_k expression is chosen as

$$\alpha_k = \frac{\langle r_k, Ar_k \rangle}{\langle Ar_k, Ar_k \rangle}, \tag{7}$$

to guarantee $\Delta V(r_k) < 0$, and in that way, proves that the residual norm of vector r_k decreases monotonically to finally get the zero vector. In particular, this method is called Orthomin(1) (Greenbaum, 1997).

When the matrix is Hermitian positive definite, the candidate function of Lyapunov is defined by $V(r_k) := \langle r_k, A^{-1} r_k \rangle$, from which

$$\alpha_k = \frac{\langle Ar_k, A^{-1} r_k \rangle}{\langle Ar_k, A^{-1} Ar_k \rangle} = \frac{\langle r_k, r_k \rangle}{\langle r_k, Ar_k \rangle},$$

making $\Delta V < 0$, which corresponds to Richardson's method for symmetric matrices. The method can be viewed as a steepest descent method in which the step size α_k is chosen following a control Lyapunov function (Greenbaum, 1997). From a control viewpoint, Richardson's method can be viewed as the application of proportional controller with a time-varying gain α_k (Bhaya, Kaszkurewicz, 2006).

3 Non-Stationary methods

3.1 Krylov Methods

The GMRES method builds an orthogonal basis for the Krylov subspace

$$\mathcal{K}_m(A, r_0) = \text{span}\{r_0, Ar_0, ..., A^{m-1}r_0\} \tag{8}$$

using the modified Gram-Schmidt process (*Arnoldi's method*). This method can be written in matrix form as (Saad, Schultz, 1986).

$$AQ_m = Q_m H_m + h_{m+1,m} q_{m+1} \xi_k^T = Q_{m+1} H_{m+1,m}, \tag{9}$$

where columns of matrix Q_k are an orthogonal basis for $\mathcal{K}_m(A, r_0)$ and $H_k$an upper Hessenberg matrix.

In the GMRES method, the approximate solution x_m is taken to be of the form $x_m = x_0 + Q_m y_m$ for some vector y_m, which is obtained by solving the minimization problem

$$\min_y \|r_0 - AQ_m y\|_2. \tag{10}$$

The (full) GMRES start with $k = 0$ up to either convergence or $dim(x)$. This makes the algorithm impractical because of increasing stage and computational work requirements when the number of iterations needed to solve the linear system is large. The restarted GMRES algorithm (denoted as GMRES(m)) restarts the GMRES every m steps.

The GMRES(m) used in this work is presented in the algorithm 1.

Algorithm 1 The j-th cycle of GMRES(m).

Require: Given A, b, m, k .
1: $r_i = b - Ax_i$
2: Compute m iterations of restarted GMRES to solve $Ae_i = r_i$;
3: $x_i = x_i + e_i$
4: $r_i = b - Ax_i$
5: **if** $\| r_i \|_2 < tolerance$ **then**
6: stop;
7: **else**
8: $e^j = e_i$
9: **end if**
10: $j = j + 1$

3.2 The proposed method: Richardson-GMRES(n,m,k)

Our new method is denoted as Richardson-GMRES(n, m, k), where n is the number of Richardson iterations, m the dimension of the Krylov subspace and k is the number of cycles of GMRES(m). Any cycle of the proposed method cover n iterations of Richardson method and $m \times k$ iterations of GMRES.

The pseudocode for the j-th cycle of the proposed method maned as Richardson-GMRES(n, m, k) is presented in the Algorithm 2.

4 NUMERICAL RESULTS

We show the potential of the proposed method by presenting experimental results from a variety of problems. We tested 4 problems. The first example is a systems arising from a finite difference discretizations of the Poisson equation on rectangular grids using a five-point difference approximation. The following examples come from the Matrix Market Collection.

For problems in which the right-hand side are not specified, the vector b is generated randomly using a uniform distribution with values between the minimum and maximum values A(i, j). Two kinds of problems are consideredwith

A zero initial guess is used for all problems. If a right-hand side es not provided, we generate a random right-hand side. We stop the algorithm when the relative residual norm is less than the convergence tolerance, i.e., when $\frac{\|r^j\|}{\|r^0\|} \leq 10^{-12}$ or when the maximum number of cycles is exceeded ($j \leq 1000$).

Algorithm 2 The j-th cycle of Richardson-GMRES(n, m, k).

Require: Given A, b, n, m, k, x^{j-1}, e^{j-1} .
$x_i = x^{j-1}$
$e_i = e^{j-1}$
Compute n iterations of Richardson to solve $Ae_i = r_i$;
for $i = 1 : n$ **do**
    $r_i = b - Ax_i$
    $\alpha_i = \frac{\langle r_i, r_i \rangle}{\langle r_i, Ar_i \rangle}$
    $e_i = e_i + \alpha_i r_i$
end for
$r_i = b - Ax_i$
Compute $m \times k$ iterations of restarted GMRES to solve $Ae_i = r_i$;
$x_i = x_i + e_i$
$r_i = b - Ax_i$
if $\| r_i \|_2 < tolerance$ **then**
    stop;
else
    $e^j = e_i$
end if
$j = j + 1$

Table 1. The matrices information.

Matrix	size	nnz	Condest	Aplication area
Poisson2D	10000	3.01E+3	49600	Mechanics
add20	2395	13151	1.20E+4	Circuit simulation
sherman1	1000	3750	2.26E+4	Computational fluid dynamics
sherman4	1104	3786	7.16E+3	Computational fluid dynamics

These matrices are presented in Table 1 . The columns labeled $size$ and nnz are for matrix dimension and number of non-zeros entries. $Condest$ refers to the condition number. All test are run on a desktop machine with Intel Core i7-6700T CPU @ 2.80GHz X 4 and 8 GB of main memory, by using MATLAB R2016b for Windows 10.

We chose the initial restart parameter $m = 30$ for GMRES(30) because it is a common choice and often the default in general linear solver packages such as PETSc (see Balay et al., 2001). In the subsection §4.1 we compare Richardson, GMRES(30) and Richardson-GMRES(n, m, k) using the best parameter values found by varying n between 100 and 500 in steps of 100; m between 1 and 5 in steps of 1 and k between 1 and 10 in steps of 1. The nonconverging problems are denoted by NC in Tables 2.For the GMRES(m) executions, the MATLAB R2016b functions were used.

In Section §4.2, the proposed method is used with an incomplete LU preconditioner to compare the efficiency with the same matrices and comparative methods.

4.1 Comparison to standard methods

Experiment 1. Simple example for a common Partial Differential Equations encountered in various areas of engineering:

$$\Delta u = f \ \ in \ \Omega \tag{11}$$

$$u = 0 \ \ on \ \Gamma \tag{12}$$

where Ω is now the rectangle $(0, l_1) \times (0, l_2)$ and Γ its boundary. Both intervals are discretized uniformly.

The proposed method with parameter values $n = 400$, $m = 5$, and $k = 3$ has the lowest number of cycles with respect to the standards GMRES(30) and Richardson methods. The latter fails to converge (see Figure 1). The advantage of having small values of m allows for very low computational costs (Saad, 2003).

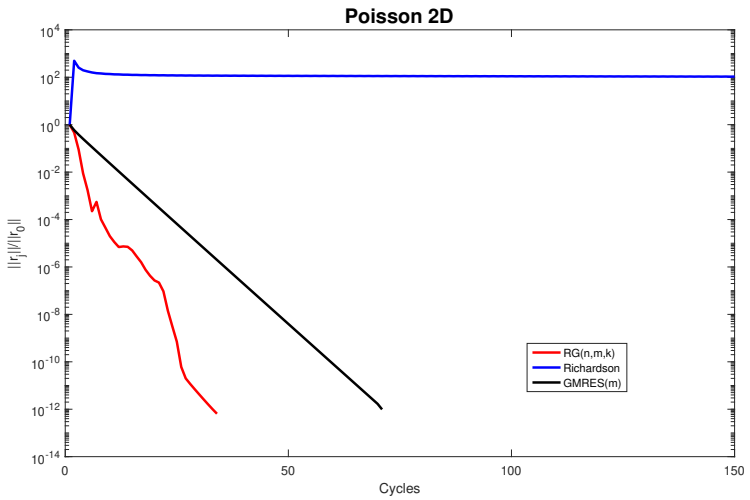

Figure 1. The convergence curves of matrix Poisson2D.

Experiment 2. We considered a group of benchmark problems from Applied in Computational Mechanics. All matrices in this group are non- symmetric with high conditioning numbers (see Table 1)

We consider the matrices sherman 1 and sherman 4. Both matrices are real non-symmetric from computational fluid problems. The proposed method has the lowest number of cycles with respect to the GMRES(30) and Richardson standard. The latter fails to converge. In both cases, the best values of n and k are the same. See Figures 2 and 3 respectively.

In the case of add20, the proposed method have the smallest number of cycles to achieve convergence (see Figure 4). The best values are $n = 500$, $m = 1$ and $k = 9$.

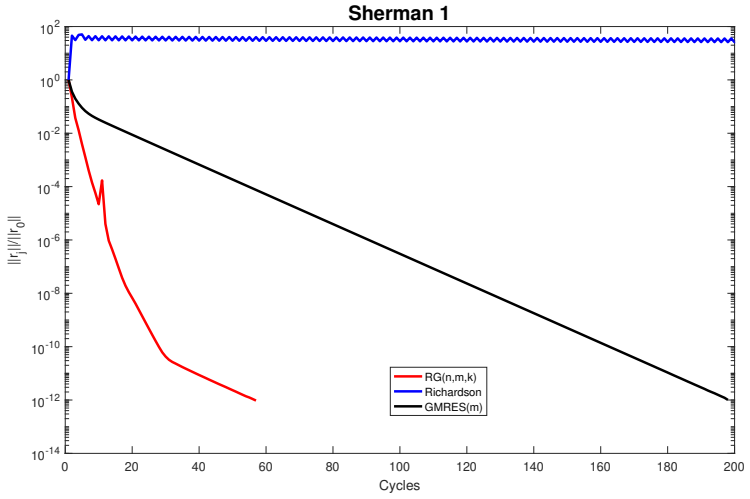

Figure 2. The convergence curves of matrix Sherman1.

4.2 Using ILU preconditioning

A common strategy in the iterative solution of linear systems is the use of preconditioners. In this work we use the incomplete LU factorization denoted as ILU to compare the efficiency of the proposed method, with respect to preconditioned systems. Specifically, we used the ILUTP version of MatLab 9.1.0 (R2016b) with the value $droptol = 10^{-6}$. For fairness in comparing the performance of the preconditioners for improving the rate of convergence of the methods, the tables do not include the running time for computing the ILU pprocess.

In Table 3, it is shown experimentally that all methods with a preconditioner converge with very fast conver-

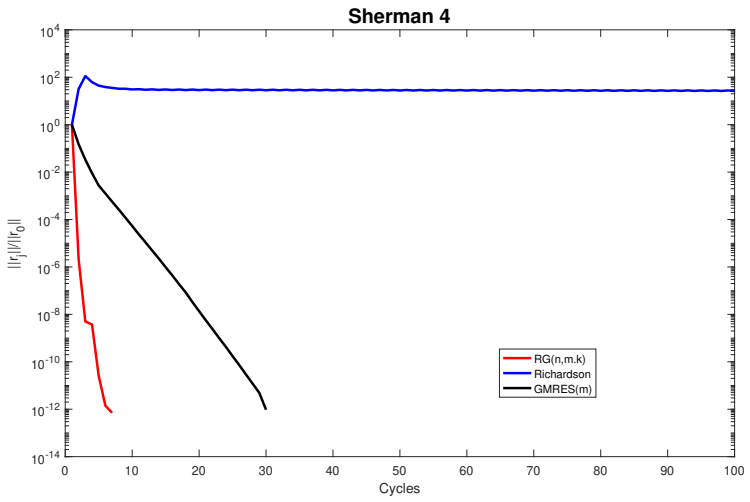

Figure 3. The convergence curves of matrix Sherman4.

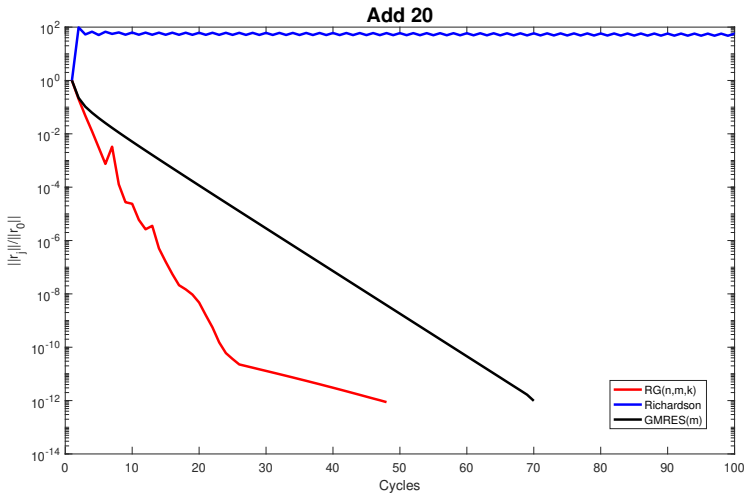

Figure 4. The convergence curves of matrix Add20.

Table 2. Numerical results for matrices without ILU preconditioning.

Matrix	Richarson		GMRES(30)		RG(n,m,k)	
	cycle	time	cycle	time	cycle	time
Poisson2D	NC	-	71	2.6778	34	2.913
add20	NC	-	70	0.370	48	0.878
sherman1	NC	-	198	0.590	57	0.373
sherman 4	NC	-	30	0.103	7	0.091

gence with respect to the numbers of cycles without a preconditioner, but in some cases, a considerable increase in execution time is given. In general, the selection of an appropriate preconditioner requires some a priori knowledge of the problem. For simplicity we choose the ILU version of Matlab, even though sometimes requires large memory requirements which exceed the computer capacity or produce Matrix close to singular or badly scaled. This is why we get the Wanring message during method execution.

Table 3. Numerical results for matrices with ILU preconditioning. The WM means that during execution a near-singular or poorly scaled matrix alert was given.

Matrix	Richarson		GMRES(m)		RG(n,m,k)	
	cycle	time	cycle	time	cycle	time
Poisson2D	8	1.611	WM	-	2	113.593
add20	WM	-	WM	-	2	5.708
sherman 1	6	0.008	WM	-	2	0.386
sherman 4	6	0.005	WM	-	2	0.241

5 CONCLUSION

A Hybrid Algorithm based on stationary and non-stationary for solving linear systems has been presented and tested. We find that the selected stationary method (Richardson) cannot reach convergence by itself, but by combining it with a non-stationary method (GMRES(m)), it does achieve convergence and even with fewer cycles. It is important to remark that the convergence is improved for all tested cases. Moreover, in order to achieve an improvement in terms of the number of cycles of convergence, it is important to choose adequately the parameters. Also, experimental results show that the ILU preconditioning improves the method proposed but has difficulties for the Richardson and GMRES(m), because the ILU sometimes yields an ill-conditioned system. For the problems presented, the superiority of the proposed method, denoted as RG(n, m, k) over Richardson and GMRES(30) is clear, but it has difficulty selecting the best set of parameters a priori. However, we show that with very small values of m, we achieve very good results. Making our proposal quite interesting from a computational point of view.

REFERENCES

Balay S., Buschelman K., Gropp W. D., Kaushik D., Knepley M.G., McInnes L.C., Smith B.F., & Zhang H. PETSc Web page, 2001. https://www.mcs.anl.gov/petsc/.

Bhaya, A., & Kaszkurewicz, E. (2006). Control perspectives on numerical algorithms and matrix problems. Society for Industrial and Applied Mathematics.
Greenbaum, A. (1997). Iterative methods for solving linear systems. Society for Industrial and Applied Mathematics.

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

Saad, Y., 2003. Iterative methods for sparse linear systems. SIAM. Philadelphia. USA.

Sidi, A., & Shapira, Y. (1998). Upper bounds for convergence rates of acceleration methods with initial iterations. Numerical Algorithms, 18(2), 113-132.

Schaerer, C. E., & Kaszkurewicz, E. (2001). The shooting method for the solution of ordinary differential equations: A control-theoretical perspective. International Journal of Systems Science, 32(8), 1047-1053.