Journal of Computational and Applied Mathematics 337 (2018) 209–224

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A proportional-derivative control strategy for restarting the GMRES(m) algorithm

Rolando Cuevas Núñez [a], Christian E. Schaerer [a,*], Amit Bhaya [b]

[a] Polytechnic School, National University of Asuncion, P.O. Box: 2111 SL, Central, Paraguay
[b] Department of Electrical Engineering, COPPE, Federal University of Rio de Janeiro, Rio de Janeiro 21945-970, Brazil

ARTICLE INFO

Article history:
Received 10 March 2016
Received in revised form 8 December 2017

Keywords:
GMRES(m)
Adaptive restarting parameter
Acceleration
Control Lyapunov law

ABSTRACT

Restarted GMRES (or GMRES(m)) is normally used for solving large linear systems $Ax = b$ with a general, possibly nonsymmetric, matrix A. Although, the restarted GMRES consumes less computational time than its counterpart full GMRES, if the restarting parameter is not correctly chosen its convergence cannot be guaranteed and the method may converge slowly. Unfortunately, it is difficult to know how to choose this parameter a priori. In this article, we regard the GMRES(m) method as a control problem, in which the parameter m is the controlled variable and propose a new control-inspired strategy for choosing the parameter m adaptively at each iteration. The advantage of this control strategy method is that only a few additional vectors need to be stored and the controller has the capacity to modify the dimension of the Krylov subspace whenever any convergence problem is detected. Numerical experiments, based on benchmark problems, show that the proposed control strategy accelerates the convergence of GMRES(m).

1. Introduction

The Generalized Minimal Residual (GMRES) [1,2] method is a popular algorithm for the solution of large, sparse and non-symmetric linear systems of equations

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

where A is an $n \times n$ matrix. To briefly describe GMRES, let x_0 be an initial approximation of the solution. Let $r_0 = b - Ax_0$ be the initial residual and $\mathcal{K}_k(A, r_0) = \text{span}\,\{r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0\}$ be the Krylov subspace defined by A and r_0. At each step k, GMRES constructs an $n \times k$ matrix W_k whose columns form an orthogonal basis for $\mathcal{K}_k(A, r_0)$. The solution at the kth GMRES step is

$$x_k = x_0 + W_k y_k \tag{2}$$

for some $y_k \in \mathbb{R}^k$ which minimizes the l_2 norm of the residual $r_k = b - Ax_k$ (denoted as $\|\cdot\|$ dropping the subscript 2), i.e.:

$$\|r_k\| = \min_{x \in x_0 + \mathcal{K}_k(A, r_0)} \|b - Ax\|. \tag{3}$$

* Corresponding author.
E-mail addresses: rcuevas@pol.una.py (R. Cuevas Núñez), cschaer@pol.una.py (C.E. Schaerer), amit@nacad.ufrj.br (A. Bhaya).

https://doi.org/10.1016/j.cam.2018.01.009
0377-0427/

Using the Arnoldi relationship $AW_k = W_{k+1}\hat{H}_k$ where $\hat{H}_k$ is a Hessenberg matrix; the right hand side of the expression (3) takes the form:

$$\|r_0 - AW_k y_k\| = \|r_0 - W_{k+1}\hat{H}_k y_k\| = \|\beta e_1 - \hat{H}_k y_k\|, \tag{4}$$

where W_{k+1} is an orthogonal matrix and $\beta = \|r_0\|$ [3]. GMRES has input r_0 and returns $z_k = W_k y_k$. Hence, the approximate solution to Eq. (1) is obtained by the expression (2). GMRES selects $x_k \in x_0 + \mathcal{K}_k(A, r_0)$ such that $\|b - A(x_0 + z_k)\|$ is a minimum over all $z \in \mathcal{K}_k(A, r_0)$ and its minimum residual requirement is equivalent to the condition $r_k \perp A\mathcal{K}_k(A, r_0)$ [4,3,1].

The classical GMRES method (based on the full Arnoldi recurrence) for non-symmetric matrices is in general very expensive. GMRES converges to the exact solution of (1) when $k = n$ orthogonalizations are used. However, as k grows, the Arnoldi matrix W_k becomes too large to be stored and computed. A standard alternative is to restart GMRES when a pre-specified maximum subspace dimension m is reached (i.e. when $k = m$) and the current approximate solution becomes the new initial guess for the next iteration. The restarted GMRES method is denoted as GMRES(m) [1,3,5].

At the jth iteration, GMRES(m) uses the residual r_j as the initial residual, and performs GMRES using only m vectors of $\mathcal{K}_m(A, r_j)$, which returns an updated vector z_j. We denote the application of GMRES to the vector r_j using a Krylov subspace of dimension m by: $z_j =$GMRES(A, r_j, m). Therefore, the $(j + 1)$th iterate x_{j+1} is obtained as follows:

$$x_{j+1} = x_j + z_j. \tag{5}$$

GMRES(m) uses x_{j+1} to compute a new initial residual r_{j+1} and proceeds to the next iteration (i.e. the next call to the GMRES) until a stopping criterion is met. The overall procedure can be expressed in terms of recurrences as:

$$\begin{cases} z_j &= \text{GMRES}(A, r_j, m), \\ x_{j+1} &= x_j + z_j, \\ r_{j+1} &= b - Ax_{j+1}. \end{cases} \tag{6}$$

The advantage of a restarted procedure is that at most m steps of the Arnoldi method are carried out per iteration of GMRES(m), so that both computational costs and memory allocations per iteration are kept under control. In fact, normally (at each iteration) the restart parameter m is set to a constant value. However, restart slows the convergence of GMRES since, when restarting, the orthogonality of the generated vectors with respect to the previously generated subspaces is not preserved [6,5,7,1]. Moreover, if the appropriate m is not chosen, the convergence of GMRES(m) is not guaranteed [8], and the method may either stagnate or converge slowly [9]. Unfortunately, it is difficult to know how to choose m a priori. In fact, if stagnation occurs, a simple strategy could be to enlarge the maximum subspace dimension m. This however does not always ensure faster convergence and a large m can be prohibitive and unnecessary [10,8,5,11].

The problem of improving the convergence of the GMRES(m) can be formulated as a control problem, where the parameters that are being modified can be regarded as the control variables. Control theory deals specifically with the choice of parameters or controls, based on some type of feedback and is therefore also suitable for analyzing numerical dynamical systems.

The use of control theory in numerical analysis for improving numerical algorithms or developing new ones is quite recent. For instance, Feedback control [12], Control Lyapunov Function [13–16], Lyapunov Optimizing Control [13,17], Internal Model Principle [18,19], Linear Matrix Inequality [19,20], Linear Quadratic Regulator related [21–23] and Hierarchical Model Principle [24,25] have all been used in this context. Specifically, the authors of [22,23] proposed a new minimal residual method, called LQRES because it is based on a Linear Quadratic Regulator (LQR).

LQR theory which is a particular branch of control theory guarantees the convergence of the method, but the LQRES method needs to find the computationally expensive solution of an algebraic Riccati equation (ARE). Later, in [23] the LQRESD was introduced which uses an approximate solution to ARE. Although, numerical experiments give evidence that this technique is useful for improving convergence, the computation of the solution of an ARE is expensive and even for an approximate ARE, the computational cost of the LQRESD algorithm can become prohibitive when the dimension of the problem grows.

GMRES(m) algorithm has two parameters in the formulation (6): the residual r_j and the number of orthogonalizations m. Thus, in order to improve its convergence we could try to modify one or both of them. Several techniques are found in the literature, for instance a strategy introduced in [26] changes only r_{j-1} using a hybrid residual and overcomes stagnation in some cases [27]. Another strategy changes only the parameter m [28,7,29–32]. Experimentally, it can be observed that m has much more influence on the rate of convergence and is strongly related to the computational cost of the method. Thus, in this paper, we concentrate our efforts on designing a control law to adjust m adaptively. Several papers have proposed different adaptive choices of the parameter m. Roughly speaking, these adaptive strategies or rules can be categorized into three groups: (a) simple rules, which contain empirical parameters that are hard to guess/estimate [29,30]; (b) rules involving non-trivial calculations of eigenvalues or zeros of polynomials [31,32,11,33,34] and (c) simple empirical rules [28,7].

We are interested in designing a rule to update the Krylov subspace dimension m in such a way as to enhance the robustness and accelerate the convergence of GMRES(m). This is done by considering the GMRES(m) to be a control problem in which the parameter m is the control variable. We have been inspired by the results presented in [28] and the observation that the GMRES(m) residual can alternate directions ($r_{j+1} \approx \eta r_{j-1}$ with $\eta \leq 1$) [6,26]. Observe that if this happens the convergence of the GMRES(m) deteriorates. Therefore, heuristically, a modification of the control variable m should attempt to avoid the alternating direction residuals.

In this article we propose a new control-inspired strategy for choosing the parameter m adaptively at each iteration j. The advantage of this method, denoted as PD-GMRES(m), is that only a few additional vectors need to be stored and the controller has the capacity to modify (increase and decrease) the dimension of the Krylov subspace if any convergence problem is detected. As usual, for ill-conditioned problems, mere modification of the parameter m may not suffice and, in these cases, a preconditioner might also be needed.

This paper is organized as follows. In Section 2 we introduce the control feedback formulation for GMRES(m). In Section 3, the feedback control law for GMRES(m) is presented, which yields PD-GMRES(m). This is one of the contributions of the article. In addition, a sufficient condition for its convergence, as well as a criterion for evaluating the rate of convergence, are presented. This approach constitutes another contribution of this article, since the same theoretical framework can be used for designing more sophisticated Krylov subspace-based iterative methods. In Section 4, numerical experiments using the proposed feedback control law are presented together with some considerations about stagnation. The results show that the proposed control law improves the convergence of the GMRES(m), and accelerates its convergence at the price of storing just the norm of three (residual) vectors.

2. The control formulation

A classical recurrence in control theory called a discrete time state space model is given by [35,36]:

$$\mathcal{P}\begin{cases} x_{j+1} &= Hx_j + Bz_j \\ y_j &= Cx_j + Du_j, \qquad j = 1, 2, 3, \dots \end{cases} \tag{7}$$

where x_j (with dimension n) is called the state vector, and the matrices H, B, C, D have dimensions $n \times n$, $n \times p$, $s \times n$ and $s \times p$, respectively; z_j is the input and y_j is the output (with appropriate dimensions). The deviation of the output with respect to a reference b defines the error $e_j = b - y_j$. Using this error, the regulation problem in control theory is that of choosing an appropriate sequence for the control variable z_j in order to minimize e_j or equivalently $\lim_{j\to\infty} y_j = b$. Eqs. (7) define the object which it is desired to control, known as the plant in control theory and denoted by $\mathcal{P}$.

A general feedback controller, denoted by $\mathcal{C}$, relating the control variable z_j and the error e_j variable has the form:

$$z_j = \phi(y_j, e_j), \tag{8}$$

and the problem to be solved is to find a feedback law $\phi(y, e)$ that drives the error e to zero as fast as possible and is easy to implement.

In the context of the iterative solution of a linear system $Ax = b$ from a control feedback point of view, we consider the residual $r_j = b - Ax_j$ as the error or deviation at jth iteration and zeroing the residual r_j corresponds to solving the linear system $Ax = b$. A general iterative method can be expressed by a recurrence of the form [4,24]:

$$\begin{cases} x_{j+1} &= x_j + Bz_j \\ y_j &= Ax_j, \end{cases} \tag{9}$$

where in terms of the model (7), the matrix $H = I$, $C = A$, $D = 0$ and $B = I$. The latter (matrix B) relates the dimension of vectors z and x and it can also be viewed as a preconditioner (for details see [13]).

For an appropriate sequence of state vectors x_j, the residual r_j is driven to zero and the output y_j is driven to b. This is formulated in control terms as follows. Find a control sequence z_j that will drive the error to zero and, consequently the state variable x_j to the desired solution of $Ax = b$, denoted by x^*. Observe that the feedback error $e_j = b - y_j$ corresponds to $r_j = b - Ax_j$, thus (8) can also be written as $z_j = \phi(y_j, r_j)$.

The control model for the proposed GMRES(m) algorithm is given as follows:

$$\begin{cases} r_j &= b - Ax_j, \\ z_j &= \mathrm{GMRES}(A, r_j, m), \\ x_{j+1} &= x_j + z_j \end{cases} \tag{10}$$

where the plant is $x_{j+1} = x_j + z_j$. Note that the behavior of the state vector is determined by the controller $z_j = \phi(y_j, r_j)$, in this case, the GMRES(m). Next we discuss how to design this controller GMRES(m).

Observe in (10) that z_j is a function of matrix A, the residual at jth iteration r_j and the maximum dimension allowed for the Krylov subspace m. In adaptive methods this dimension m can be updated at each iteration, hence in (10) the function for z_j takes the form:

$$z_j = \mathrm{GMRES}(A, r_j, m_j) \tag{11}$$

where m_j is the parameter specifying the maximum dimension of the Krylov space at the jth iteration of GMRES(m). Of course if $m \geq n$, only one iteration of the restarted GMRES(m) is performed, and the restarted GMRES(m) reduces to the standard GMRES discussed in Section 1 (and if we set $m = 1$ the Orthomin method is recovered). We mention this to emphasize the

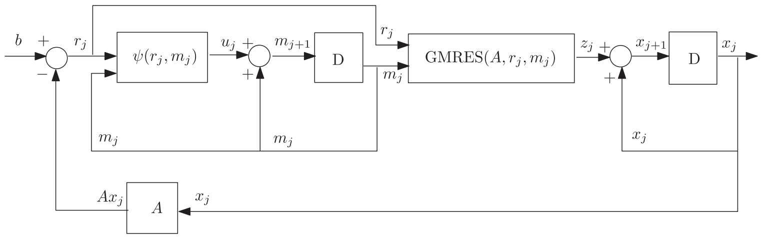


Fig. 1. Block diagram for adaptive GMRES(m). The block labeled D delays its input by one iteration, i.e. $x_j = \mathrm{D}(x_{j+1})$ and similarly for m_{j+1}.

iterative nature of GMRES(m) and that, with respect to the jth iteration, the initial residual r_j (as an input) remains constant. Therefore, to improve the value of z_j, the parameter m_j must be adapted appropriately. In order to update m_j, we propose to use a rule similar to (9) for updating x_j, i.e.:

$$m_{j+1} = m_j + u_j \tag{12}$$

where u_j is the *control variable* for m. A general feedback control law for u_j has the form:

$$u_j = \psi(r_q, m_j) \tag{13}$$

where the index q refers to the fact that the function ψ can use not just the current residual r_j but also the previous residuals $(r_{j-1}, r_{j-2}, \dots, r_0)$, which corresponds to a controller with memory as will be seen later. A schematic block diagram is presented in Fig. 1. The controlled GMRES(m) algorithm can be summarized as follows:

$$\begin{cases} r_j &= b - Ax_j, \\ z_j &= \mathrm{GMRES}(A, r_j, m_j), \\ u_j &= \psi(r_j, m_j), \\ m_{j+1} &= m_j + u_j, \\ x_{j+1} &= x_j + z_j. \end{cases} \tag{14}$$

Next, we introduce the proportional-derivative feedback control law, and within this framework, we also formulate some methods encountered in the literature.

3. Feedback control law

In this section we describe our proposal for $\psi(\cdot, \cdot)$, the feedback control law (13). To this end, following [6], we refer to the angle between consecutive residual vectors (corresponding to the jth and $(j-1)$th iterations, respectively) as the *sequential* angle, denoted by $\angle(r_j, r_{j-1})$. To obtain an expression for the sequential angle note that

$$\langle r_j, r_{j-1}\rangle = \langle r_j, r_j + AW_m y\rangle, \tag{15}$$

$$= \langle r_j, r_j\rangle + \langle r_j, AW_m y\rangle, \tag{16}$$

$$= \|r_j\|^2 + \langle r_j, AW_m y\rangle, \tag{17}$$

where W_m is a basis for the Krylov subspace $\mathcal{K}_m(A, r_{j-1})$. In addition, since $r_j \perp AW_m y$, we get $\langle r_j, r_{j-1}\rangle = \|r_j\|^2$. This motivates the following definition for the sequential angle $\angle(r_j, r_{j-1})$ (details are in [37,6]).

Definition 1. The sequential angle $\theta_j = \angle(r_j, r_{j-1})$ between the residual vector at the end of restart iterations j and $j-1$, r_j and r_{j-1}, respectively, is given by the following implicit relationship:

$$\cos(\theta_j) = \cos(\angle(r_j, r_{j-1})) = \frac{\langle r_j, r_{j-1}\rangle}{\|r_j\|\,\|r_{j-1}\|} = \frac{\|r_j\|}{\|r_{j-1}\|}. \tag{18}$$

The angle between *every other* residual vector is known as a *skip* angle. We are particularly interested in the residual skip angle between the jth and $(j-2)$th iterations, i.e. $\angle(r_j, r_{j-2})$ which is defined implicitly by $\cos(\varphi_j) = \cos(\angle(r_j, r_{j-2})) = \langle r_j, r_{j-2}\rangle/(\|r_j\|\,\|r_{j-2}\|)$. We consider that the angles are always between 0° and 90° [6].

Notice that given a Krylov subspace $\mathcal{K}_m$ then $x_j \in x_{j-1} + \mathcal{K}_m(A, r_{j-1})$ and by construction $r_{j-1} = r_j + AW_m y$ and $b - Ax_j \perp A\mathcal{K}_m(A, r_{j-1})$. Consequently, $\|r_j\| \leq \|r_{j-1}\|$ since $A\mathcal{K}_m(A, r_{j-1})y \neq 0$. The importance of sequential and skip angles derives from the observation that very often residual vectors point in nearly the same direction after every other restart iteration. This alternating behavior deteriorates the convergence of GMRES(m) [6,26]. Since by the construction of the solution in the Krylov subspace method $x_j \in x_{j-1} + \mathcal{K}_m(A, r_{j-1})$, then $x_j = x_{j-1} + p_{m-1}(A)r_{j-1}$ where p_{m-1} is a polynomial of degree $m-1$. The residual associated to the approximation x_j is

$$r_j = b - Ax_j = r_{j-1} - Ap_{m-1}(A)r_{j-1} = \hat{p}_m(A)r_{j-1} \tag{19}$$

with $\hat{p}_m(0) = 1$. Hence r_j is a combination of the basis vectors of $\mathcal{K}_m$ (matrix W_m). If r_j is in the direction of w_1 from the $(j-1)$th iteration and m is constant at iterations jth and $(j-1)$th, then the subspaces $\mathcal{K}_m(A, r_j) \approx \mathcal{K}_m(A, r_{j-1})$ and a significant increment is not obtained in the approximation of the solution x_j, that is $\|r_j\| \approx \|r_{j-1}\|$. Hence, to improve the rate of convergence the first part of our strategy consists in avoiding small sequential angles by varying the restart parameter. Repetitive behavior may happen even for residuals r_j and r_{j-p} with $p > 2$ (see [38] for details).

Next, we relate the quality of the convergence and the convergence itself by analyzing a Lyapunov function, the residuals and sequential angles. The following definition for Lyapunov functions [14] is used.

Definition 2. The function $V(r) : \mathbb{R}^n \to \mathbb{R}$ is said to be a candidate Lyapunov function on a subset $\mathcal{H}$ of $\mathbb{R}^n$, if (i) V is continuous on $\mathcal{H}$ and (ii) locally positive definite $V(0) = 0$ and $V(r) > 0$, $\forall r \in \mathcal{H} \setminus \{0\}$ with $\mathcal{H}$ being a neighborhood of $r = 0$; in addition $V(r)$ is a Lyapunov function if its increment $\Delta V(r)$ is negative definite (denoted < 0), whenever $r \in \mathcal{H}$.

The importance of Lyapunov functions in the context of this work arises from the fact that the system (10) is asymptotically stable (this implies that $\lim r_j \to 0$ as $j \to \infty$) whenever $\Delta V(r_j) < 0$ for the sequence generated by system (10). A natural question is how the control variable m_j should be chosen in order to force $\Delta V(r_j)$ to be negative definite.

We adopt the squared norm of the residual at the current iteration as a candidate Lyapunov function, i.e. $V(r_j) = \langle r_j, r_j \rangle$ at the jth iteration.

Definition 3. Let the residuals at iterations j, $(j-1)$ and $(j-2)$ be r_j, r_{j-1} and r_{j-2}, respectively. Let the candidate Lyapunov function be chosen as $V(r_j) = \langle r_j, r_j \rangle$. The local increment $\Delta V(r_j)$ and the skip local increment $\Delta\tilde{V}(r_j)$ at the jth iteration are defined as

$$\Delta V(r_j) := \|r_j\|^2 - \|r_{j-1}\|^2 \tag{20}$$

and

$$\Delta\tilde{V}(r_j) := \left(\Delta V(r_j) + \Delta V(r_{j-1})\right) = \left(\|r_j\|^2 - \|r_{j-2}\|^2\right); \tag{21}$$

and the relative skip increment rate $D(r_j)$ with respect to the $(j-1)$th iteration is defined as:

$$D(r_j) := \left(\|r_j\| - \|r_{j-2}\|\right) / \left(\|r_{j-1}\|\right). \tag{22}$$

Next we will formulate results in [6,38] in terms of the Lyapunov function (Definition 2) and the local increment (20).

Proposition 1. *Let $\cos(\theta_j)$ and $V(r_j)$ be as in Definitions 1 and 3, respectively. Assuming that $\|r_j\| \le \|r_{j-1}\|$ then $-1 \le \Delta V(r_j)/\|r_{j-1}\|^2 \le 0$.*

Proof. Dividing $\Delta V(r_j)$ by $\|r_{j-1}\|^2 > 0$ we get

$$\frac{\Delta V(r_j)}{\|r_{j-1}\|^2} = \frac{\|r_j\|^2}{\|r_{j-1}\|^2} - 1. \tag{23}$$

If $\|r_j\| < \|r_{j-1}\|$ then $\Delta V(r_j) < 0$ and conversely for (23) to be negative definite, we must have $0 \le \|r_j\|/\|r_{j-1}\| < 1$. This yields $-1 \le \Delta V(r_j)/\|r_{j-1}\|^2 \le 0$, since $0 < \|r_{j-1}\|$. □

Notice that if $\Delta V(r_j) < 0$ then the algorithm is locally convergent at iteration j. Moreover, the more negative $\Delta V(r_j)$ is, the better the convergence and more accentuated the reduction of the residual is (with respect to the previous one). If $\Delta V(r_j)/\|r_{j-1}\|^2 \approx -1$, then $\|r_j\|/\|r_{j-1}\| \approx 0$. This and Definition 1 imply $\cos(\theta_j) \approx 0$. Therefore, good convergence with respect to $\|r_{j-1}\|$ occurs when $\theta_j \approx 90^\circ$ (r_j almost orthogonal to r_{j-1}).

On other hand, if $\Delta V(r_j)/\|r_{j-1}\|^2 \approx 0$ then $\|r_j\| \approx \|r_{j-1}\|$ (which implies local stagnation with respect to $\|r_{j-1}\|$). This fact and Definition 1 imply $\cos(\theta_j) \approx 1$.

Next, we show a relationship between the skip local increment $\Delta\tilde{V}(r_j)$ and the relative skip increment rate $D(r_j)$.

Proposition 2. *Let $\Delta\tilde{V}(r_j)$ and $D(r_j)$ be as in Definition 3. If $\Delta\tilde{V}(r_j) < 0$ ($\Delta\tilde{V}(r_j) \approx 0$) then $D(r_j) < 0$ ($D(r_j) \approx 0$) with respect to $\|r_{j-2}\|$.*

Proof. Dividing $\Delta\tilde{V}(r_j)$ by $\|r_{j-1}\|(\|r_j\| + \|r_{j-1}\|) > 0$, considering that $\|r_j\|^2 - \|r_{j-2}\|^2 = (\|r_j\| + \|r_{j-2}\|)(\|r_j\| - \|r_{j-2}\|)$ and simplifying, it is obtained:

$$\frac{\Delta\tilde{V}(r_j)}{\|r_{j-1}\|(\|r_j\| + \|r_{j-2}\|)} = \frac{\|r_j\| - \|r_{j-2}\|}{\|r_{j-1}\|} = D(r_j). \tag{24}$$

If $\Delta\tilde{V}(r_j)$ is negative, then $D(r_j)$ is negative which means that $\|r_j\| < \|r_{j-2}\|$, while if $\Delta\tilde{V}(r_j) \approx 0$, then $D(r_j) \approx 0$ and consequently, $\|r_j\| \approx \|r_{j-2}\|$. □

From Propositions 1 and 2, it can be observed that good convergence of GMRES(m) implies $\Delta V(r_j)$ and $\Delta\tilde{V}(r_j)$ negative, $\|r_j\|/\|r_{j-1}\| \approx 0$ and $\|r_j\| < \|r_{j-2}\|$. Conversely when GMRES(m) has poor convergence, then $\Delta V(r_j)$ and $\Delta\tilde{V}(r_j)$ are close to zero, which implies that $\|r_j\|/\|r_{j-1}\| \approx 1$ and $(\|r_j\| - \|r_{j-2}\|)/\|r_{j-1}\| \approx 0$.

These interpretations motivate the controller to be introduced in the next section: it modifies the parameter m at each iteration with the intention of avoiding the construction of approximately the same Krylov subspaces $\mathcal{K}_m(A, \cdot)$ at consecutive iterations.

For completeness, a relationship between $D(r_j)$ and the skip angle $\cos(\varphi_j) = \cos(\angle(r_j, r_{j-2}))$ is presented in the next proposition using Lyapunov function arguments, for the case when GMRES(m) has poor convergence.

Proposition 3. *Assuming that GMRES(m) has poor convergence (* $\Delta V(r_j) \approx 0$ *and* $D(r_j) \approx 0$*) then* $\cos(\varphi_j) = \cos(\angle(r_j, r_{j-2})) \approx 1$.

Proof. If GMRES(m) presents poor convergence, then $D(r_j) \approx 0$ and $\|r_j\| \approx \|r_{j-2}\|$ or equivalently $\|r_j\|/\|r_{j-2}\| \approx 1$ (by Definition 3). In addition, since $\|r_{j-1}\|(\|r_j\| + \|r_{j-2}\|) > 0$, if

$$0 \approx D(r_j) = \frac{\Delta\tilde{V}(r_j)}{\|r_{j-1}\|(\|r_j\| + \|r_{j-2}\|)} = \frac{\Delta V(r_j) + \Delta V(r_{j-1})}{\|r_{j-1}\|(\|r_j\| + \|r_{j-2}\|)} \tag{25}$$

and $\Delta V(r_j) \approx 0$ then $\Delta V(r_{j-1}) \approx 0$.

Considering $\cos(\varphi_j) = \cos(\angle(r_j, r_{j-2})) = \langle r_j, r_{j-2}\rangle/(\|r_j\|\,\|r_{j-2}\|)$. Using the trigonometric identities $\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$ and $\cos^2(a) + \sin^2(a) = 1$,

$$\cos(\varphi_j) = \frac{\langle r_j, r_{j-2}\rangle}{\|r_j\|\|r_{j-2}\|} \tag{26}$$

$$= \cos(\theta_j)\cos(\theta_{j-1}) - \sin(\theta_j)\sin(\theta_{j-1}) \tag{27}$$

$$= \frac{\|r_j\|}{\|r_{j-2}\|} - \sqrt{1 - \frac{\|r_j\|^2}{\|r_{j-1}\|^2}}\sqrt{1 - \frac{\|r_{j-1}\|^2}{\|r_{j-2}\|^2}}. \tag{28}$$

Since $\Delta V(r_j) \approx 0$ and $\Delta V(r_{j-1}) \approx 0$, then $\sqrt{1 - \frac{\|r_j\|^2}{\|r_{j-1}\|^2}} \approx 0$ and $\sqrt{1 - \frac{\|r_{j-1}\|^2}{\|r_{j-2}\|^2}} \approx 0$ and

$$\cos(\varphi_j) = \frac{\langle r_j, r_{j-2}\rangle}{\|r_j\|\|r_{j-2}\|} \approx \frac{\|r_j\|}{\|r_{j-2}\|} \approx 1.$$

As a consequence of $\cos(\varphi_j) \approx 1$ and the angle $\angle(r_j, r_{j-2}) \approx 0$, the vectors r_j and r_{j-2} are approximately parallel. □

It is important to remark that complete stagnation ($\|r_j\| = \|r_{j-2}\|$ or equivalently $\|r_j\|/\|r_{j-2}\| = 1$) implies that r_j and r_{j-2} are exactly parallel to each other. However, it is more common to observe a degradation in the rate of convergence with r_j and r_{j-2} being approximately parallel.

3.1. *Proportional integrative derivative controller PID(m)*

Control engineers have long used an error based method known as a proportional–integral-derivative (PID) controller [39,35,40]. We consider a PID controller of the form:

$$u_j = \alpha_P\|r_j\| + \alpha_I\sum_{i=1}^{j}\|r_i\| + \alpha_D\left(\frac{\|r_j\| - \|r_{j-2}\|}{2}\right), \tag{29}$$

where α_P, α_I and α_D are real constants known as the proportional, integrative and derivative constants. Substituting (29) in the expression (12) yields:

$$m_{j+1} = m_j + \left(\alpha_P\|r_j\| + \alpha_I\sum_{i=1}^{j}\|r_i\| + \alpha_D\left(\frac{\|r_j\| - \|r_{j-2}\|}{2}\right)\right). \tag{30}$$

To work in dimensionless variables, and with the intention of defining the PID controller in terms of angles $\angle(r_j, r_{j-1})$ and $D(r_j)$; we divide the expression for the controller u_j by the norm of the residual $\|r_{j-1}\|$, then expression (30) takes the form:

$$m_{j+1} = m_j + \alpha_P\frac{\|r_j\|}{\|r_{j-1}\|} + \alpha_I\sum_{q=1}^{j}\frac{\|r_q\|}{\|r_{j-1}\|} + \alpha_D\left(\frac{\|r_j\| - \|r_{j-2}\|}{2\|r_{j-1}\|}\right). \tag{31}$$

In addition, since m is an integer, we take the integer parts of the proportional, integrative and derivative terms,

$$m_{j+1} = m_j + \left\lfloor \alpha_P \frac{\|r_j\|}{\|r_{j-1}\|} + \alpha_I \sum_{q=1}^{j} \frac{\|r_q\|}{\|r_{j-1}\|} + \alpha_D \frac{\|r_j\| - \|r_{j-2}\|}{2\|r_{j-1}\|} \right\rfloor. \tag{32}$$

In the context of this work we explore a simpler form of (32) known as proportional derivative controller, denoted here as PD(m). The PD(m) formulation is obtained by simply taking $\alpha_I = 0$ in (32):

$$m_{j+1} = m_j + \left\lfloor \alpha_P \frac{\|r_j\|}{\|r_{j-1}\|} + \alpha_D \frac{\|r_j\| - \|r_{j-2}\|}{2\|r_{j-1}\|} \right\rfloor. \tag{33}$$

We now explain the working of the control law (33). Note that when the algorithm is converging well, the term $(\|r_j\| - \|r_{j-2}\|)/\|r_{j-1}\|$ is large, causing m_j to change. In presence of slow convergence of the algorithm, the term $(\|r_j\| - \|r_{j-2}\|)/\|r_{j-1}\| \approx 0$ and its influence on m_j is minimal. On the other hand, when the GMRES(m) stagnates then the sequential angle $\angle(r_j, r_{j-1})$ is small and the proportional term becomes large, causing the parameter m_j to change.

It is important to mention that leaving m unchanged when GMRES(m) is converging well, does not necessarily imply good convergence in the next iteration, thus our strategy is to force modifications in m_j, and, whenever possible, we force a decrease in its value at each iteration. In conclusion, the proportional term modifies m_j when GMRES(m) presents stagnation (i.e. converges slowly) and the derivative part modifies m_j when GMRES(m) converges rapidly.

PD (m)-Algorithm. The proposed PD(m) controller begins with $m = m_{initial}$ and tends to decrease m. However, if $m \le m_{min}$, then the value of m for the next iteration of GMRES(m) is reset to $m = m_{initial} + m_{step}$ and $m_{initial} = m$. The pseudocode for the proposed rule is presented in Algorithm 1. The parameters used were: $m_{initial} = 30$, $m_{step} = 3$ and $m_{min} = 1$.

function PDRULE(m_j, $m_{initial}$, m_{min}, m_{step}, $\|r_j\|$, $\|r_{j-1}\|$, $\|r_{j-2}\|$)

if $(j > 3)$ **then**
$m_{j+1} = m_j + \left\lfloor \left(\alpha_P \frac{\|r_j\|}{\|r_{j-1}\|} + \alpha_D \frac{\|r_j\| - \|r_{j-2}\|}{2\|r_{j-1}\|}\right) \right\rfloor$
else if $(j > 2)$ **then**
$m_{j+1} = m_j + \left\lfloor \alpha_P \frac{\|r_j\|}{\|r_{j-1}\|} \right\rfloor$
else
$m_{j+1} = m_{initial}$
end if
if $(m_{j+1} < m_{min})$ **then**
$m_{initial} = m_{initial} + m_{step}$; $m_{j+1} = m_{initial}$
end if
end function

Algorithm 1: PD(m) strategy with threshold for m_{min}.

In this paper, the proposed controller does not impose a maximum value for m. This is done in order to evaluate the behavior of the control law. In practice, the controller could restrict the value of m between a minimum and a maximum value. Another important consideration is that for some ill-conditioned large scale problems, when the value of m grows unrestrictedly, the control law could lead to excessive memory requirement, while a method with restricted m could require more iterations to converge but be faster in terms of computational time. Unfortunately, this is not always the case, and therefore a good choice of a trade off in the control law between memory requirements and iterations to convergence is still an open problem.

3.2. Selection of control parameters α_P and α_D

Since matrix A remains constant and r_{j-1} is not changed at each iteration of GMRES(m), then for improving its rate of convergence, there is a need to change the restart parameter m for modifying the residual r_j at jth iteration.

Proposition 4. *Assuming that $\|r_j\|/\|r_{j-1}\| \approx 1$ (poor GMRES(m) convergence), A and r_{j-1} are unchanged, then the restart parameter m has to be changed to improve the rate of convergence of GMRES(m).*

Proof. Consider

$$r_{j-1} = r_j + Az_{j-1} \tag{34}$$

such that $r_j \perp Az_{j-1}$ by construction. Hence

$$\|r_{j-1}\|^2 = \|r_j\|^2 + \|Aw_{j-1}\|^2. \tag{35}$$

Using Definition 3, $V(r_j) = \|r_j\|^2$ and $V(r_{j-1}) = \|r_{j-1}\|^2$ and if GMRES(m) presents poor convergence $\|r_{j-1}\| \approx \|r_j\|$. Then

$$\Delta V(r_j) = \|r_j\|^2 - \|r_{j-1}\|^2 = -\epsilon^2 \approx -\|Aw_{j-1}\|^2 \tag{36}$$

with $\epsilon \ll 1$. In addition $\epsilon = \|Az_{j-1}\| = \|Ap_m(A)r_{j-1}\|$ and by the minimality of the polynomial $p_m(A)$, the only way to modify ϵ consists in modifying the restarting parameter m, since matrix A and residual r_{j-1} remain unchanged. □

For GMRES(m) and given a specific restart parameter m, the residual r_j can be written as

$$r_j = r_{j-1} + \sum_{k=1}^{m} \gamma_k A^k r_{j-1}. \tag{37}$$

Assuming that GMRES(m) has poor convergence, then $\epsilon = \|\sum_{k=1}^{m} \gamma_k A^k r_{j-1}\|$ where the coefficients γ_k are small with $|\gamma_k| < \delta$ for all k.

If a variation of m is made, say $\tilde{m} = m + \Delta m$ with positive Δm, a new residual at jth iteration is obtained as follows:

$$\tilde{r}_j = r_{j-1} + \sum_{k=1}^{\tilde{m}} \tilde{\gamma}_k A^k r_{j-1}. \tag{38}$$

Subtracting expressions (37) and (38) yields

$$\tilde{r}_j - r_j = \sum_{k=1}^{\tilde{m}} (\tilde{\gamma}_k - \gamma_k) A^k r_{j-1}, \tag{39}$$

$$= \sum_{k=1}^{m} (\tilde{\gamma}_k - \gamma_k) A^k r_{j-1} + \sum_{k=m+1}^{m+\Delta m} \tilde{\gamma}_k A^k r_{j-1}. \tag{40}$$

Since $|\gamma_k|$ is bounded and small, the variation of $\tilde{r}_j - r_j$ is given mainly by the coefficients $\tilde{\gamma}_k$, independent of the sign of Δm (positive or negative). A similar analysis can be made for a negative Δm.

Observing expression (40) and assuming that GMRES(m) has a poor convergence $((\|r_j\| - \|r_{j-2}\|)/\|r_{j-1}\| \approx 0$ and $\|r_j\|/\|r_{j-1}\| \approx 1)$ then either an increment or decrement of m can result. It is not possible to know *a priori* what decision will improve the rate of convergence (a larger decrement in the residual norm of r_j with respect to the residual norm of r_{j-1}). In this situation, we chose to decrease m by choosing α_P negative. It is important to remark that if the decision was not appropriate, then the controller will continue to decrease the parameter m until a predefined minimum value m_{min} and then it will introduce an increment in m.

On the contrary, if GMRES(m) has a good rate of convergence, by Propositions 1 and 2, the term $(\|r_j\| - \|r_{j-2}\|)/\|r_{j-1}\|$ is large while $\|r_j\|/\|r_{j-1}\|$ is small; i.e. the dominant term is the derivative term associated to α_D independently of the sign and size of α_P (the proportional term of (33)). As a consequence, in order to reduce the number of matrix–vector products in the algorithm, the parameter α_D is taken as a positive value.

Besides the extreme situations of convergence (good convergence or poor convergence), GMRES(m) can present intermediate convergence behavior. In such a case the controller decreases or increases the parameter m depending on the relative size of the proportional and derivative terms.

Let t_l be the relative execution time of the PD-GMREs(m) with respect to the GMRES(m) with $m = 30$. We assume that there exists a pair (α_P^l, α_D^l) which minimizes t_l and satisfies the criterion mentioned above. We denote the dependence of t_l on (α_P^l, α_D^l) by $t_l(\alpha_P^l, \alpha_D^l)$. The pair (α_P^l, α_D^l) varies from one problem to another. For a set of n_p problems, we propose to find a pair (α_P, α_D) which minimizes the performance functional $\sum_{l=1}^{n_p} t_l$, i.e.,

$$(\alpha_P, \alpha_D) = \arg \min_{(\alpha_P^l, \alpha_D^l) \in \mathcal{F}} \sum_{l=1}^{n_p} t_l, \tag{41}$$

where $t_l = t_{PD-GMRES(m)}/t_{GMRES(30)}$. We adopt $\mathcal{F} = [-9, -2] \times [2, 9]$ (a wider range could be used, if required). Since $\frac{\|r_j\|}{\|r_{j-1}\|} \le 1$ and the expression $\frac{\|r_j\| - \|r_{j-2}\|}{2\|r_{j-1}\|}$ is negative when GMRES(m) is converging, the choice of $\mathcal{F}$ ensures a decrease in m. Additionally, the range of values of $\mathcal{F}$ ensures that both α_P and α_D are comparable. Note that parameter identification can be stated as future work. Solving matrix equations and coupled matrix equations are closely related to parameter estimation [41–44], and when parameters cannot be found directly an iterative or a recursive algorithm for parameter identification should be used.

Table 1
List of problems tested : n is the matrix size, nnz the number of nonzero elements.

	Problem	n	nnz	Application area
1	add20	2395	17 319	Computer component design
2	cdde1	961	4 681	2D convection–diffusion operator
3	circuit_2	4510	21 199	Circuit simulation
4	fpga_trans_01	1220	7 382	Circuit simulation
5	orsirr_1	1030	6 858	Oil reservoir simulation
6	orsreg_1	2205	14 133	Oil reservoir simulation
7	pde2961	2961	14 585	Model PDE equation
8	raefsky1	3242	294 276	Incompressible fluid flow
9	raefsky2	3242	293 551	Incompressible fluid flow
10	rdb2048	2048	12 032	Reaction–diffusion Brusselator model
11	sherman4	1104	3 786	Oil reservoir simulation
12	steam2	600	13 760	Injected steam oil recovery
13	wang2	2903	19 093	Electron continuity equations
14	watt_1	1856	11 360	Petroleum engineering
15	young3c	841	3 988	Acoustic scattering

Table 2
Summation of relative execution times. $(\alpha_P, \alpha_D) = \arg\min_{\mathcal{F}} \sum_{prob=1}^{15} \frac{t(PD-GMRES)}{t(GMRES(30))}$.

		α_D							
		9	8	7	6	5	4	3	2
α_P	−2	9.9933	9.0770	8.9532	9.1054	9.1249	9.4848	9.2532	9.8774
	−3	**8.6881**	9.5174	9.4579	9.2113	9.0489	9.1039	9.3484	9.2673
	−4	8.9292	9.1588	9.3705	9.4879	9.4287	9.3195	9.1447	9.2452
	−5	9.5002	9.1723	9.2949	9.2319	9.5033	9.6774	9.5098	9.5893
	−6	9.3876	9.4667	9.5873	9.2314	9.2485	9.3048	9.6437	9.8889
	−7	9.6774	9.4461	9.5015	9.7570	9.8140	9.7507	9.6046	9.3199
	−8	9.5715	9.4566	9.9125	9.7014	9.9797	10.0522	9.8257	10.0500
	−9	10.2221	9.8085	10.1873	10.3653	10.5971	11.3001	11.2451	11.2556

4. Numerical experiments

In this section we provide experimental comparative results which demonstrate the efficiency of the PD-GMRES(m) algorithm presented in Section 3.1. We concentrate on the resolution of a linear system $Ax = b$ with matrices A obtained from the University of Florida's Matrix Repository [45]. The problems for the experiments are listed in Table 1. The problems are nonsymmetric and with high condition number. For problems in which the right hand side b was not specified, it was generated randomly using a uniform distribution with values between the minimum and maximum values $(A)_{i,j}$ of A. All the experiments were run on a PC with an Intel Centrino Core 2 Duo processor with 2 Mb of RAM, running Kubuntu 9.1 and Matlab for Linux. We solved the system $Ax = b$ using the standard GMRES(m) (obtained from NetLib) with $m = 30$, and Baker's rule (obtained from [28]), Gonçalez's rule (obtained from [7]) and the PD(m) controller rule for m. The stopping criterion at the jth iteration tolerance is $\|r_j\|/\|r_0\| \leq 10^{-9}$. The initial solution for GMRES(m) is $x_0 = 0$ or equivalently $r_0 = b$.

We solved each problem with PD-GMRES using 64 pairs (α_P, α_D) and measured the execution times. We also solved each problem using GMRES(30). Table 2 presents the execution time $\sum_{l=1}^{n_p} t_l$ with $n_p = 15$ corresponding to the list of problems presented in Table 1. Observe that, in accordance with the criteria presented in Section 3.2 and (41), the best pair is $\alpha_P = -3$ and $\alpha_D = 9$.

Comparative performance of the PD-GMRES(m). For the PD-rule (see Algorithm 1) the parameters for the experiments are $m_{initial} = 30$, $m_{step} = 3$, $m_{min} = 1$. For Baker's rule (see [28]) the parameters are $m_{min} = 1$, $m_{max} = 30$ and $m_{step} = 3$. For Gonçalez's rule (see [7]) we used $m_{ini} = 10$, $m_{max} = 30$ and $tolerance = 10^{-9}$.

Table 3 presents the computational time of the algorithms implemented and two pairs of (α_P, α_D), with $(\alpha_P, \alpha_D) = (-3, 5)$ and $(\alpha_P, \alpha_D) = (-3, 9)$. Observe that PD-GMRES computational times present small variations with the best running times in comparison with the other algorithms implemented. Comparing the running times in Tables 2 and 3, it can be seen that the best time depends on the choice of the parameters α_P and α_D. The criterion for choosing was given in Section 3.2. and expression (41).

The bar graph in Fig. 2 summarizes the experimental results: each algorithm was run five times and the running times were averaged, using the same vector b. Although most of the best computational times correspond to the choice $(\alpha_P, \alpha_D) = (-3, 9)$, we decided to be more conservative in the influence of the derivative part of the controller, therefore we adopted $(\alpha_P, \alpha_D) = (-3, 5)$ for the comparisons. To avoid scale differences between problems, the running times are displayed relative to the running time of GMRES(30). Thus the value 1 corresponds to GMRES(30), and values below 1 correspond to execution times better than GMRES(30), while values above 1 are worse than GMRES(30). Bars are grouped by problems: the problem number corresponds to the numbers listed in Table 1. Observe that for the prescribed tolerance of

Table 3
Computational times for each problem and the algorithms implemented. The table presents the average time over 5 runs. Aver. (a) corresponds to the parameter choice of $\alpha_P = -3$, $\alpha_D = 5$, while Aver. (b) corresponds to $\alpha_P = -3$, $\alpha_D = 9$.

Problem	GMRES(30) Time (s)	Baker GMRES(m) Time (s)	PD-GMRES(m) Time (s)		Gonçalez GMRES(m) Time (s)
	Aver.	Aver.	Aver. (a)	Aver. (b)	Aver.
1	17.7437	9.2157	9.0660	**8.3891**	11.5592
2	2.3194	1.6069	1.2431	**1.2020**	2.8601
3	36.8237	36.8000	25.8835	**22.7708**	42.8250
4	18.1224	7.4236	5.6771	**5.4094**	18.5770
5	13.4254	7.1122	5.8543	**5.7252**	13.7273
6	7.8283	6.4592	**5.6494**	5.6803	7.4054
7	7.1557	**6.2200**	6.8923	6.6989	6.2554
8	99.4634	42.5339	**26.3677**	29.1733	103.1257
9	136.1778	109.1609	69.4782	**62.8267**	135.9549
10	5.0067	5.0617	**4.5223**	4.6538	5.7373
11	2.2404	1.4729	**1.3356**	1.4016	2.2317
12	0.3702	0.3600	0.2450	**0.2411**	0.4044
13	40.9925	19.9110	**17.2837**	17.6929	30.4704
14	27.9854	7.8887	7.4021	**6.7007**	14.2184
15	20.4976	19.8914	18.1827	**18.1970**	20.1603

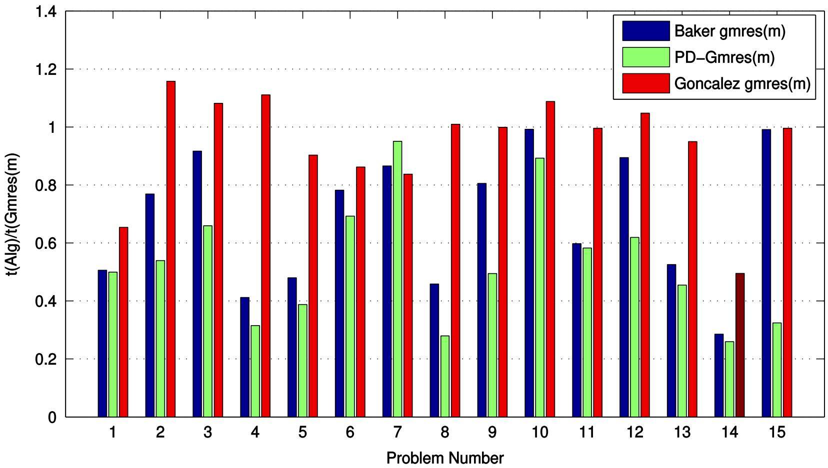


Fig. 2. Relative execution times for problem numbers 1 through 15. The shorter the bar, the smaller the convergence time. For problem 15, convergence in less than 1000 iterations occurred only for the proposed PD-GMRES(m) algorithm.

10^{-9} all algorithms converge for all problems, except for Problem 15. For this problem, the methods stop at 1000 iterations and only the PD-GMRES algorithm converges. In general, the PD-GMRES has better performance than the other algorithms tested except for the problems 7 and 10, where it has a behavior similar to GMRES(30). Even for these cases, the difference in performance is not more than 10%.

Analysis of Problems 2, 5 and 7. Here we discuss in detail three representative problems extracted from Table 1. We analyze the influence of the proportional and derivative terms in the convergence of PD-GMRES(m). Results show that the Lyapunov function introduced in Section 3 does indeed provide a sufficient condition for the convergence of the method.

Problem 2. Here we analyze a 2D convection diffusion problem denoted by "cdde1" (Problem 2) in Table 1. Fig. 3(a) and (b) show the behavior of residuals and values of m with respect to the number of iterations, respectively. Fig. 4(a) shows the behavior of the proportional and derivative parts of the controller, while Fig. 4(b) shows the behavior of the cosine of skip angles of GMRES(30) and PD-GMRES(m).

In terms of number of iterations PD-GMRES(m) converges faster than the other GMRES(m) rules implemented as can be seen from Fig. 3(a). Comparing Fig. 3(a) and (b), it is observed that the controller tends to decrease the value of m, but when the residual stagnates, the controller forces a variation of m. If this variation produces a value of m smaller than a given m_{min} then the controller forces an increase in m.

In Fig. 3(b), the parameter m at jth iteration is the starting parameter for GMRES(m) which gives a residual at $(j+1)$th iteration plotted in Fig. 3(a). It is observed that when m is subject to large changes, from a value m_j to another m_{j+1}, the local increment $\Delta V(r_j)$ is more pronounced. Due to the form of the Lyapunov function, this is equivalent to a large reduction in

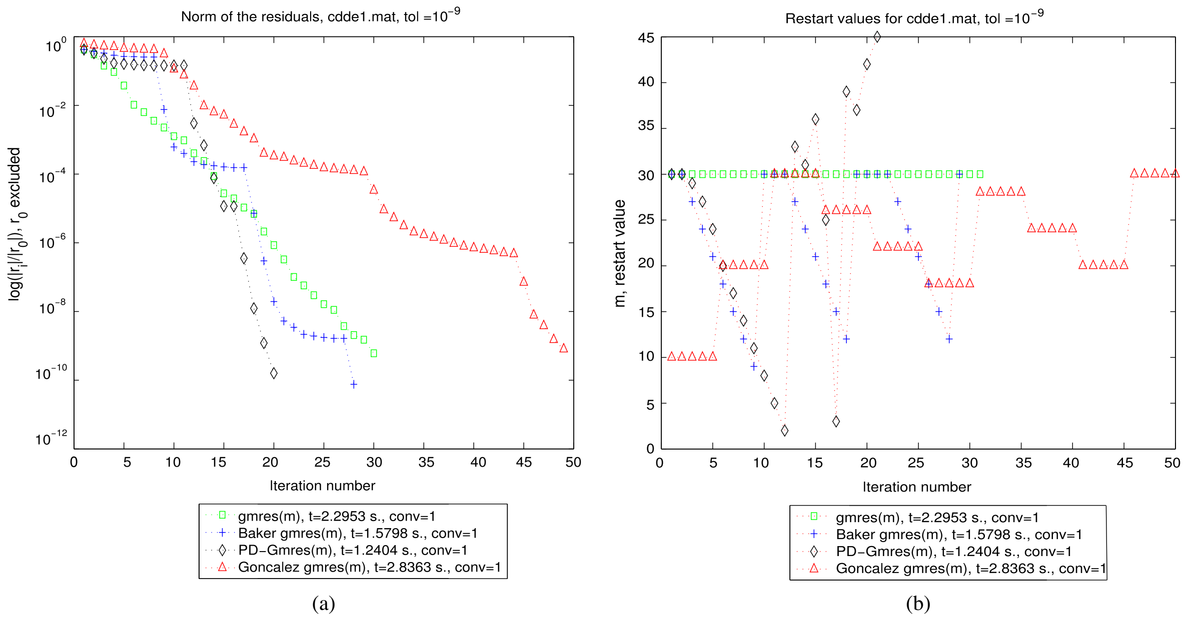


Fig. 3. Problem 2: (a) relative residual norms $\|r_j\|/\|r_0\|$ versus iteration number j; (b) restart value m_j versus iteration number j. Black diamond $\diamond$: PD-GMRES(m); Red triangle $\triangle$: Gonçalez rule; Green box $\square$: GMRES(30) and Blue $+$: Constant (Baker et al.) Rule.

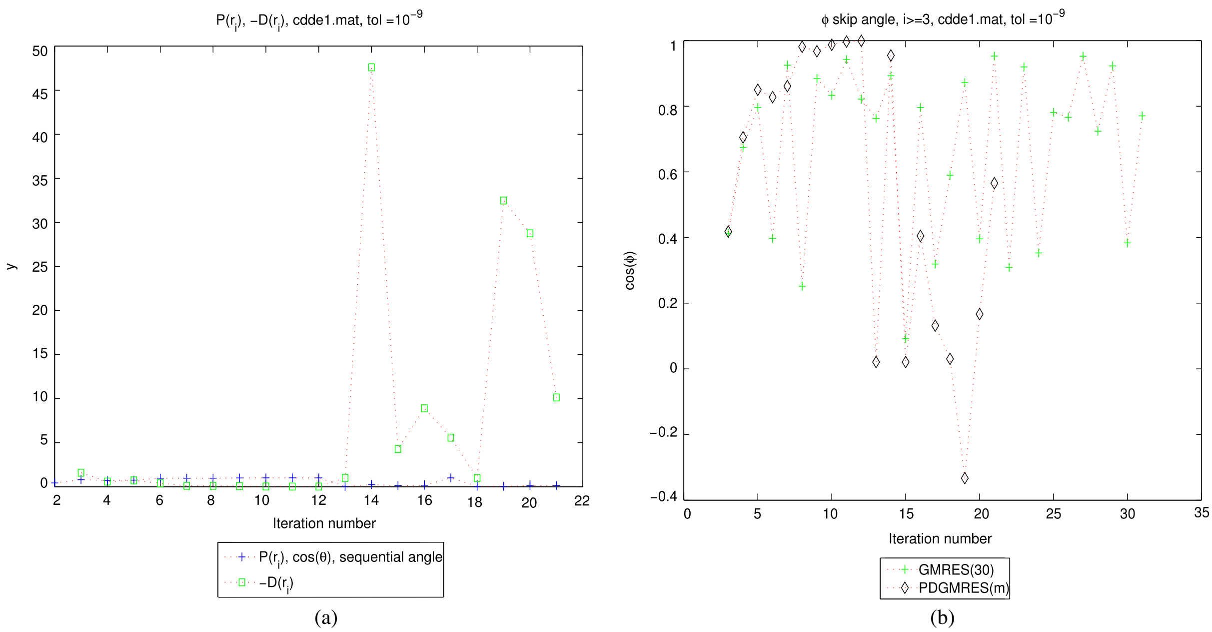


Fig. 4. Problem 2: (a) proportional (blue $+$) and derivative (green box $\square$) values without factors α_P and α_D calculated by PD-GMRES; (b) cosine skip angles, green $+$ for GMRES(30) and black diamond $\diamond$ for PD-GMRES(m). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the residual norm. Large reductions in the residual occur when m is changed, from a minimum value to a large one and vice versa. This result corroborates previous results presented in [28] and [26].

When the value of m is incremented to a larger one, there is a possibility of continuing with this value of m, but given the objective of reducing memory requirements, the value of m is decreased, so that a large m is used only when it is required. This property of increasing m when it is necessary is a controller prerogative as can be observed in Fig. 3(b) between iterations 12 and 20.

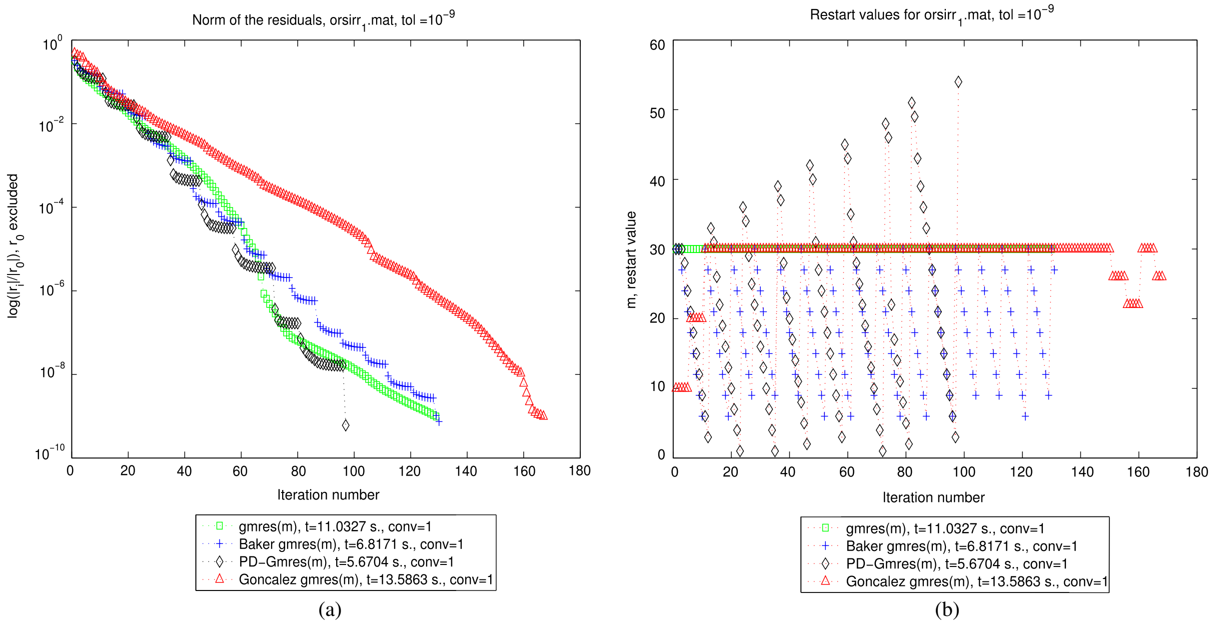


Fig. 5. Problem 5: (a) relative residual norms $\|r_j\|/\|r_0\|$ versus iteration number j; (b) restart value m_j versus iteration number j. Black diamond $\diamond$: PD-GMRES(m); Red triangle $\triangle$: Gonçales rule; Green box $\square$: GMRES(30) and Blue $+$: Constant (Baker et al.) Rule.

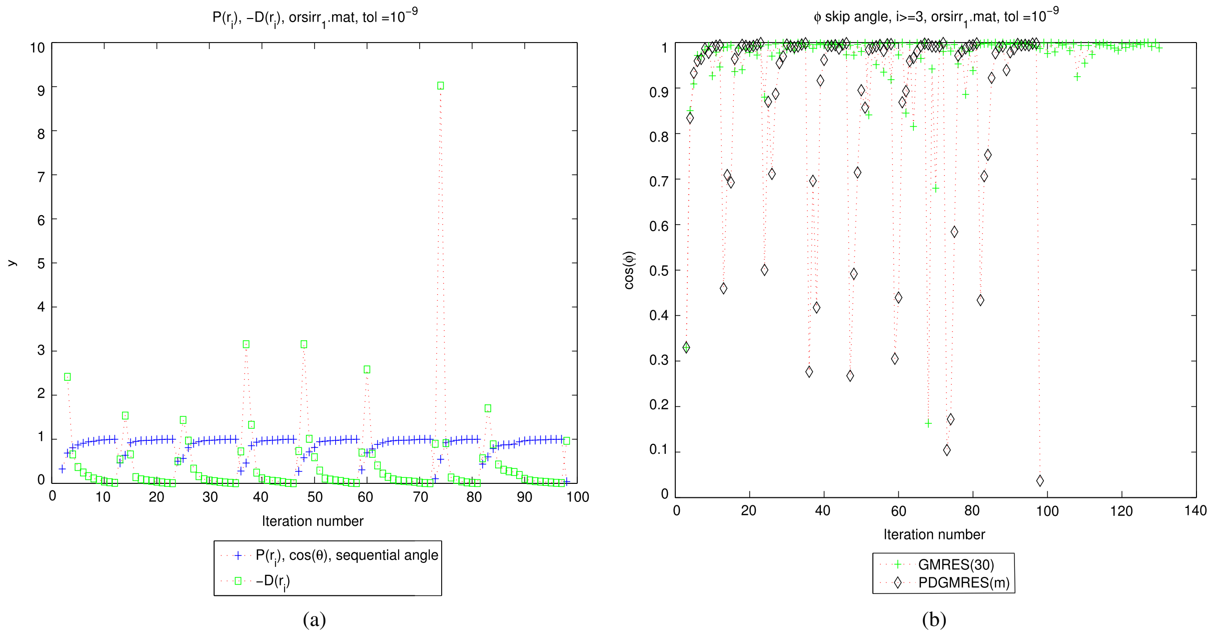
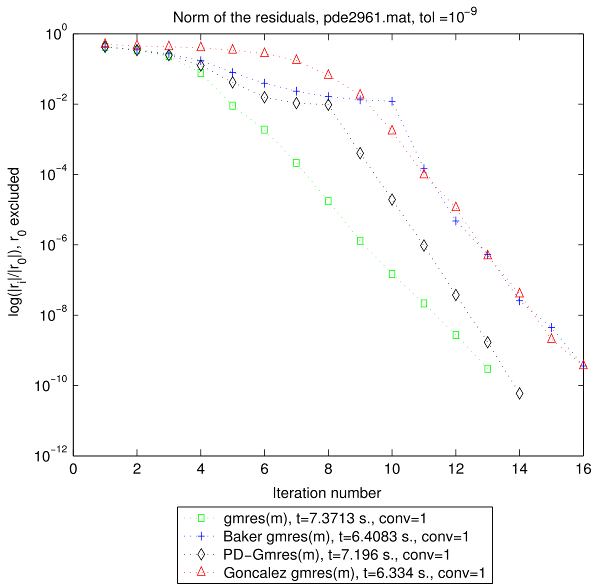


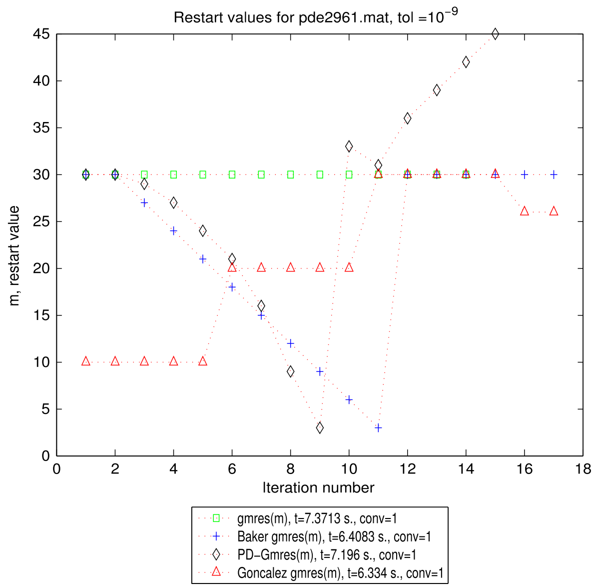
Fig. 6. Problem 5: (a) proportional (blue $+$) and derivative (green box $\square$) values without factors α_P and α_D calculated by PD-GMRES; (b) cosine skip angles, green $+$ for GMRES(30) and black diamond $\diamond$ for PD-GMRES(m). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Problem 5. Arises in oil reservoir simulation and is denoted "*orsirr*$_1$" (Problem 5) in Table 1. Fig. 5 presents the behavior of the residuals and values of m with respect to the number of iterations. In addition, Fig. 6(a) shows the behavior of the proportional and derivative parts of the controller. Fig. 4(b) shows the behavior of the skip angles of GMRES(30) and PD-GMRES(m).

A comparison of Fig. 5(a) and (b) leads to the same conclusion reached in Problem 2, namely that large reductions in the residual norm correspond to large changes in m. It can also be observed that GMRES(30) converges, however, it produces



(a)



(b)

Fig. 7. Problem 7: (a) relative residual norms $\|r_j\|/\|r_0\|$ versus iteration number j; (b) restart value m_j versus iteration number j. Black diamond $\diamond$: PD-GMRES(m); Red triangle $\triangle$: Gonçales rule; Green box $\square$: GMRES(30) and Blue $+$: Constant (Baker et al.) Rule.

small variety in the skip angles. The authors verified that for the tested cases when GMRES converges slowly skip angles are approximately zero or 180°. This is in accordance with the theoretical results presented in Section 3 and numerical results reported in [28,26]. It can also be observed that PD-GMRES produces a large variety of skip angles as compared with GMRES(30); in fact the skip angles exhibit more variations (see Fig. 6(b)) in PD-GMRES than in GMRES(30). In addition, since the parameter m is not fixed and varies from a maximum to a minimum value, it can save computational time since, overall, it could perform fewer orthogonalizations than GMRES(30). A comparison of the behavior of the residuals (Figs. 5(a) and 6(a)) relates the behavior of the sequential and skip angles, so that we can verify the qualitative interpretations of Propositions 1 and 2.

Problem 7. Here we analyze a model PDE equation denoted as "PDE2961" in Table 1. Fig. 7(a) and (b) show the behavior of residuals and values of m with respect to the number of iterations. In this case, although the PD-GMRES(m) requires more iterations to converge than GMRES(30), it converges faster in terms of execution time (see Fig. 2). This is due to the fact that the PD controller saves computational time during the first 9 iterations, since the controller decreases the restarting parameter m (see Fig. 7(b)), while the increment of m, after the iteration number 10, decided by the feedback controller, is quite modest.

Fig. 8(a) and (b) show the behavior of the proportional and derivative parts of the controller, and cosine skip angles of the GMRES(30) and PD-GMRES(m). Observe that, in this problem, when a deterioration of convergence occurs (roughly from iteration 2 to iteration 5), the derivative part is larger than the proportional part (8(a)), but the parameters α_P and α_D are not large enough to introduce a sufficiently large modification of m (compare Fig. 8(b)). This shows the relevance of choosing the proportional and derivative parameter adequately. It is also observed that, corroborating the theory, large derivative terms cause large variations in the parameter m and a larger decrement in the residual.

4.1. Considerations about slow convergence

As mentioned before, for Problem 15 in Table 1, only PD-GMRES converges for the prescribed tolerance while the other algorithms have been stopped because they reached the prescribed maximum number of iterations. In order to compare PD-GMRES with another method, we implemented a rule for modifying the residual r_j if necessary, denoted as GMRES with a hybrid residual [27,26]. In addition, we also implemented a PD-GMRES with hybrid residual which modifies the parameter m_j and the residual r_j at each iteration (see Figs. 9 and 10).

The GMRES with hybrid residual assumes that slow convergence occurs when residuals are approximately parallel [26], so that it is detected by monitoring the cosine of the angle between residuals. The parameter m is fixed but when slow convergence is detected, a new hybrid residual is computed. The hybrid residual is obtained by taking an affine combination of r_j and r_{j-1}, i.e., $r_h = \alpha r_{j-1} + (1-\alpha) r_j$ where α is given by $\alpha = -\frac{\langle r_j, r_{j-1} - r_j\rangle}{\langle r_{j-1} - r_j, r_{j-1} - r_j\rangle}$, obtained minimizing $\|r_h\|$ (see [27]).

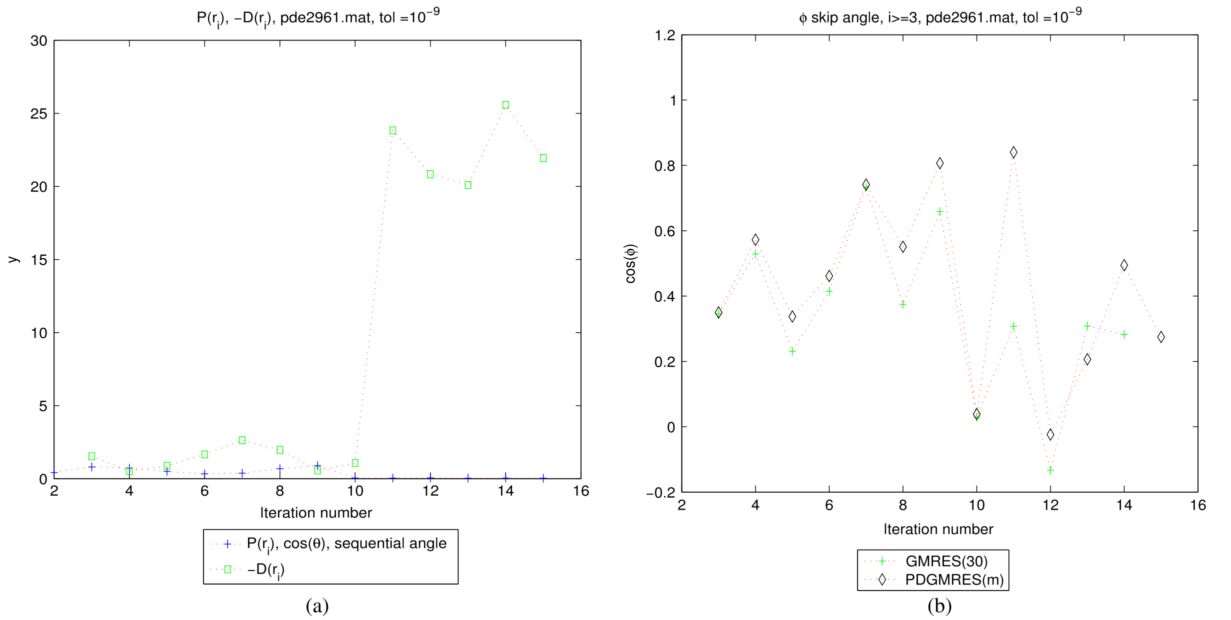


Fig. 8. Problem 7: (a) proportional (blue $+$) and derivative (green box $\square$) values without factors α_P and α_D calculated by PD-GMRES; (b) cosine skip angles, green $+$ for GMRES(30) and black diamond $\diamond$ for PD-GMRES(m). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

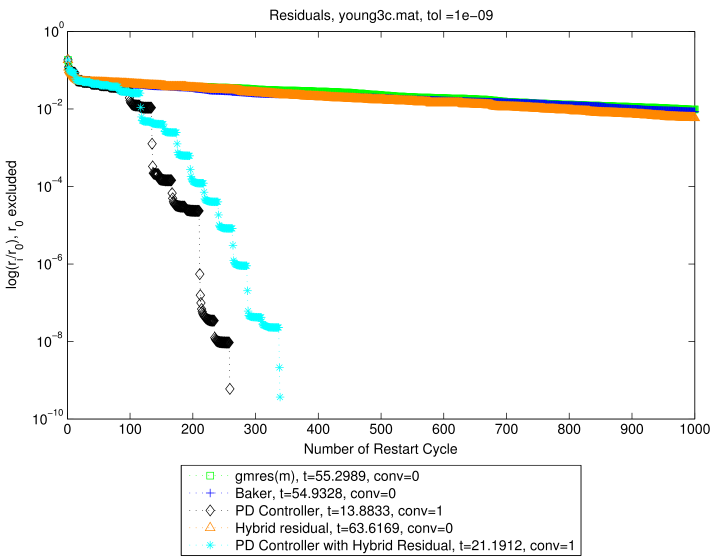


Fig. 9. Problem 15: Relative residual norms $\|r_j\|/\|r_0\|$ versus iteration number j. $\|r_0\|$ excluded.

Fig. 9 shows residuals for Problem 15. For this problem, GMRES(30), Baker's rule, Gonçalez's rule and even GMRES with hybrid residuals converge very slowly, and they do not converge even after 5000 iterations for the prescribed tolerance. However, the proposed PD-GMRES converges in less than 300 iterations. PD-GMRES with hybrid residual also converges but consumes more iterations than PD-GMRES. Fig. 10 shows the behavior of the restarting parameter m. Observe that in this case PD-GMRES increases m for reducing the residual norm.

5. Concluding remarks

In this paper, we explored the formulation of the restarted GMRES as a control problem where the restarting parameter m is the control variable. This formulation allows us to design an adaptive law (control law), in this case a proportional

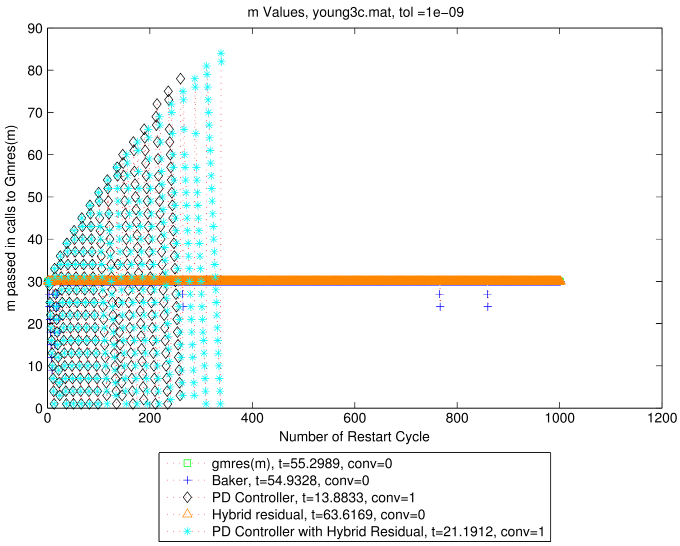


Fig. 10. Problem 15: Restart value m_j versus iteration number j.

derivative (PD) controller, to update the restarting parameter adaptively. The proposed controller was tested using benchmark matrix problems. The proposed PD-controller for GMRES(m) introduces, at each iteration, variations in the restarting parameter m to modify the dimension of the Krylov subspace and consequently the sequential and skip angles. This improves the convergence of GMRES(m) algorithm.

The Lyapunov function inspires the choice of a control law that generalizes some strategies presented in the literature. In addition, it leads to the introduction of a new term in the control law that depends on the tendency of the residual behavior. The resulting PD controller is more flexible than other strategies, and incorporates more information for modifying m. In summary, the proposed PD controller is more effective than other strategies because: (1) it is more flexible, therefore the controller itself decides at each iteration whether the value of m has to be decreased or increased; (2) the value of m can be increased to large values when necessary. This allows recovery of the convergence rate which typically deteriorates after the value of m has been successively decreased. (3) The tendency of the residual behavior is included in the decision of how and when to modify m. This is incorporated in the derivative part of the controller, which has the role of predicting the behavior, and incorporating this into the calculation of a new value for m. This combination of proportional and derivative terms results in more precise control of the restart parameter m, which is reflected in the improved performance of the proposed PD-GMRES algorithm. In addition, the PD-controller has the advantage that it is simple to implement using the norm of the residuals of the last three iterations.

In the proposed control law some parameters α_P and α_D have to be tuned. The choice affects the performance of the method. In this paper we have introduced a criterion for choosing those parameters. The controller also allows to incorporate a memory (history of m values) in the decision of the new value of m at each iteration. The latter was not analyzed and should help to further improve the performance of the method.

The Lyapunov function approach provides a sufficient condition for the convergence of GMRES(m). In this paper, we considered the simplest Lyapunov function, and the challenge is to come up with more sophisticated Lyapunov functions that lead to control laws, which deliver better performance of the corresponding algorithms.

Acknowledgments

This work was partially supported by CONACyT-Paraguay & CNPq-Brazil, through a grant BPP/PQ 310646/2016-2 to AB. CES acknowledges FEEI - PRONII - CONACyT and project FEEI - PROCIENCIA - CONACyT: 14-INV-186.

References

[1] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual method for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[2] I.C.F. Ipsen, C.D. Meyer, The idea behind Krylov methods, Amer. Math. Monthly 105 (1997) 889–899.
[3] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.
[4] A. Greenbaum, Iterative Methods for Solving Linear Systems, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.
[5] V. Simoncini, D.B. Szyld, Recent computational developments in Krylov subspace methods for linear systems, Numer. Linear Algebra Appl. 14 (1) (2007) 1–59.

[6] A.H. Baker, E.R. Jessup, T. Manteuffel, A technique for accelerating the convergence of restarted GMRES, SIAM J. Matrix Anal. Appl. 26 (4) (2005) 962–984.
[7] T. Gonçalez, R. da Cunha, Seleção dinâmica da dimensão do subespaço de Krylov no método GMRES(m) e suas variantes, TEMA: Tend. Mat. Apl. Comput. 7 (2006) 277–286.
[8] M. Embree, The tortoise and the hare restart GMRES, SIAM Rev. 45 (2003) 259–266.
[9] J. Zítko, Some remarks on the restarted and augmented GMRES method, Electron. Trans. Numer. Anal. - ETNA 31 (2008) 221–227.
[10] M. Eiermann, O.G. Ernst, O. Schneider, Analysis of acceleration strategies for restarted minimal residual methods, J. Comput. Appl. Math. 123 (2000) 261–292.
[11] W. Joubert, On the convergence behavior of the restarted GMRES algorithm for solving nonsymmetric linear systems, Numer. Linear Algebra Appl. 1 (5) (1994) 427–447.
[12] C.E. Schaerer, E. Kaszkurewicz, The shooting method for the solution of ordinary differential equations: a control-theoretical perspective, Int. J. Syst. Sci. 32 (8) (2001) 1047–1053.
[13] A. Bhaya, E. Kaszkurewicz, Control Perspectives on Numerical Algorithms and Matrix Problems, SIAM, Philadelphia, Philadelphia, PA, USA, 2006.
[14] A. Bhaya, E. Kaszkurewicz, A control-theoretic approach to the design of zero finding numerical methods, IEEE Trans. Automat. Control 52 (6) (2007) 1014–1026.
[15] I. Karafyllis, Feedback stabilization methods for the solution of nonlinear programming problems, J. Optim. Theory Appl. 161 (2014) 783–806.
[16] B. Goh, W. Leong, K. Teo, Robustness of convergence proofs in numerical methods in unconstrained optimization, in: Optimization and Control Methods in Industrial Engineering and Construction. Intelligent Systems, Control and Automation: Science and Engineering, Vol. 72, Springer, 2014, pp. 1–9.
[17] F. Pazos, A. Bhaya, Adaptive choice of the Tikhonov regularization parameter to solve ill-posed linear algebraic equations via Liapunov optimizing control, J. Comput. Appl. Math. 279 (2015) 123–132.
[18] K. Gustafsson, M. Lundh, G. Söderlind, A PI stepsize control for the numerical solution of ordinary differential equations, BIT 28 (2) (1988) 270–287.
[19] K. Kashima, Y. Yamamoto, System theory for numerical analysis, Automatica 43 (7) (2007) 1156–1164.
[20] Y. Yang, S.X. Ding, A control-theoretic study on iterative solutions to nonlinear equations for applications in embedded systems, Automatica 48 (2012) 583–594.
[21] E. Kaszkurewicz, A. Bhaya, P.R.V. Ramos, A control-theoretic view of diagonal preconditioners, Int. J. Syst. Sci. 26 (9) (1995) 1659–1672.
[22] U. Helmke, J. Jordan, A. Lanzon, A control theory approach to linear equation solvers, in: Proceedings 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006, pp. 1401–1407.
[23] U. Helmke, J. Jordan, Control and stabilization of linear equation solvers, in: J.C. Willems, S. Hara, Y. Ohta, H. Fujioka (Eds.), Perspectives in Mathematical System Theory, Control, and Signal Processing: A Festschrift in Honor of Yutaka Yamamoto on the Occasion of his 60th Birthday, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 73–82.
[24] L. Xie, Y. Liu, H. Yang, Gradient based and least squares based iterative algorithms for matrix equations $AXB + CX^TD = F$, Appl. Math. Comput. 217 (2010) 2191–2199.
[25] H. Zhang, F. Ding, Iterative algorithms for $X + A^TX^{-1}A = I$ by using the hierarchical identification principle, J. Franklin Inst. B 353 (2016) 1132–1146.
[26] M.A. Gomes-Ruggiero, V.L.R. Lopes, J.V. Toledo-Benavides, A safeguard approach to detect stagnation of GMRES(m) with applications in Newton - Krylov methods, Comput. Appl. Math. 27 (2008) 175–199.
[27] C. Brezinski, Hybrid methods for solving systems of equations, in: G.W. Althaus, E. Spedicato (Eds.), Algorithms for Sparse Large Scale Linear Algebraic Systems, 1998, pp. 271–290.
[28] A. Baker, E. Jessup, T. Kolev, A simple strategy for varying the restart parameter in GMRES(m), J. Comput. Appl. Math. 230 (2) (2009) 751–761.
[29] M. Sosonkina, L.T. Watson, R.K. Kapania, H.F. Walker, A new adaptive GMRES algorithm for achieving high accuracy, Numer. Linear Algebra Appl. 5 (4) (1998) 275–297.
[30] M. Habu, T. Nodera, GMRES(m) algorithm with changing restart cycle adaptively, in: Proceedings of Algoritmy 2000. Conference on Scientific Computing. Slovak Univ. of Tech, Bratislava, Comenius Univ., Bratislava, and Univ. of Technology, Zvolen; Vysoke Tatry-Podbanske, Slovakia, 2000, pp. 254–263.
[31] K. Moriya, T. Nodera, New adaptive GMRES(m) method with choosing suitable restart cycle m, in: R. Wyrzykowski, J. Dongarra, M. Paprzycki, J. Waśniewski (Eds.), Parallel Processing and Applied Mathematics: 5th International Conference, PPAM 2003, Czestochowa, Poland, September 7–10, 2003, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 1105–1113.
[32] L. Zhang, T. Nodera, A new adaptive restart for GMRES(m) method, in: R. May, A. J. Roberts (Eds.), Proc. of 12th Computational Techniques and Applications Conference CTAC-2004, Vol. 46, 2005, pp. C409–C425.
[33] R.B. Morgan, A restarted GMRES method augmented with eigenvectors, SIAM J. Matrix Anal. Appl. 16 (4) (1995) 1154–1171.
[34] R.B. Morgan, Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations, SIAM J. Matrix Anal. Appl. 21 (4) (2000) 1112–1135.
[35] V. Strecj, State Space Theory of Discrete Linear Control, John Wiley & Sons, 1981.
[36] F. Ding, X. Liu, X. Ma, Kalman state filtering based least squares iterative parameter estimation for observer canonical state space systems using decomposition, J. Comput. Appl. Math. 301 (2016) 135–143.
[37] M. Eiermann, O. Ernst, Geometric aspects of the theory of Krylov subspace methods, Acta Numer. 10 (2001) 251–312.
[38] B. Zhong, R.B. Morgan, Complementary cycles of restarted GMRES, Numer. Linear Algebra Appl. 15 (6) (2008) 559–571.
[39] A. Visioli, Practical PID Control (Advances in Industrial Control), Springer-Verlag, 2006.
[40] L. Xu, A proportional differential control method for a time-delay system using Taylor expansion approximation, Appl. Math. Comput. 236 (2014) 391–399.
[41] L. Xu, Application of the Newton iteration algorithm to the parameter estimation for dynamical systems, J. Comput. Appl. Math. 288 (2015) 33–43.
[42] F. Ding, H. Zhang, Gradient-based iterative algorithm for a class of the coupled matrix equations related to control systems, IET Control Theory Appl. 8 (2014) 1588–1595.
[43] L. Xu, F. Ding, The parameter estimation algorithms for dynamical response signals based on the multi-innovation theory and the hierarchical principle, IET Signal Process. 11 (2) (2017) 228–237.
[44] F. Ding, X. Liu, Y. Gu, An auxiliary model based least squares algorithm for a dual-rate state space system with time-delay using the data filtering, J. Franklin Inst. B 353 (2) (2016) 398–408.
[45] T.A. Davis, Y. Hu, The University of Florida sparse matrix collection, ACM Trans. Math. Software 38 (1) (2011) 1:1–1:25.