

A safeguard approach to detect stagnation of GMRES(m) with applications in Newton-Krylov methods*

MÁRCIA A. GOMES-RUGGIERO, VÉRA L. ROCHA LOPES and
JULIA V. TOLEDO-BENAVIDES

Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas
CP 6065, 13083-970 Campinas SP, Brazil

E-mails: marcia@ime.unicamp.br / vlopes@ime.unicamp.br / juliatoledob@yahoo.com.br

Abstract. Restarting GMRES, a linear solver frequently used in numerical schemes, is known to suffer from stagnation. In this paper, a simple strategy is proposed to detect and avoid stagnation, without modifying the standard GMRES code. Numerical tests with the proposed modified GMRES(m) procedure for solving linear systems and also as part of an inexact Newton procedure, demonstrate the efficiency of this strategy.

Mathematical subject classification: 65H10, 65F10.

Key words: linear systems, restarting GMRES, inexact Newton method, nonlinear systems.

1 Introduction

The objective of this work is to improve the performance of the restarted GMRES [25]. A well known difficulty with the restart of GMRES, algorithm for solving $Ax = b$, $A : n \times n$ is that it can *stagnate* when the matrix A is not positive definite [24], in the sense that the residual sequence does not converge to zero within a reasonable time frame. Simoncini, [27], and Sturler, [31], modified the GMRES using spectral analysis. In Parks et al., [21], Ritz values are used to improve the performance of the restart. Morgan, [16], [17], also uses eigenvectors for

#729/07. Received: 22/VI/07. Accepted: 28/II/08.

*Supported by FAPESP, CNPq, PRONEX-Optimization.

improving convergence of restarted GMRES. He considers some vectors from the previous subspace, adding them to the new subspace in order to deflate eigenvalues. Convergence issues and stagnation are discussed by Simoncini, [29], and Zavorin, [36]. Van der Vorst and Vuik introduce a strategy to prevent stagnation in GMRES, by including LSQR steps in some phases of the process [33].

This work aims an early detection of stagnation; once detected stagnation, the initial residue of the next cycle is steered away from the stagnation zone by means of a simple hybrid safeguard which mostly involves, in addition, the current residue. The strategies proposed here have the following objectives:

- avoid stagnation;
- use previous information given by the GMRES, avoiding any modification in it. At most, the new program should ask for few information besides that usually provided by GMRES;
- take into account information from the previous cycles performed by the GMRES.

It is also important to analyze the GMRES as a solver for the linear systems generated by Inexact-Newton type methods for solving nonlinear systems of equations. Our interest in choosing these methods relies, basically, on their popularity among practitioners, and on our research interest in Newton-Krylov methods, [10, 32]. Since strategies that improve the efficiency of GMRES, are fundamental in a better performance of these methods, they became one of the main subjects of our research.

In Section 2 we describe briefly the GMRES and the restarted GMRES algorithms. We also study the effect of the GMRES cycle length on the decrease of the residual norm. In Section 3 we establish our stagnation criteria and describe hybrid safeguards which modify the GMRES method, obtaining a version called here GMRESH. In Section 4 we show that GMRESH is capable to reduce considerably the effect of stagnation at the resolution of some linear systems. We also compare GMRESH with the GMRESDR method proposed by Morgan, [17]. In Section 5 we discuss the implementation of GMRESH within the Newton-Krylov method and test its performance on a ray-tracing problem and on a set of boundary value problems. Some concluding remarks are given in Section 6.

2 GMRES(m)

The method GMRES was proposed in [25] for solving linear systems $As = b$, where A is a nonsingular $n \times n$ matrix (not necessarily symmetric) and $b \in \mathbb{R}^n$. If s_0 is a initial approximation for the solution and $r_0 = b - As_0$ is the corresponding residual vector, the Krylov subspace after l iterations of the GMRES will be:

$$\mathcal{K}_l = [r_0, Ar_0, A^2r_0, \dots, A^{l-1}r_0]. \quad (1)$$

At each iteration l of GMRES, a value $s_l \in s_0 + \mathcal{K}_l$ is computed to minimize the residual vector, that is: $r_l = \min_{s \in s_0 + \mathcal{K}_l} \|b - As\|_2$. In what follows we always mean $\|\cdot\|_2$ whenever we use $\|\cdot\|$.

It is known that, computationally speaking, GMRES is more expensive than other Krylov subspace methods, such as Bi-CGSTAB, [14], QMR [24] for general square matrices, or LSQR [19], [20] for anti-symmetric matrices. Nevertheless, it is widely used for solving linear systems derived from the discretization of partial differential equations, since theoretically the 2-norm of the residual vector is minimized inside the Krylov subspace at each step.

We can describe GMRES as follows: given the subspace \mathcal{K}_l and the initial approximation s_0 , compute s_l , the approximate solution for $As = b$, where $s_l \in s_0 + \mathcal{K}_l$ in such a way that $r_l = b - As_l$ is orthogonal to $A\mathcal{K}_l$.

Since $s_l \in s_0 + \mathcal{K}_l$ we can write $s_l = s_0 + \delta$, $\delta \in \mathcal{K}_l$; then $r_l = b - As_l = r_0 - A\delta$. We obtain δ in such a way that r_l is orthogonal to $A\mathcal{K}_l$. Geometrically, $A\delta$ is the orthogonal projection of r_0 in $A\mathcal{K}_l$, as shown in Figure 1.

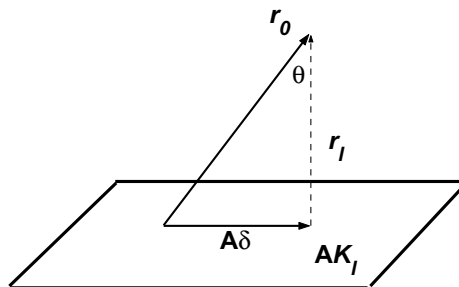


Figure 1 – Orthogonal projection of r_0 in $A\mathcal{K}_l$.

The dimension of the Krylov space keeps increasing, so the memory cost and complexity of the l -th GMRES step increase with l . A modified version called

GMRES(m) is used in large scale problems. In this version, the GMRES proceeds in cycles of m iterations, see [14], [25]. Basically, the process begins with some vector s_0 , and a fixed number m of iterations are performed. Then, a new cycle begins with s_m as initial approximation and $r_m = b - As_m$ as initial residue. Note that at each cycle an m -dimensional Krylov subspace is generated from the initial residue, following the usual GMRES procedure.

Whereas the restarted policy is computationally more feasible, convergence cannot be guaranteed in general, and stagnation becomes possible [11], [24], [28], [31] and [36]. A rather expensive remedy would be to monitor the eigenvalues of the Hessenberg matrices generated during the GMRES, [28]. Other schemes, such as the one mentioned in [31], store some vectors created at the j -th cycle and use them at the $(j + 1)$ -th cycle. We present a different strategy.

3 Stagnation

In this section we present a new strategy to generate an approximation to the new cycle that bypasses the stagnation of the method. We use the following notation: r_0^j is the initial residue of the j -th cycle and r_m^j is the residual vector at the end of this cycle.

Stagnation in GMRES(m) is usually described as *slowness in the decrease of the consecutive residual norms*, $\|r_l^j\|$, $l = 1, 2, 3, \dots, m$. However, a situation where r_m^j and r_0^1 are roughly linearly dependent could also be classified as stagnant.

To prevent stagnation we need to control the cycles in such a way that it is possible to make a comparison between the norm of the last residue, $\|r_m^j\|$, and the norm of the initial residual vector of this cycle, $\|r_0^j\|$. Moreover, we need to guarantee that the basis generated for the Krylov subspace in the $(j + 1)$ -th cycle is linearly independent with respect to the basis from the last cycles.

Firstly, we need to establish a criterion to detect stagnation, which can be based on the norm of the residual vectors. However, even in the case of a reasonable decrease of the norm of residues, if the angle between the initial and final residual vectors of one cycle j is close to zero, the Krylov subspace of the new cycle, $(j + 1)$, can be similar to the previous one. This is because both subspaces began with vectors that are almost linearly dependent. It is obvious that in such case

the progress of the process towards the solution will be very slow.

In Figure 1, we can observe that, given the projection property of GMRES, if the cosine of the angle between the initial and final residual vectors is close to 1, then the norm of these residues are very close to each other so that there is an equivalence between the tests

$$\|r_m^j\|/\|r_0^j\| \sim 1 \quad (2)$$

and

$$|\cos(\theta_j)| \sim 1, \quad (3)$$

where θ_j is the angle between the vectors r_0^j and r_m^j .

We must also consider the possibility of linear dependence between r_0^l , $l = 1, \dots, j - 1$ and r_m^j , where r_m^j is the last residue of the j -th cycle whereas r_0^l is the initial residue of the l -th cycle. Nevertheless we decided to make the comparison only between r_0^1 and r_m^j . Our stagnation criterion is based on simplicity and heuristics. Simplicity dictates a small number of angle comparisons; heuristics is justified as follows: comparison with r_0^1 is justified by the fact that, empirically, the largest descent occurs during the first GMRES cycles. Angle comparison with r_0^j is based on the heuristic assumption that once stagnation is present it will always repeat itself; in other words, if the angle between r_0^{j+1} and r_0^{j-1} were small, stagnation would have already been discovered in the previous cycle.

If a linear dependence between r_0^1 and r_m^j occurs, the Krylov subspace of the cycle $(j + 1)$ would be close to the Krylov subspace generated in the first cycle. This would lead to the stagnation of the process. If this is the case, there is no equivalence between the tests (2) and (3), since r_0^1 and r_m^j belong to different Krylov subspaces. Thus it is not possible to use the test with the norms of the residual vectors. Linear dependence can be detected testing the cosine of the angle between the residues r_0^1 and r_m^j :

$$|\cos(\theta_{j,1})| \sim 1. \quad (4)$$

In the strategies proposed in this work, the analysis will be always done at the end of each cycle, to reduce a too big computational costs. Stagnation will be declared when:

$$|\cos(\theta_j)| \sim 1 \quad \text{or} \quad |\cos(\theta_{j,1})| \sim 1. \quad (5)$$

In case of stagnation, another initial approximation must be chosen for the new cycle. This new approximation is obtained using information generated during the process. However, it needs to be constructed in such a way to guarantee a reduction in the norm of the residual vector. The new strategy that is proposed generates an approximation for the $(j + 1)$ -th cycle using a hybrid scheme, based on the strategy proposed by Brezinski and Redivo-Zaglia in 1994, [3]. This hybrid scheme uses the approximations s_0^1 and s_m^j which, in some sense, take into account the information generated by GMRES(m). In this way, we are trying to get out of a sequence which yields little decrease for the residuals. In what follows, we will briefly describe the scheme proposed in [3] for linear systems.

Consider the linear system of equations $As = b$. Once the approximations \bar{s} and \hat{s} are known, and the corresponding residues, $\bar{r} = b - A\bar{s}$ and $\hat{r} = b - A\hat{s}$ are computed, the objective is to construct a new approximation as a linear combination of \bar{s} and \hat{s} , $s = \alpha\bar{s} + \beta\hat{s}$, with the aim of reducing the residual norm. As a simplifying tool in obtaining these parameters is to fix $\beta = 1 - \alpha$, and then the corresponding residue r will be given by

$$r = b - As = \alpha\bar{r} + (1 - \alpha)\hat{r}.$$

Therefore our problem is reduced to find $\alpha \in \mathbb{R}$, the least square solution for:

$$\min_{\alpha} \|r\| = \min_{\alpha} \|(\bar{r} - \hat{r})\alpha + \hat{r}\|,$$

for which the optimal α is given by:

$$\alpha = -(\bar{r} - \hat{r})^T \hat{r} / (\bar{r} - \hat{r})^T (\bar{r} - \hat{r}). \quad (6)$$

The new approximation will be $s = \alpha\bar{s} + (1 - \alpha)\hat{s}$, and the corresponding residue is $r = \alpha\bar{r} + (1 - \alpha)\hat{r}$.

Let us go back to the solution for $As = b$ by the GMRES(m). Let s_0^1 and s_m^j denote the initial approximation of the whole process, and the last approximation obtained after performing the m GMRES iterations of the j -th cycle, respectively. The following safeguards are tested and the computation of the new approximation is done using the hybrid scheme, where \bar{s} corresponds to s_0^1 and \hat{s} to s_m^j .

Strategy H:

if $j \neq 1$:

Safeguard 1: test the angle θ_j between r_0^j and r_m^j :

if $|\cos(\theta_j)| \sim 1$, compute $s_0^{j+1} = \alpha s_0^j + (1 - \alpha)s_m^j$, with α given by (6).

Otherwise,

Safeguard 2: test the angle $\theta_{j,1}$ between r_0^1 and r_m^j :

if $|\cos(\theta_{j,1})| \sim 1$, then compute $s_0^{j+1} = \alpha s_0^1 + (1 - \alpha)s_m^j$, with α given by (6).

if $j = 1$:

Test the angle θ_1 between r_m^1 and r_0^1 :

if $|\cos(\theta_1)| \sim 1$, compute s_a as a random vector and $s_0^2 = \alpha s_a + (1 - \alpha)s_m^1$, with α given by (6).

In the case $j = 1$, due to the orthogonality and minimization properties of GMRES, the vector calculated from s_0^1 is the same as the one encountered by the hybrid process. Thus, s_0^1 should be modified. We add that the corresponding residual vector r_a , related to s_a , is normalized so as to guarantee the monotone decrease of the residues.

In Figure 2 we depict the situation tested by Safeguard 2, when the decrease in the residual norm is sufficient, so that Safeguard 1 is not triggered. Thus, r_m^j is necessarily much smaller than r_0^1 . If indeed the angle between them is small, or close to π , the residue formed by them will show a pronounced decrease in norm.

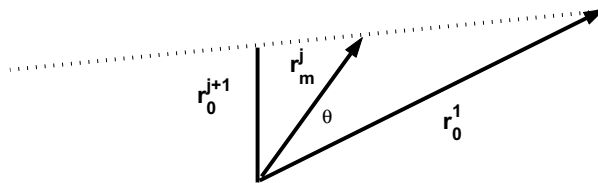


Figure 2 – Safeguard 2: angle between r_0^1 and r_m^j , and residue obtained by the hybrid scheme.

The hybrid scheme also presents the advantage of maintaining the minimization of the 2-norm of r , as is the purpose of GMRES(m). The important difference here is that the GMRES(m) solves this problem in the Krylov subspace generated in the last cycle of GMRES(m). In the hybrid scheme the minimization is carried out

in the plane generated by vectors belonging to two different Krylov subspaces: the first Krylov subspace (\mathcal{K}_1) and the last Krylov subspace (\mathcal{K}_j). Therefore our scheme keeps some information about these two subspaces.

The hybrid vectors are used quite often. This idea is similar to the *residual smoothing* used in [26], but in that case, it is used as a stopping criterion for the Conjugate Gradient method; in [34] the authors compared the behavior of the smooth residue and the usual residue for the MRS, QMR and BCG methods. Hybrid preconditioners are used in [23] and [33]. We do not use the term “hybrid GMRES” since in the literature it is sometimes used in other contexts, such as in [18], [30]. So we are calling it GMRESH.

4 Numerical experiments with GMRES and GMRESH

We present two examples comparing GMRES and GMRESH for 3×3 matrices.

Our procedure in this paper is the following: a hybrid restart is triggered in the first 5 occurrences of $\cos(\theta_j) > 0.8$ or $\cos(\theta_{j,1}) > 0.8$ and in the next 5 occurrences of $\cos(\theta_j) > 0.9$ or $\cos(\theta_{j,1}) > 0.9$. In all other next iterations, the usual GMRES(m) is used. The point is that if GMRESH(m) shows persistent stagnation then further progress is not likely to occur.

Example 1. Zavorin [35] brings the linear system $Ax = b$:

$$A = \begin{pmatrix} 3.64347104554523 & -1.30562625697964 & 2.12276233724947 \\ 3.81895186997748 & -0.33626408416579 & 8.43952325416869 \\ 0.12754105943518 & 0.13002776444227 & 2.98820549610000 \end{pmatrix} \quad (7)$$

and

$$b = \begin{pmatrix} -0.22385545043433 \\ -0.30471918583417 \\ 0.92576182418211 \end{pmatrix}.$$

For this example, the null vector was taken as the initial approximation. GMRES was applied with restarts at each two iterations and it was allowed 100 cycles. The sequence of GMRES(2) residues is constant and equal to 1, showing the stagnation of this method. For the GMRESH(2), the safeguards were triggered and the process converged in 19 iterations with precision 10^{-4} .

We also tested GMRES(2) and GMRESH(2) on 64 problems, which were created as follows: consider the systems $As = b_{ijl}$ where $b_{ijl} = b + v_{ijl}$, v_{ijl} being a vector in \mathbb{R}^3 with entries in $[-0.1, 0.1]$. We used 4 points in each interval. Figure 3 shows the logarithmic relative error $\|r_{end}\|/\|r_0^1\|$ for each problem, where r_{end} is the last residue of the whole process. Although both methods stagnated in some cases, GMRESH(2) shows a clear improvement.

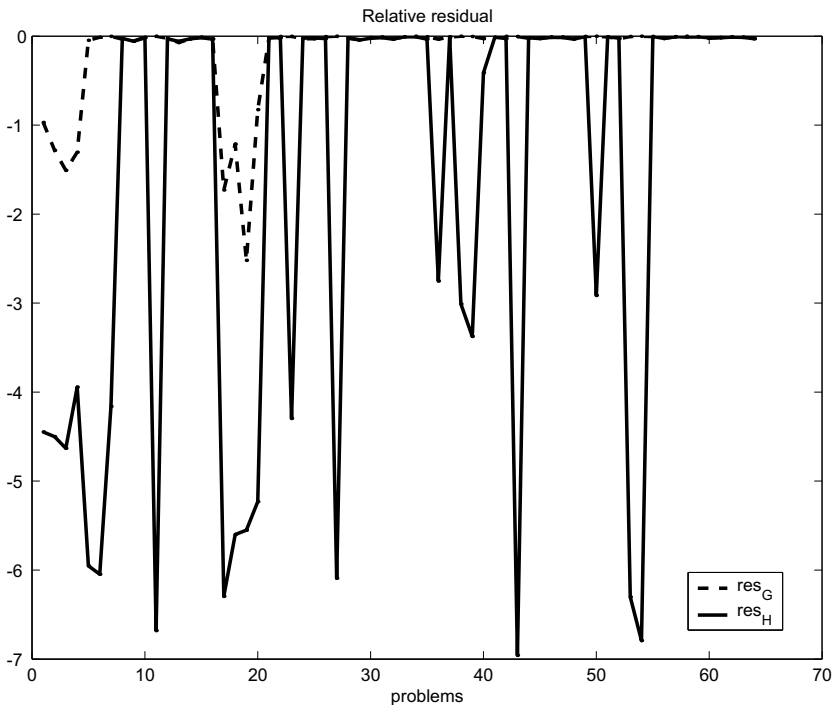


Figure 3 – Logarithm of relative residual norms for the matrix (7). res_G and res_H represent relative residues obtained by GMRES(2) and GMRESH(2), respectively.

Example 2. Embree [8] gives the linear system $Ax = b$:

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 2 \\ -4 \\ 1 \end{pmatrix} \quad (8)$$

as an example where GMRES(1) converges in three iterations but for GMRES(2) the relative residue stagnates near 0.3. We considered 1681 linear systems,

$As = b_{v\mu}$, where

$$r_0^1 = b_{v\mu} := (v, \mu, 1)^T, \quad v, \mu \in [-10, 10]$$

and 10^{-6} was taken as precision. Figure 4 shows the logarithm of the relative residual norms, $(\|r_{end}\|/\|r_0^1\|)$, ranging from 0 (white) to -8 (black). Actually, some of the GMRESH relative residues calculated were less than 10^{-16} . We can see that the GMRESH(2) relative residue is much smaller than the GMRES(2) relative residue in the vast majority of cases. Observing the size of the white region in both graphics, it is easy to conclude the better performance of GMRESH(2).

We applied the deflated GMRESDR method proposed by Morgan in [17] at the resolution of the linear systems (7) and (8). The results were obtained using the program GMRESDR.m, kindly sent to us by R. Morgan (Baylor University). The deflation of small eigenvalues can greatly improve the convergence of restarted GMRES. This method is denoted by GMRESDR(m, k), where m represents the maximum dimension of the subspace and k be the number of approximate eigenvectors (harmonic Ritz values) retained at a restart.

We observe that the matrices A of these systems are not ill-conditioned. At the resolution of system (7), GMRESDR(2,1) performed 69 inner iterations and obtained relative residual of 5.8544e-005. Meanwhile, GMRESH(2) performed 19 inner iterations as seen before. Considering the 1681 problems of type (8), we had difficulties in running GMRESDR(2,1) due to the ill conditioning introduced by the change of the eigenvalue done by this method.

In conclusion, GMRESH(2) did a better job in avoiding stagnation in these problems.

5 Numerical experiments

Here we apply the GMRES and GMRESH procedures, as linear solvers for the inexact Newton method with a nonmonotone line search, [1], for solving a nonlinear system

$$F(x) = 0, \quad F : \mathbb{R}^n \rightarrow \mathbb{R}^n.$$

In the inexact Newton methods, [5], the sequence x_k (called sequence of outer iterations) is generated by

$$x_{k+1} = x_k + s_k;$$

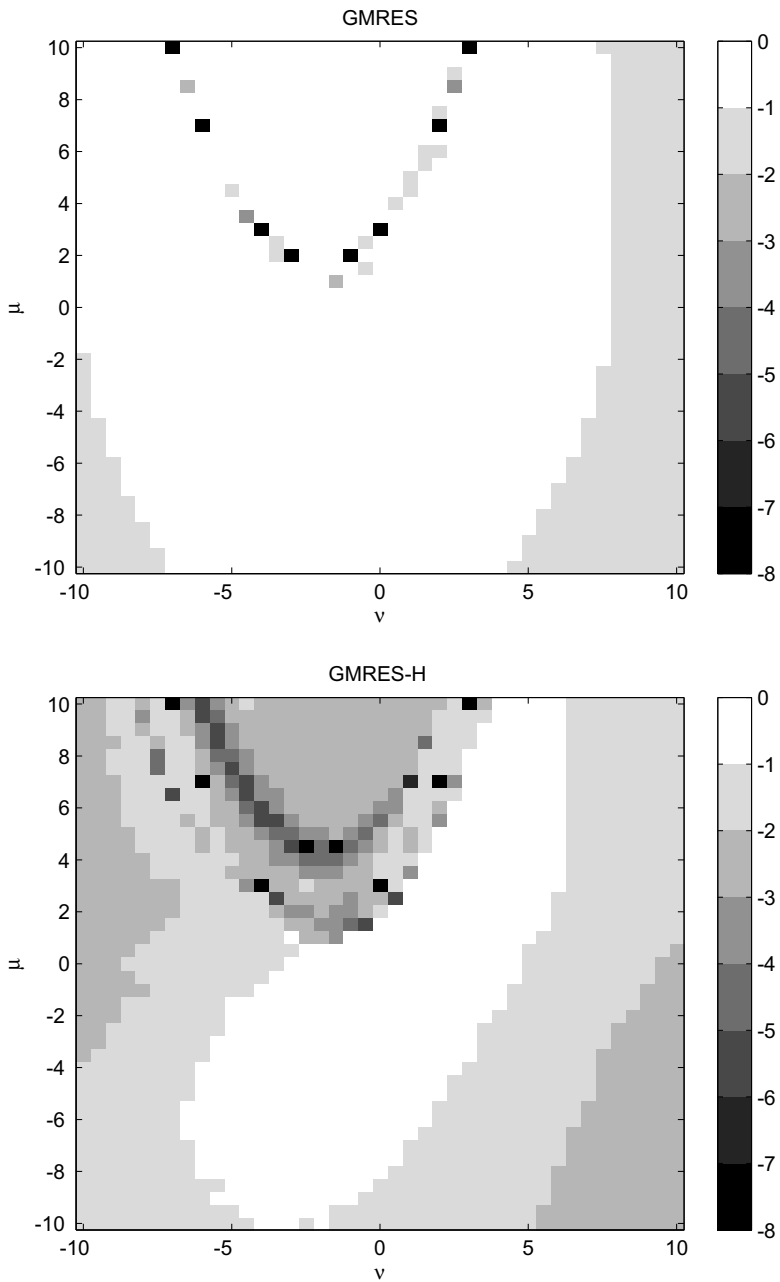


Figure 4 – Logarithm of relative residual norms about matrix (8).

s_k solves approximately the linear system $J(x_k)s = -F(x_k)$, using this stopping criterion:

$$\|J(x_k)s + F(x_k)\| \leq \eta_k \|F(x_k)\|, \quad (9)$$

where $J(\cdot)$ is the Jacobian matrix, $\eta_k \in (0, 1]$ is the tolerance which is called *the forcing term*, [7].

In the line search procedure, it is needed the following parameters: $\sigma \in (0, 1)$, ϱ_{\min} and ϱ_{\max} such that $0 < \varrho_{\min} < \varrho_{\max} < 1$ and the sequence $\{\mu_k\}$ such that $\mu_k > 0$ for all $k = 0, 1, 2, \dots$ and $\sum_{k=0}^{\infty} \mu_k = \mu < \infty$.

Now we present the inexact Newton algorithm with the nonmonotone line search procedure. Let $x_0 \in \mathbb{R}^n$ be an arbitrary initial approximation to the solution for $F(x) = 0$. Given $x_k \in \mathbb{R}^n$, and the tolerance $\varepsilon > 0$, the steps to obtain a new iteration x_{k+1} are the following:

Algorithm 1. (Inexact Newton method with nonmonotone line search):

While $\|F(x_k)\| > \varepsilon$, perform steps 1 to 4:

Step 1: Choose η_k .

Step 2: Find s_k such that $\|F(x_k) + J(x_k)s_k\| \leq \eta_k \|F(x_k)\|$;

Step 3: Take $\xi = 1$, compute $x_{aux} = x_k + s_k$ and $F(x_{aux})$.

While

$$\|F(x_{aux})\| > [1 - \xi\sigma]\|F(x_k)\| + \mu_k,$$

perform the steps 3.1 and 3.2:

step 3.1: compute $\xi_{new} \in [\varrho_{\min}\xi, \varrho_{\max}\xi]$;

step 3.2: set $\xi = \xi_{new}$ and compute $x_{aux} = x_k + \xi s_k$.

Step 4: Take $\xi_k = \xi$, compute $x_{k+1} = x_k + \xi_k s_k$ and update k .

In the Step 1 we examine the following choices for the forcing term:

Constant (Cte): we chose $\eta_k = 0.1$;

EW1: $\eta_k = \frac{\|F(x_k) - F(x_{k-1}) - J(x_{k-1})s_{k-1}\|}{\|F(x_{k-1})\|}$ (see Eisenstat and Walker [7]);

EW2: $\eta_k = \gamma \left(\frac{\|F(x_{k+1})\|}{\|F(x_k)\|} \right)^\alpha$, $\gamma \in [0, 1]$, $\alpha \in (1, 2]$ (see Eisenstat and Walker [7]);

GLT: $\eta_k = [1/(k + 1)]^\rho \cos^2(\phi_k) \frac{\|F(x_k)\|}{\|F(x_{k-1})\|}$, $\rho > 1$ and $-\pi/2 \leq \phi_k \leq 0$, (see Gomes-Ruggiero et al. [10]).

The vector s_k of the Step 2 is obtained by $\text{GMRES}(m)$ and $\text{GMRESH}(m)$.

The line search performed at Step 3 by Algorithm 1 follows the one proposed in [1] which is a nonmonotone strategy similar to the one introduced by Li and Fukushima, [12]. So, Algorithm 1 has global convergence [10]. Besides that, with the choices EW1, EW2 and GLT the convergence rate is superlinear, [7], [10].

5.1 Implementation features

We give now more details about the implementation of the algorithms. The implementation details can be found in [32], pages 26 and 49. All the tests were performed in a Centrino Duo 1.6GHz with 1 GB Ram computer, using the software MatLab 6.1.

- Line search procedure (Step 3 of Algorithm 1):
 for the parameter σ , we took $1.d - 04$;
 if the vector $x_{aux} = x_k + \xi s_k$ does not give an acceptable decrease in the value of the function, then we compute the new step size as $\xi_{new} = 0.5\xi$;
 for the sequence μ_k , we define:

$$\begin{aligned} ftip(0) &= \|F(x_0)\|, \\ ftip(k) &= \min\{\|F(x_k)\|, ftip(k - 1)\}, \text{ if } k \text{ is a multiple of } 3 \text{ and} \\ ftip(k) &= ftip(k - 1), \text{ otherwise;} \end{aligned}$$

then, we set:

$$\mu_k = \frac{ftip(k)}{(k + 1)^{1.1}}.$$

- The initial value and safeguards for η :
 for all the choices for η_k we set the initial value $\eta_0 = 0.1$. For the choices EW1 and EW2 of [7] and for the choice GLT, we take $\eta_k = \min\{\eta_k, 0.1\}$ if $k \leq 3$, and $\eta_k = \min\{\eta_k, 0.01\}$ if $k > 3$. We also take $\eta_k = 0.1$ when $\phi_k > 0$. At the final iterations we have adopted the safeguard introduced in [22] which can be described as: since the linear model is $F(x) \sim F(x_k) + J(x_k)s$, at the final iterations, we can have: $\|F(x_{k+1})\| \sim \|F(x_k) + J(x_k)s_k\| \leq \eta_k \|F(x_k)\|$. In this case it is important to set η_k such

that $\eta_k \|F(x_k)\| \sim \varepsilon$ where ε is the precision required for the nonlinear system. A safeguard which represents these ideas is: if $\eta_k \leq 2\varepsilon$ then we set $\eta_k = (0.8\varepsilon) \|F(x_k)\|$.

- **Parameters choice for η_k :**
for the choice EW2 it was taken $\gamma = 1$ and $\alpha = 0.5(1 + \sqrt{5})$ and for the choice GLT it was taken $\rho = 1.1$.
- **Stopping criterion:**
the process is finished successfully if $\|F(x_k)\| \leq 10^{-6}$ and $k < 100$.
- **Restarts and the maximum number of iterations in GMRES(m):**
we fix the restarts at each m iterations, $m = 30$ or $m = 50$, allowing initially a maximum of 100 cycles ($100m$ iterations). This maximum number is called here by `maxit` and it is adjusted during the process. This is the case, if the value of $\|F\|$ increases, when F is computed at the solution obtained after an inner iteration, before doing the line search. Also, `maxit` is adjusted when the number of GMRES iterations have exceeded a certain value. Indeed, `maxit` is computed according to: if $1 < \|F(x_{new})\|/\|F(x_{old})\| < 100$, then `maxit` is fixed as 50; if $\|F(x_{new})\|/\|F(x_{old})\| > 100$ or if the maximum number of GMRES iterations has been exceeded, then `maxit` is fixed as 30. We observe that this procedure is repeated just in two consecutive iterations. After that, the value of `maxit` is taken as 100 again.
- **Strategy H:**
after each GMRES iteration, a possible stagnation is detected by the tests (3) and (4). Initially, we use 0.9 as a tolerance for the value of the cosine of θ_j or $\theta_{j,1}$. If the hybrid scheme is triggered 5 times, we change this tolerance to 0.8. In the case of the hybrid process be triggered at the beginning of 10 cycles, we stop the test as in Section 4.

We present some numerical results obtained from the solution for a ray-tracing problem [13] and also from a set of boundary value problems.

5.2 A ray-tracing problem, [2]

We will present here a very simplified ray-tracing problem. First of all, let us represent the earth surface, as if it was two dimensional. Imagine that an acoustic wave transmitter **S** and a geophone receiver **G**, are located in two different points on the earth surface. Each acoustic wave will cross a certain number of layers, under the surface, and when coming back, it will be captured by the geophone. Such layers considered elastic, isotropycs and homogeneous, compose the structure of the undersurfaces of the earth.

The problem that we are interested in, consists on determining the trajectory of a ray, when crossing the earth undersurface [13].

The interfaces between two consecutive layers, are defined as functions of the horizontal coordinate x . These functions are continuous and smooth – we will not consider the intersections between two consecutive interfaces. The functions will be represented by $z = f_i(x)$, $i = 1, \dots, m$, where m corresponds to the number of interfaces and z denotes the vertical coordinate.

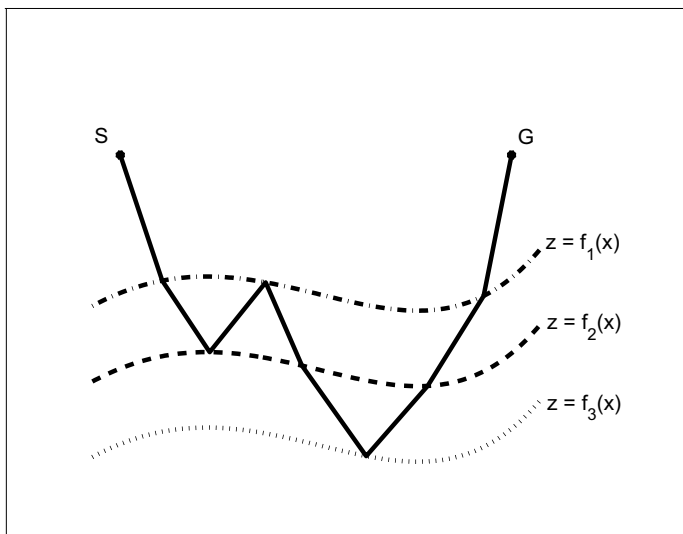


Figure 5 – A ray-tracing model with two layers and signature $a = (2, 1)$.

The number of times that a ray crosses the layer – downwards or upwards – in a layer between the interfaces f_i and f_{i+1} will be denoted by (a_i) , $i = 1, \dots, m - 1$. Figure 5 is an example of this, where $a_1 = 2$ and $a_2 = 1$. The

vector $a = (a_1, \dots, a_{m-1})$ is called *the signature* of the ray. For uniqueness, we assume that all the downward crossings in a layer occur before those in the next layer, as in Figure 5.

Under these conditions, the rays must satisfy the Snell law, and so, they describe a straight line trajectory in each layer. Let the initial point S , the end point G on the earth surface, and the velocity of the ray in the j -th layer v_j , $j = 1, \dots, m - 1$, be given. The problem consists on finding the points $X_k = (x_k, f_{ik}(x_k))$, for $k = 1, 2, \dots, n$ that satisfy the Snell law in each reflector i_k . We will take $X_0 = S$ and $X_{n+1} = G$. The dimension of the problem is given by

$$n = 2 \sum_{j=1}^{m-1} a_j + 1.$$

The tangent vector to the k -th interface at the point X_k , will be called τ_k , and is defined by $\tau_k = (1, f'_{ik}(x_k))$. The Snell law is given by

$$\frac{1}{v_{i_k}} \frac{\tau_k^T (\mathbf{X}_k - \mathbf{X}_{k-1})}{\|\tau_k\| \|\mathbf{X}_k - \mathbf{X}_{k-1}\|} = \frac{1}{v_{i_{k+1}}} \frac{\tau_k^T (\mathbf{X}_{k+1} - \mathbf{X}_k)}{\|\tau_k\| \|\mathbf{X}_{k+1} - \mathbf{X}_k\|}, \quad (10)$$

where $\|\cdot\|$ denotes the Euclidean norm. Thus, using equation (10) for $k = 1, \dots, n$ we obtain a nonlinear system of equations with n equations and n unknowns, given by

$$\Phi(\mathbf{x}) = 0, \quad (11)$$

with $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_n(\mathbf{x}))^T$ and the functions $\phi_k : \mathbb{R}^n \rightarrow \mathbb{R}$ ($k = 1, \dots, n$) defined by

$$\begin{aligned} \phi_k(\mathbf{x}) = & v_{i_{k+1}} \frac{(x_k - x_{k-1}) + f'_{i_k}(x_k) (f_{i_k}(x_k) - f_{i_{k-1}}(x_{k-1}))}{\left[(x_k - x_{k-1})^2 + (f_{i_k}(x_k) - f_{i_{k-1}}(x_{k-1}))^2 \right]^{1/2}} \\ & - v_{i_k} \frac{(x_{k+1} - x_k) + f'_{i_k}(x_k) (f_{i_{k+1}}(x_{k+1}) - f_{i_k}(x_k))}{\left[(x_{k+1} - x_k)^2 + (f_{i_k}(x_{k+1}) - f_{i_k}(x_k))^2 \right]^{1/2}}. \end{aligned} \quad (12)$$

For the tests presented in Table 1, we considered a ray-tracing problem with two horizontal layers and signature $a = (500, 500)$. Each ordered pair represents

the result of GMRES (left) and GMRESH (right). Column *iterex* represents the total number of external iterations that were performed; *iterin*, the number of inner iterations, that is, the number of iterations performed by GMRES(*m*) in the whole process; the column *Feval* represents the total number of function evaluations needed during the whole process; finally, *CPU time* represents the CPU time in seconds. The stopping criterion was $\|\Phi\| \leq 10^{-5}$.

We observe that in this example the second safeguard in Strategy **H** was never triggered, while the first safeguard decreased considerably the number of inner iterations.

η_k	<i>iterex</i>	<i>iterin</i>	<i>Feval</i>	<i>CPU time</i>
Cte	(6, 4)	(7000, 4978)	(7, 5)	(135, 104)
EW1	(7, 4)	(7840, 4054)	(8, 5)	(151, 84)
EW2	(7, 5)	(7924, 4024)	(8, 6)	(153, 85)
GLT	(7, 5)	(7986, 4086)	(8, 6)	(155, 83)

Table 1 – Results for the ray-tracing problem with GMRES(30).

5.3 Boundary value problems

The general formulation of the boundary value problems solved in this work is finding $u : \Omega = [0, 1] \times [0, 1] \rightarrow \mathbb{R}$, such that, for $\lambda \in \mathbb{R}$,

$$-\Delta u + h(\lambda, u) = f(s, t), \text{ in } \Omega, \quad u(s, t) = 0 \text{ on } \partial\Omega. \tag{13}$$

The real valued function $h(\lambda, u)$, the different values for the parameter λ and the function f define the different problems tested. All the problems were discretized using central differences on a grid with L inner points in each axis. The discretized system obtained has L^2 equations and variables. In these problems we work with grids of $L = 63$ and $L = 127$ inner points. We now make a brief description of the particular problems that were solved:

- Bratu problem: the function h is given by $h(\lambda, u) = \lambda \exp(u)$, and the function $f(s, t)$ is constructed so that $u^*(s, t) = 10st(1 - s)(1 - t)e^{s^{4.5}}$ is the exact solution for the problem. When $\lambda < 0$, the problem is considered relatively easy; not surprisingly, the hybrid strategy has never been triggered in our tests. The problem is more difficult for $\lambda > 0$, [9];

- a convection-diffusion problem: in this problem, the function h is given by $h(\lambda, u) = \lambda u(u_s + u_t)$, where u_s and u_t denote the partial derivatives of the function u with respect to s and t , and again the function $f(s, t)$ is defined so that $u^*(s, t) = 10st(1 - s)(1 - t)e^{s^{4.5}}$ is the exact solution for the problem. This is a problem considered difficult to solve [9], in particular for values of λ greater than 50;
- a third problem: P3. This problem appears in the book of Briggs, Henson and McCormick [4], page 105. In this case, $h(\lambda, u)$ is given by $h(\lambda, u) = \lambda ue^u$ and the function

$$f(s, t) = ((9\pi^2 + \gamma e^{(x^2-x^3)\sin(3\pi y)})(x^2 - x^3) + 6x - 2) \sin(3\pi y).$$

Table 2 shows the results of Newton-GMRES and Newton-GMRESH applied to the above problems, with: $L = 63, n = 3969, x_0 = (0, 0, \dots, 0)^T$ and $\lambda = 100$. This value for λ was chosen by the occurrence of stagnation. Each ordered pair should be read as in Table 1 and `iterex`, `iterin`, `feval` and `CPU time` have the same meaning as before.

η_k	problem	iterex	iterin	Feval	CPU time
Cte	Bratu	(21, 9)	(32015, 11206)	(22, 10)	(161.52, 60.27)
	conv-dif	(20, 20)	(5826, 7987)	(75, 71)	(70.80, 80.44)
	P3	(5, 5)	(167, 167)	(6, 6)	(1.83, 2.22)
EW1	Bratu	(25, 9)	(36793, 8977)	(26, 10)	(190.69, 47.53)
	conv-dif	(21, 21)	(8885, 7267)	(77, 79)	(85.36, 81.36)
	P3	(5, 5)	(167, 167)	(6, 6)	(2.17, 2.05)
EW2	Bratu	(17, 9)	(23450, 9605)	(18, 10)	(121.05, 54.48)
	conv-dif	(21, 21)	(8884, 7294)	(77, 79)	(85.69, 78.67)
	P3	(4, 4)	(173, 173)	(5, 5)	(2.20, 2.22)
GLT	Bratu	(19, 9)	(27517, 10709)	(20, 10)	(147.28, 61.70)
	conv-dif	(20, 21)	(7582, 6425)	(73, 78)	(80.14, 74.60)
	P3	(5, 5)	(200, 200)	(6, 6)	(2.25, 2.34)

Table 2 – Results for the boundary value problems with GMRES (30) .

For the Bratu problem, GMRESH shows a considerable decrease in the number of inner iterations, thus effectively mitigating the stagnation problem. The line

search was not triggered and `iterex` is slightly reduced. In the convection-diffusion problem, the number of inner iterations was reduced with GMRES in most of the cases. For this problem the Jacobian matrices are always ill-conditioned near the solution. The line search was activated several times, indicating stagnation coupled with insufficient decrease. For all the problems considered, `iterex` was practically the same (or slightly better) using GMRESH. Problem P3 can be considered an easy problem with no occurrence of stagnation.

We show in Table 3, the performance of the Algorithm 1 with GLT choice for the forcing term at the resolution of Bratu problem with $\lambda = 100$. Our purpose is to compare the number of inner iterations performed by GMRES and GMRESH at each outer iteration. We also show how many times the Strategy **H** is triggered. We observe the occurrence of stagnation at 4-th outer iteration. After this iteration, the strategy **H** was triggered with a very good performance since the number of inner iterations was extremely reduced and the convergence was accelerated.

We chose the Bratu and convection-diffusion problems to exploit other parameters, such as the grid size $L = 63$ and $L = 127$. With these choices the dimension of the nonlinear systems solved were 3969 and 16129, respectively. We also implemented GMRES (m) with $m = 30$ and $m = 50$ for $L = 63$ and $m = 50$ and $m = 80$ for $L = 127$. Table 4 shows the results for these systems. Observe that even with a very large dimension there was no difficulty for the Algorithm 1 to solve the corresponding nonlinear systems, in both cases: GMRES (m) and GMRESH (m). As before, for the Bratu problem, GMRESH (m) shows a considerable decrease in the number of inner iterations, thus effectively mitigating the stagnation problem.

We also tested GMRESDR(30,5) at the resolution of the linear system generated at the first outer iteration of Algorithm 1. It was slightly better than GMRESH(30) in solving Bratu problem but much better when solving convection-diffusion problem which has a very ill conditioned Jacobian matrix.

5.4 *The performance profile of the methods*

The performance profile, proposed by Dolan and Moré [6], is a useful tool to compare a set of algorithms used for solving a set of problems. As comparison

iterex	iterin	strategy H	$\ F\ _2$
0			(2.6964e+03, 2.6964e+03)
1	(23, 23)	(0, 0)	(1.7464e+03, 1.7464e+03)
2	(20, 20)	(0, 0)	(2.8962e+02, 2.8962e+02)
3	(474, 316)	(0, 5)	(1.5226e+01, 1.4708e+01)
4	(3000, 3000)	(0, 15)	(2.8250e+00, 1.9868e+00)
5	(900, 900)	(0, 25)	(1.6173e+00, 4.6646e-01)
6	(900, 360)	(0, 32)	(1.0029e+00, 1.4749e-02)
7	(3000, 3000)	(0, 42)	(1.9558e-01, 1.1036e-03)
8	(900, 90)	(0, 44)	(1.0517e-01, 7.4272e-06)
9	(900, 3000)	(0, 54)	(5.6424e-02, 5.0644e-07)
10	(3000, -)		(7.0456e-03, -)
11	(900, -)		(3.7184e-03, -)
12	(900, -)		(1.9620e-03, -)
13	(3000, -)		(2.3322e-04, -)
14	(900, -)		(1.2304e-04, -)
15	(900, -)		(6.4918e-05, -)
16	(3000, -)		(7.7044e-06, -)
17	(900, -)		(4.0649e-06, -)
18	(900, -)		(2.1446e-06, -)
19	(3000, -)		(2.5451e-07, -)

Table 3 – Performance of GMRES (30) and GMRESH (30) at the linear systems in Bratu problem with $\lambda = 100$.

measures, we can use, for instance, the number of iterations performed, the number of function evaluations, the CPU elapsed time, etc.

In this subsection we analyze the performance profile of the Algorithms Newton-GMRES and Newton-GMRESH when applied to solve the boundary value problems. A total of 18 problems were tested: Bratu, convection-diffusion and P3 with the following values for $\lambda := 10, 25, 30, 50, 75$ and 100. We compare the performance of this Algorithms using for η_k only the choices EW2 and GLT, in order to get an understandable figure. We used a grid with $L = 63$ inner points and $m = 30$ because with these parameters the average behaviour of both Algorithms is well represented. In the legends of Figure 6, Algorithm Newton-GMRESH is indicated by an H, after the name of the choice.

(L, m)	η_k	problem	iterex	iterin	Feval	CPU time
(63, 30)	EW2	Bratu	(17, 9)	(23450, 9605)	(18, 10)	(121, 54)
		conv-dif	(21, 21)	(8884, 7294)	(77, 79)	(86, 79)
	GLT	Bratu	(19, 9)	(27517, 10709)	(20, 10)	(147, 62)
		conv-dif	(20, 21)	(7582, 6425)	(73, 78)	(80, 75)
(63, 50)	EW2	Bratu	(9, 7)	(17772, 8171)	(10, 8)	(120, 58)
		conv-dif	(19, 19)	(2730, 2814)	(73, 73)	(53, 58)
	GLT	Bratu	(9, 7)	(17706, 5738)	(10, 8)	(100, 37)
		conv-dif	(19, 19)	(2505, 2544)	(71, 71)	(51, 54)
(127, 50)	EW2	Bratu	(20, 9)	(47171, 17991)	(21, 10)	(1093, 478)
		conv-dif	(30, 21)	(35090, 13730)	(107, 79)	(2108, 1144)
	GLT	Bratu	(20, 8)	(47415, 13746)	(21, 9)	(1092, 393)
		conv-dif	(20, 21)	(8524, 13793)	(72, 80)	(1014, 1140)
(127, 80)	EW2	Bratu	(15, 8)	(51705, 21510)	(16, 9)	(1713, 689)
		conv-dif	(20, 20)	(7256, 8596)	(79, 79)	(1011, 1084)
	GLT	Bratu	(16, 8)	(59862, 19930)	(17, 9)	(1903, 740)
		conv-dif	(19, 19)	(6303, 7005)	(70, 71)	(975, 973)

Table 4 – Results for boundary value problems with $L = 63$, GMRES(30), GMRES(50) and $L = 127$, GMRES(50), GMRES(80).

The measures used for comparison were the total numbers of inner and outer iterations. Note that in Figure 6 a higher curve means better performance. With respect to both measures, it is evident that the two Newton-GMRESH (NGH) outperformed the two Newton-GMRES (NG) versions. In particular, the percentage of problems solved with a small number of inner (resp. outer) iterations was roughly 43–53% against 30% (resp. 80–95% against 60–70%). It can be observed that NG versions solved some problems with a number of inner iterations 3 and 4 times greater than the minimum required to solve that problem with NGH versions. Thus the algorithm NG had a worse performance than the new version NGH. The strategies worked successfully.

6 Conclusions

In this work we presented an inexact Newton-like Algorithm with a nonmonotone line search, in which it was introduced an strategy to prevent stagnation of the linear solver GMRES. This strategy showed as advantages: (i) the simplicity

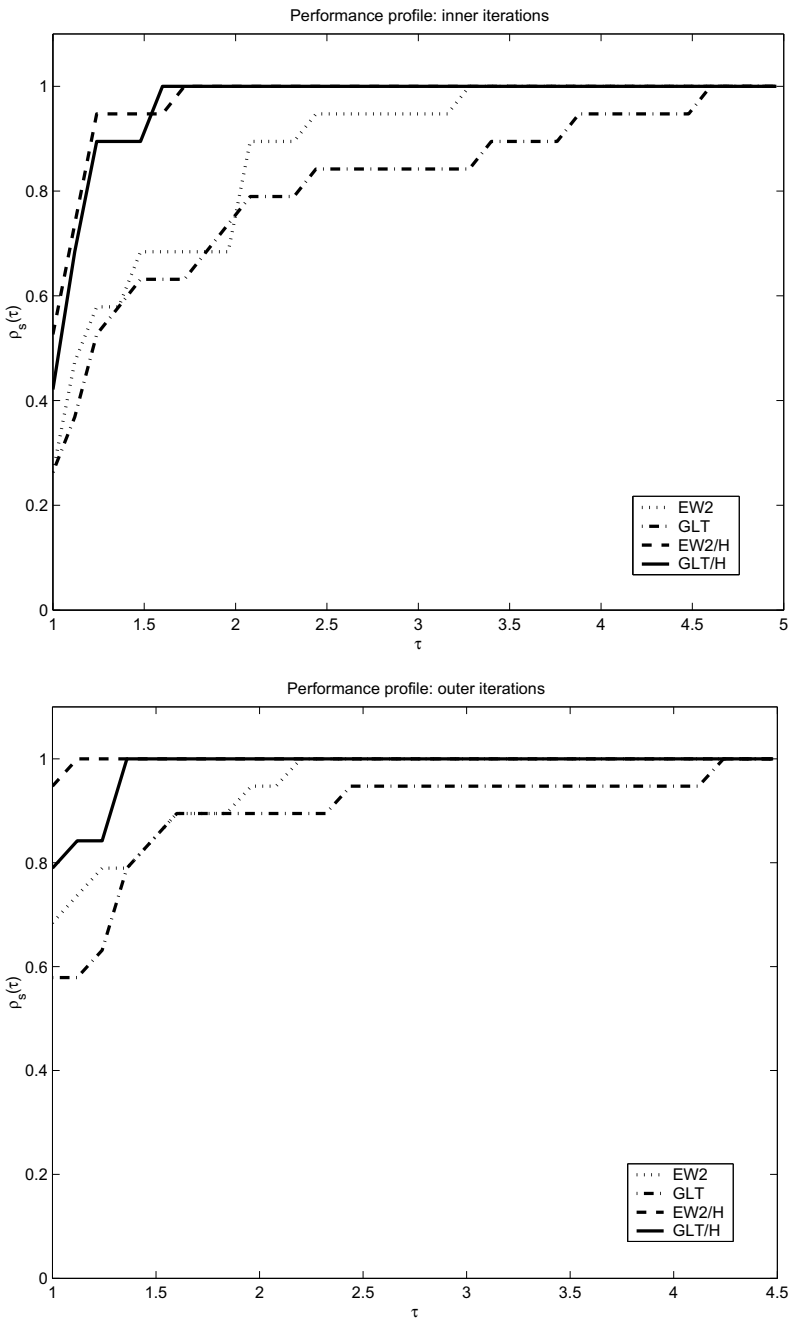


Figure 6 – Performance profile using as measures: inner iterations on the top and outer iterations on the bottom.

of implementation, since it does not requires interfering in the inner procedure of the linear solver GMRES; (ii) it can be monitored at each iteration; (iii) from the numerical results obtained, we can conclude that this strategy is efficient, either in the test for detecting the stagnation of the inner solver, or with respect to the safeguards triggered in this case. This conclusion can be seen in the solution of the ray-tracing problem, showed in Table 1, as well as in the solution of a bunch of boundary value problems whose performance profile is showed in Figure 6; (iv) GMRESH(m) is more efficient at the resolution of no ill conditioning stagnated systems. Maybe we can use GMRESH(m) together with GMRESDR(m, k) to obtain a more efficient algorithm.

Acknowledgements. The authors are indebted to an anonymous referee for the careful reading of this paper and also for the suggestions.

REFERENCES

- [1] E.G. Birgin, N. Krejić and J.M. Martínez, *Globally convergent inexact quasi-Newton methods for solving nonlinear systems*. Numer. Algorithms, **32** (2003), 249–260.
- [2] N. Bleistein, *Mathematical Methods for Wave Phenomena*. Academic Press, (1984).
- [3] C. Brezinski and M. Redivo-Zaglia, *Hybrid procedures for solving linear systems*. Numer. Math., **67** (1994), 1–19.
- [4] W.L. Briggs, V.E. Henson and S.F. McCormick, *A Multigrid Tutorial*. Second Edition, SIAM, Philadelphia (1987).
- [5] R.S. Dembo, S.C. Eisenstat and T. Steihaug, *Inexact Newton methods*. SIAM J. Numer. Anal., **19** (1982), 401–408.
- [6] E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profile*. Math. Program. ser. **A91** (2002), 201–213.
- [7] S.C. Eisenstat and H.F. Walker, *Choosing the forcing terms in inexact Newton method*. SIAM J. Sci. Comput., **17** (1996), 16–32.
- [8] M. Embree, *The tortoise and the hare restart GMRES*. SIAM Review, **45** (2003), 259–266.
- [9] M.A. Gomes–Ruggiero, D.N. Kozakevich and J.M. Martínez, *A numerical study on large-scale nonlinear solvers*. Comput. Math. Appl., **32** (1996), 1–13.
- [10] M.A. Gomes-Ruggiero, V.L. Rocha Lopes and J.V. Toledo-Benavides, *A new choice for the forcing term and a global convergent inexact Newton method*. Ann. Oper. Res., **157** (2008), 193–205.

- [11] W. Joubert, *On the convergence behavior of the restarted GMRES algorithm for solving nonsymmetric linear systems*. Numer. Linear Algebra Appl., **1** (1994), 427–447.
- [12] D-H. Li and M. Fukushima, *Derivative-free line search and global convergence of Broyden-like method for nonlinear equations*. Optim. Methods Softw., **13** (2000), 181–201.
- [13] H.B. Keller and D.J. Perozzi, *Fast seismic ray-tracing*. SIAM J. Appl. Math., **43** (1983), 981–992.
- [14] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia (1995).
- [15] L.F. Mendonça and V.L. Rocha Lopes, *Uma análise comparativa do desempenho de métodos quase-Newton na resolução de problemas em sísmica*. Tema-SBMAC, **5** (2004), 107–116.
- [16] R.B. Morgan, *A restarted GMRES method augmented with eigenvectors*. SIAM J. Matrix Anal. Appl., **16** (1995), 1154–1171.
- [17] R.B. Morgan, *GMRES with deflated restarting*. SIAM J. Sci. Comput., **24** (2002), 20–37.
- [18] N.M. Nachtigal, L. Reichel and L.N. Trefethen, *A hybrid GMRES algorithm for nonsymmetric linear systems*. SIAM J. Matrix Anal. Appl., **13** (1992), 796–825.
- [19] C.C. Paige and M.A. Saunders, *Algorithm 583 LSQR: Sparse linear equations and least squares problems*. ACM Trans. Math. Software, **8** (1982), 195–209.
- [20] C.C. Paige and M.A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*. ACM Trans. Math. Software, **8** (1982), 43–71.
- [21] M. Parks, E. Sturler, G. Mackey, D.D. Johnson and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*. SIAM J. Sci. Comput., **28** (2006), 1651–1674.
- [22] M. Pernice and H. Walker, *NITSOL: a Newton iterative solver for nonlinear systems*. SIAM J. Sci. Comput., **19** (1998), 302–318.
- [23] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*. SIAM J. Sci. Comput., **14** (1993), 461–469.
- [24] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Second Edition, SIAM, Philadelphia (2003).
- [25] Y. Saad and M.H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*. SIAM J. Sci. Comput., **7** (1986), 856–869.
- [26] W. Schonauer, *Scientific Computing on Vector Computers*. North-Holland, Amsterdam (1987).
- [27] V. Simoncini, *A new variant of restart GMRES(m)*. Numer. Linear Algebra Appl., **6** (1999), 61–77.
- [28] V. Simoncini, *On the convergence of restarted Krylov subspace methods*. SIAM J. Matrix Anal. Appl., **22** (2000), 430–452.

- [29] V. Simoncini and D.B. Szyld, *Theory of inexact Krylov subspace methods and applications to scientific computing*. SIAM J. Sci. Comput., **25** (2003), 454–477.
- [30] G. Starke and R.S. Varga, *A hybrid Arnoldi–Faber iterative method for nonsymmetric systems of linear equations*. Numer. Math., **64** (1993), 213–240.
- [31] E. de Sturler, *Truncation strategies for optimal Krylov subspace methods*. SIAM J. Numer. Anal., **36** (1999), 864–889.
- [32] J.V. Toledo-Benavides, *A globally convergent Newton-GMRES method with a new choice for the forcing term and some strategies to improve GMRES(m)*. PhD Thesis, T/UNICAMP T575m, (2005).
- [33] H.A. Van der Vorst and C. Vuik, *GMRESR: a family of nested GMRES methods*. Num. Linear Algebra Appl., **1** (1994), 369–386.
- [34] H.F. Walker and L. Zhou, *Residual smoothing techniques for iterative methods*. SIAM J. Sci. Comput., **15** (1994), 297–312.
- [35] I. Zavorin, *Spectral factorization of the Krylov matrix and convergence of GMRES*. UMIACS Technical Report CS-TR-4309, University of Maryland, College Park (2002).
- [36] I. Zavorin, D.P. O’Leary and H. Elman, *Complete stagnation of GMRES*. Lin. Algebra Appl., **367** (2003), 165–183.

Copyright of Computational & Applied Mathematics is the property of Computational & Applied Mathematics and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.