Some applications for Newton-Krylov methods with a safeguard for GMRES(m) [*]

Márcia A. Gomes-Ruggiero Véra L. Rocha Lopes
Julia V. Toledo-Benavides [†]

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 modified GMRES(m), GMRESH(m) procedure, alone and as part of an inexact Newton procedure with several choices for the forcing term, demonstrate the efficiency of the proposed strategy.

Keywords. linear systems, restarting GMRES, inexact Newton method, nonlinear systems.

AMS subject classifications: 65H10, 65F10.

1 Introduction

The objective of this work is to improve the performance of the restarted GMRES [22]. 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 [23], in the sense that the residual sequence does not converge to zero within a reasonable time frame. In [26] and [30] the authors modified the GMRES using spectral analysis. In [18], Ritz values are used to improve the performance of the restart. Convergence issues and stagnation are discussed in [28] and [35]. Van der Vorst and Vuik introduce an strategy to prevent stagnation in GMRES, by including LSQR steps in some phases of the process [32].

[*] Supported by FAPESP, CNPq, PRONEX-Optimization.

[†] 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 and juliatoledob@yahoo.com.br

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.

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. 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 is proposed in [22] for solving linear systems $As = b$, where A is a nonsingular $n \times n$ matrix (not necessarily symmetric) and $b \in I\!R^n$. 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:

$$\mathcal{K}_l = [r_0 \;, Ar_0 \;, A^2 r_0 \;, \ldots, A^{l-1} r_0]. \tag{1}$$

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 $\|.\|_2$ whenever we use $\|.\|$.

It is known that, computationally speaking, GMRES is more expensive than other Krylov subspace methods, such as Bi-CGSTAB, [15], QMR [23] 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.

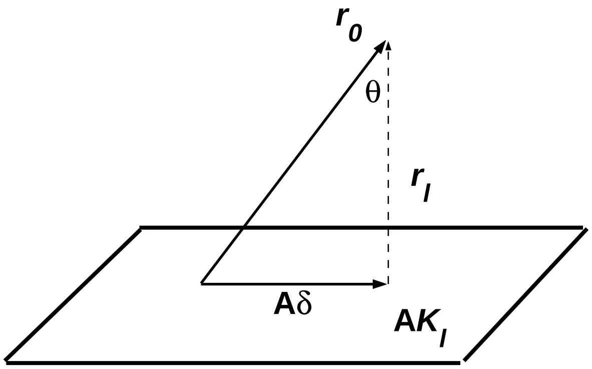

Figure 1: Orthogonal projection of r_0 in $A\mathcal{K}_l$

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 [15], [22]. 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], [23], [27], [30] and [35]. A rather expensive remedy would be to monitor the eigenvalues of the Hessenberg matrices generated during the GMRES, [27]. Other schemes, such as the one mentioned in [30], 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*, $\|r_l^j\|$, $l = 1, 2, 3, \ldots, m$. 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

$$\|r_m^j\|/\|r_0^j\| \sim 1 \tag{2}$$

and

$$|\cos(\theta_j)| \sim 1, \tag{3}$$

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_0^1 and r_m^j. Observe that r_m^j is the last residue of the j–th cycle whereas r_0^1 is the initial residue of the first cycle of the whole process. In this case, 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, too. 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 through the test between the cosine of the angle between the residues r_0^1 and r_m^j:

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

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

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

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, [12]. 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 [12] 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 $r = b - As = \alpha\bar{r} + (1-\alpha)\hat{r}$. Thus 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:

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

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^j 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 $\theta_j$ between $r_0^j$ and $r_m^j$:
    if $|\cos(\theta_j)| \sim 1$, compute $s_0^{j+1} = \alpha s_0^1 + (1-\alpha)s_m^j$, with $\alpha$ given by (6).
  Otherwise,
  Safeguard 2: test the angle $\theta_{j,1}$ between $r_m^j$ and $r_0^1$:
    if $|\cos(\theta_{j,1})| \sim 1$, then compute $s_0^{j+1} = \alpha s_0^1 + (1-\alpha)s_m^j$, with $\alpha$ given by (6).
if $j = 1$:
  Test the angle $\theta_1$ 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 $\alpha$ 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 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.

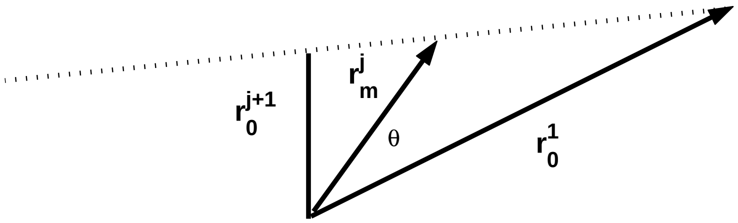

Figure 2: Safeguard 2: angle between r_0^1 and r_m^j, and residue obtained by the hybrid scheme.

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 [25], but in that case, it is used as a stopping criterion for the Conjugate Gradient method; in [33] 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 [24] and [32]. We do not use the term "hybrid GMRES" since in the literature it is sometimes used in other contexts, such as in [17], [29]. 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 calculated 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 cases, 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 [34] brings the system

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

and

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

with zero initial vector as an example where the sequence of GMRES(2) residues is constant (stagnation). The precision used was 0.0001.

The results of GMRESH(2) for the first eight cycles where the Strategy **H** is triggered are shown in Table 1: the sequences of relative residual norms and the cosine of the angle θ_j. The total number of iterations was 19, with last relative residue equal to 5.1417e-004 while the relative residue using GMRES(2) always was 1.

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 with equidistant 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.

Table 1: Cosine and relative residue between r_0^j and r_2^j

cycle	cosine	relative residue
1	1	1
2	0.9973	0.9932
3	0.9997	0.9353
4	0.9785	0.7836
5	0.9318	0.3964
6	0.8223	0.0796
7	0.9947	0.0107
8	0.7559	0.0104

Example 2. Embree [8] gives the matrix

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 2 \\ -4 \\ 1 \end{pmatrix} \tag{8}$$

as an example where GMRES(1) converges in three iterations but for GMRES(2) the relative residue stagnates near 0.3. We consider 1681 problems $As = b_{\nu\mu}$ where $r_0^1 = b_{\nu\mu} := (\nu, \mu, 1)^T$, $\nu, \mu \in [-10, 10]$ 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).

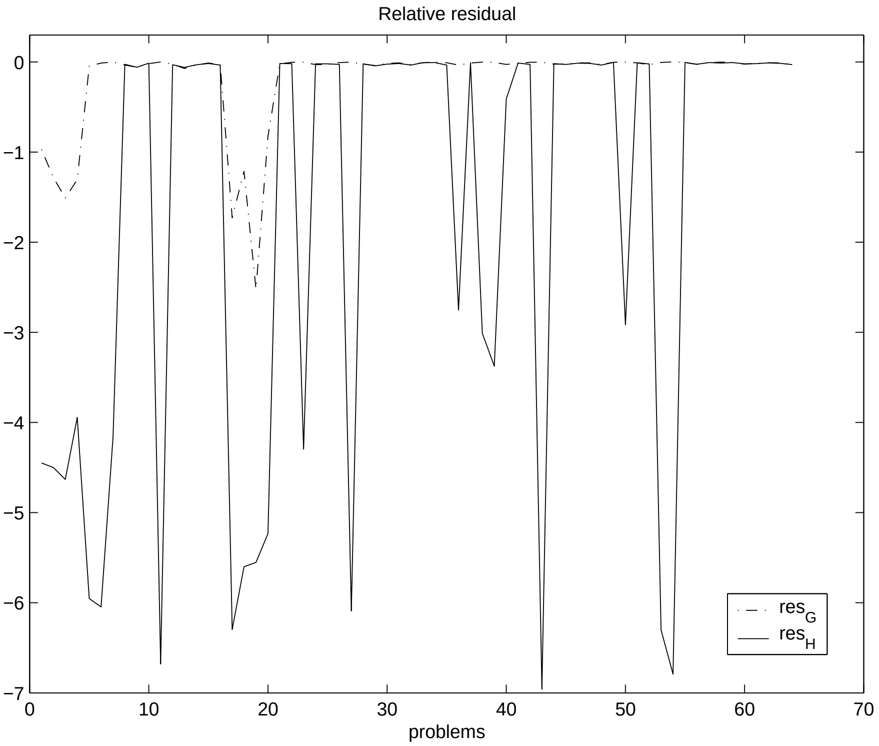

Figure 3: Logarithm of relative residual norms for the matrix (7). $\mathtt{res}_G$ and $\mathtt{res}_H$ represent relative residues obtained by GMRES(2) and GMRESH(2), respectively.

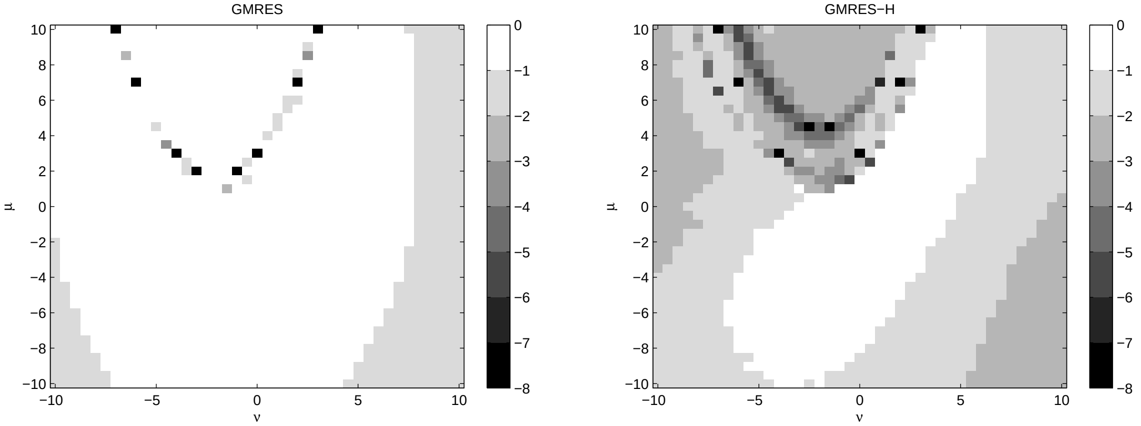

Figure 4: Logarithm of relative residual norms about matrix (8).

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

$$F(x) = 0, \quad F : I\!R^n \to I\!R^n.$$

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

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

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

$$\|J(x_k)s + F(x_k)\| \leq \eta_k \|F(x_k)\|, \tag{9}$$

where $J(.)$ 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 $k = 0, 1, 2, \ldots$ and $\sum_{k=0}^{\infty} \mu_k = \mu < \infty$.

Now we present the inexact Newton algorithm with the nonmonotone line search procedure. Let $x_0 \in I\!R^n$ be an arbitrary initial approximation to the solution for $F(x) = 0$. Given $x_k \in I\!R^n$, 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 $\eta_k$.
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`: we chose $\eta_k = 0.1$;

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

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

`GLT`: $\eta_k = [1/(k+1)]^{\rho} \cos^2(\phi_k) \dfrac{\|F(x_k)\|}{\|F(x_{k-1})\|}$, $\rho > 1$ 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 GMRES(m) and 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, [13]. 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 [31], pages 26 and 49. All the tests were performed in a Pentium III - 1.0GHz computer, using the software MatLab 6.1.

- Line search procedure:
  if the vector $x_{aux} = x_k + \xi s_k$ does not give an acceptable decrease in the value of the function, in the sense of Step 3 of Algorithm `1`, then we compute the new step size as $\xi_{new} = 0.5\xi$. For the parameter σ used in Step 3, we took $\sigma = 10^{-4}$.

- The sequence μ_k:
  we define:
  $ftip(0) = \|F(x_0)\|$,
  $ftip(k) = \min\{\|F(x_k)\|, ftip(k-1)\}$, if k is a multiple of 3 and
  $ftip(k) = ftip(k-1)$, otherwise.
  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 [21] 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 30 iterations, $m = 30$, allowing initially a maximum of 100 cycles (3000 iterations). This maximum number, called here by `maxit`, 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 $\|F(x_{new})\|/\|F(x_{old})\| > 100$, then `maxit` is fixed as 30; if $\|F(x_{new})\|/\|F(x_{old})\| > 1$ or if the maximum number of GMRES iterations has been exceeded, then `maxit` is fixed as 50.

- 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 [14] and also from a set of boundary value problems.

5.2 A ray-tracing problem, [2]

The objective here is to find the trajectory of an acoustic ray crossing a heterogeneous medium, via reflection/transmission, emitted at the point S (source) and registered at point G (geophone).

Without getting into details, the problem is reduced to solving the following nonlinear system: $\Phi(X) = 0$, where $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\Phi(X) = (\phi_1(X), \ldots, \phi_n(X))$, with $X_l = (x_l, f_{i_l}(x_l)))$ $l = 1, \ldots, n$ and $\phi_l : \mathbb{R}^n \rightarrow \mathbb{R}$, $l = 1, \ldots, n$, defined by:

$$\phi_l(X) = v_{i_{l+1}} \frac{(x_l - x_{l-1}) + f'_{i_l}(x_l)(f'_{i_l}(x_l) - f_{i_{l-1}}(x_{l-1}))}{[(x_l - x_{l-1})^2 + (f_{i_l}(x_l) - f_{i_l}(x_{l-1}))^2]^{1/2}}$$

$$- v_{i_l} \frac{(x_{l+1} - x_l) + f'_{i_l}(x_l)(f_{i_{l-1}}(x_{l+1}) - f_{i_l}(x_l))}{[(x_{l+1} - x_l)^2 + (f_{i_{l+1}}(x_{l+1}) - f_{i_l}(x_l))^2]^{1/2}} \; .$$

The functions $f_i(x)$ describe the interfaces of the problem. For a more complete treatment see [14] and [16]. The i–th signature $a_i,\ i = 1, ..., m-1$, of the ray, specifies the number of times it crosses the region Ω_i bounded between interfaces f_i and f_{i+1} in the downward direction. For uniqueness, we assume that all the downward crossings of Ω_i occur before those of Ω_{i+1}, as in Figure 5.

In Table 2 we consider a ray-tracing problem with two horizontal layers and signature 500, modeled as $\Phi(x) = 0$.

Table 2: Results for the ray-tracing problem

η_k	iterex	iterin	Feval
Cte	(7, 4)	(8800, 4978)	(8, 5)
EW1	(8, 5)	(6940, 3754)	(9, 6)
EW2	(7, 5)	(6724, 3424)	(8, 6)
GLT	(7, 5)	(6786, 3486)	(8, 6)

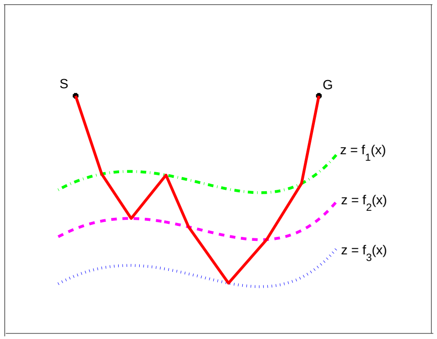

Figure 5: A ray-tracing model with two layers and signature $a = (2,\ 1)$.

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; finally, the column `Feval` represents the total number of function evaluations needed during the whole process. 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.

5.3 Boundary value problems

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

$$-\Delta u + h(\lambda,\ u) = f(s,t), \text{ in } \Omega, \quad u(s,t) = 0 \text{ on } \partial\Omega. \tag{10}$$

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 63 inner points in each axis. The discretized system obtained has 3969 equations and variables. 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 [3], page 105. In this case, $h(\lambda,\ u)$ is given by $h(\lambda,\ u) = \lambda u e^u$ 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 3: Results for the boundary value problems with $\lambda = 75$.

η_k	problem	iterex	iterin	feval
Cte.	Bratu	(8, 7)	(15809, 4959)	(9, 8)
	conv–dif	(12, 12)	(1729, 1829)	(27, 27)
	P3	(5, 5)	(189, 189)	(6, 6)
EW1	Bratu	(9, 7)	(18007, 5350)	(10, 8)
	conv–dif	(11, 11)	(1514, 1828)	(26, 26)
	P3	(5, 5)	(213, 217)	(6, 6)
EW2	Bratu	(8, 7)	(15302, 6409)	(9, 8)
	conv–dif	(11, 11)	(1517, 1826)	(26, 26)
	P3	(4, 4)	(198,182)	(5, 5)
GLT	Bratu	(8, 6)	(15452, 5497)	(9, 7)
	conv–dif	(11, 11)	(1518, 1824)	(26, 26)
	P3	(5, 5)	(227, 212)	(6, 6)

Table 3 shows the results of Newton-GMRES and Newton-GMRESH applied to the above problems, for $\lambda = 75$, chosen by the occurrence of stagnation in the 3 problems, and $x_0 = (0,\ 0, \ldots, 0)^T$. Each ordered pair should be read as in Table 2. iterex, iterin and feval are as before.

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 activated and `iterex` is slightly reduced.

The P3 problem can be considered an easy problem. Nevertheless, we noticed a slight improvement using GMRESH with the choices `EW2` and `GLT`. In the convection-diffusion problem, the Jacobian matrices are always ill-conditioned near the solution. The line search was activated various times, indicating stagnation coupled with insufficient decrease. For all the problems considered, `iterex` was practically the same (or slighly better) using GMRESH.

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 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. In the legends of Figure 6, Algorithm Newton-GMRESH is indicated by an `H`, after the name of the choice.

The measures used for comparison were the total number of inner iterations and the number of outer iterations. Considering the number of inner iterations, we observe that for the Algorithm Newton-GMRESH the choice `GLT` for η_k, obtained the best performance. This is easily seen because it solved about 58% of the problems with the smallest number of inner iterations, which is shown in the vertical axis. `EW2/H` had the second best performance, since it solved about 30% in the same situation. Moreover the performance of Newton-GMRES represented in Figure 6 by `EW` and `GLT` is under the one obtained by the algorithm Newton-GMRESH.

It can be observed that `EW` and `GLT` solved some problems with a number of inner iterations 5 and 6 times (respectively) greater than the minimum required to solve that problem. Thus, the version of the Algorithm without incorporating the strategies to prevent stagnation of the GMRES had a much worse performance than the new version Newton-GMRESH, where these strategies are incorporated in GMRES. The strategies worked successfully.

Analyzing now the performance with the number of outer iterations as measure, we observe again that the best performance was that of `GLT/H`, followed by `EW2/H`.

In both versions, GMRES and GMRESH, the safeguards were triggered only during the first external iterations. Subject to the usual precautions concerning the convergence of inexact Newton methods, and of the restarted GMRES method, the GMRESH version shows the same type of convergence rate, since this rate only depends on the sequence of forcing terms. In most of the examples checked, the GMRESH version had a better numerical performance.

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 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 very 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 2, as well as in the solution of a bunch of boundary value problems whose performance profile is showed in Figure 6.

References

[1] E. G. Birgin, N. Krejić, and J. M. Martínez, Globally convergent inexact quasi-Newton methods for solving nonlinear systems, *Numerical Algorithms* **32** (2003) 249–260.

[2] N. Bleistein, *Mathematical Methods for Wave Phenomena*, Academic Press, 1984.

[3] W. L. Briggs, V. E. Henson and S. F. McCormick *A Multigrid Tutorial*, Second Edition, SIAM, Philadelphia, 1987.

[4] P. N. Brown, A theoretical comparison of the Arnoldi and GMRES algorithms, *SIAM J. Sci. Stat. Comput.* **12** 1 (1991) 58–78.

[5] R. S. Dembo, S. C. Eisenstat and T. Steihaug, Inexact Newton methods, *SIAM J. Numer. Anal.* **19** 2 (1982) 401-408.

[6] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profile, *Mathematical Programming* 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** 1 (1996) 16–32.

[8] M. Embree, The tortoise and the hare restart GMRES, *SIAM Review* , **45** 2 (2003) 259–266.

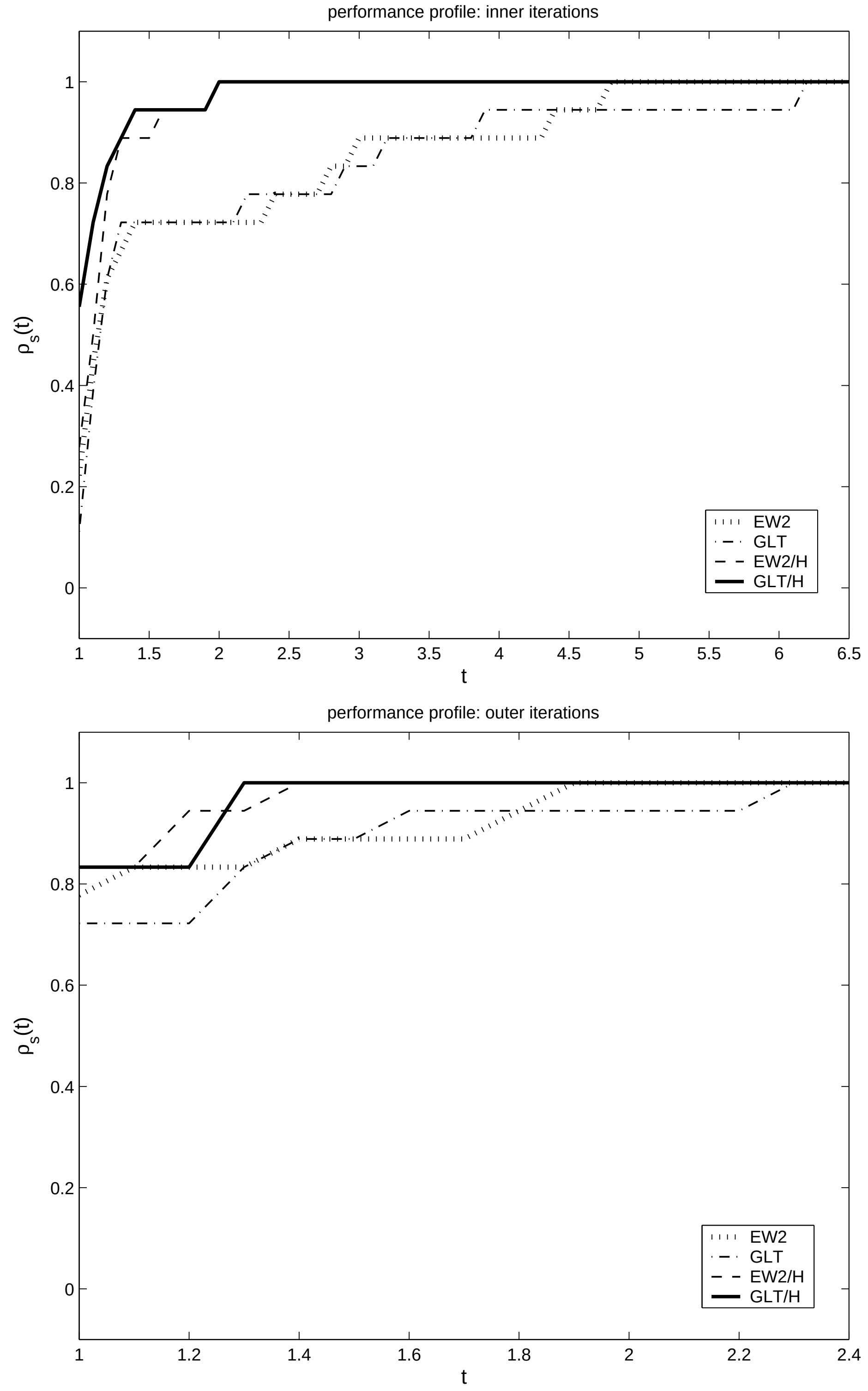

Figure 6: Performance Profile using as measures: inner iterations on the top and outer iterations on the bottom.

[9] M. A. Gomes–Ruggiero, D. N. Kozakevich and J. M. Martínez, A numerical study on large–scale nonlinear solvers, *Computers and Mathematics with Applications* **32** 3 (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, *to appear in Annals of Operations Research*

[11] W. Joubert, On the convergence behavior of the restarted GMRES algorithm for solving nonsymmetric linear systems, *Numerical Linear Algebra with Applications* **1** 5 (1994) 427-447.

[12] D. Brezinski and Redivo-Zaglia, Hybrid procedures for solving linear systems, *Numerische Mathematik* **67** (1994) 1–19.

[13] Li, D-H. and Fukushima, M., Derivative-free line search and global convergence of Broyden-like method for nonlinear equations, *Optimization Methods and Software* **13** (2000) 181-201.

[14] H. B. Keller and D. J. Perozzi, Fast seismic ray-tracing, *SIAM J. of Appl. Math.* **43** 4 (1983) 981–992.

[15] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, 1995.

[16] 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.

[17] N. M. Nachtigal, L. Reichel and L. N. Trefethen, A hybrid GMRES algorithm for nonsymetrical matrix itrations. *SIAM J. Matrix Anal. Appl.* **13** (1992) 796–825.

[18] Parks, Sturler, Mackey, Johnson, Maiti, Recycling Krylov subespaces for sequences of linear systems, *Technical Report UIUCDCS-R-2004-2421(CS), UILU-ENG-2004-1722 (ENGR), March 2004.*

[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. Pernice and H. Walker, NITSOL: a Newton iterative solver for nonlinear systems, *SIAM J. Sci. Comput.* **19** 1 (1998) 302–318.

[22] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Comput.* **7** 3 (1986) 856–869.

[23] Y. Saad, *Iterative Methods for Sparse Linear Sistems*, The PWS Publishing Company, 1996.

[24] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing* **14** 2 (1993) 461–469

[25] W. Schonauer, *Scientific computing on vector computer*, 1987.

[26] V. Simoncini, A new variant of restart GMRES(m), *Numer. Linear Algebra Appl.* **6** (1999) 61–77.

[27] V. Simoncini, On the Convergence of restarted Krylov subspace methods, *SIAM J. Matrix Anal. Appl.* **22** 4 (2000) 430–452.

[28] V. Simoncini and D. B. Szyld, Theory of inexact Krylov subspace methods and applications to scientific computing, *SIAM J. on Sci. Comput.* **25** (2003) 454–477.

[29] G. Starke and R. S. Varga, A hybrid Arnoldi-Faber iterative method for nonsymeric systems of linear equation, *Numer. Anal.* **36** (1999) 864–889.

[30] E. de Sturler, Truncation strategies for optimal Krylov subspace methods, *SIAM J. Numer. Anal.* **36** (1999) 864–889.

[31] 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.

[32] Van der Vorst, Vuik, GMRESR: a family of nested GMRES methods. *Numerical linear algrebra with applications.* **1** 4 (1994) 369–386.

[33] H. F. Walker, Zhou, Residual smoothing tecniques for iterative methods, *SIAM J. Sci. Comput.* **15** 2 (1994) 297–312.

[34] I. Zavorin, Spectral factorization of the Krylov matrix and convergence of GMRES, *UMIACS Technical Report CS-TR-4309, University of Maryland, College Park, January 2002.*

[35] I. Zavorin, D. P. O'Leary and H. Elman, Complete stagnation of GMRES, *Linear Algebra and its Applications*, **367** (2003) 165–183.