Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

RAINALD EHRIG PETER DEUFLHARD

GMERR – an Error Minimizing Variant of GMRES

Preprint SC 97-63 (Dezember 1997)

GMERR – an Error Minimizing Variant of GMRES

Rainald Ehrig Peter Deuflhard

ehrig@zib.de, deuflhard@zib.de

Abstract

The paper analyzes a recently proposed iterative error minimizing method for the solution of linear systems. Sufficient and necessary conditions for convergence are studied, which show that the method essentially requires normal matrices. An efficient implementation similar to GMRES has been worked out in detail. Numerical tests on general non–normal matrices, of course, indicate that this approach is not competitive with GMRES. Summarizing, if error minimizing is important, one should rather choose CGNE. A computational niche for GMERR might be problems, where normal but non–symmetric matrices occur, like dissipative quantum mechanics.

AMS Subject Classification: 65F10, 65F25, 65F50

Keywords: linear systems, Krylov subspace methods, error minimizing methods, preconditioning

Contents

1 Introduction

The generalized minimal residual algorithm (GMRES) of SAAD AND SCHULTZ [8] is a very popular method for solving large non–symmetric linear systems. Within this method, as in some other commonly used iterative algorithms, the progress of the iteration is controlled by the norm of the residuals, although in many applications the really interesting property is the error of the approximate solutions. Due to this reason WEISS [9] has proposed an error minimizing method, the general minimal error algorithm (GMERR). Until now no successful algorithmic realization of GMERR has been published and no detailed investigation of its advantages or disadvantages has been presented. In this note we analyze the convergence properties of GMERR and develop an effective algorithmic realization, which is very similar to the usual GMRES implementation. In passing, we show that the well known CGNR algorithm is not only a residual minimizing method, but also guarantees an iterative decrease of the error norm.

2 Derivation of the method

We consider a linear system $Ax = b$, where A is a large, usually sparse, nonsymmetric real matrix of size n and an initial guess $x_0 \in \mathbb{R}^n$. Projection methods generate approximate solutions x_k in a k–dimensional affine subspace $x_0 + \mathcal{K}_k$, for which the following Galerkin condition

$$r_0 - A(x_k - x_0) = b - Ax_k \perp \mathcal{L}_k$$

with $r_0 = b - Ax_0$ holds. $\mathcal{K}_k$ and $\mathcal{L}_k$ are Krylov subspaces of the general form

$$\mathcal{K}_k(v, A) = span\{v, Av, \ldots, A^{k-1}v\} \; .$$

Let V_k and W_k denote the basis of $\mathcal{K}_k$ resp. $\mathcal{L}_k$ in (n, k)–matrix notation. Then the Galerkin condition can equivalently be written as

$$W_k^T(r_0 - A(x_k - x_0)) = 0 \; . \tag{1}$$

For the approximate solutions one obtains

$$x_k = x_0 + V_k \left[W_k^T A V_k\right]^{-1} W_k^T \, r_0 \; . \tag{2}$$

Following SAAD [7] the usual Krylov subspace projection methods can be subdivided into three different classes:

1. $\mathcal{L} = \mathcal{K}$, orthogonal projection methods or Ritz–Galerkin approach. $\mathcal{K}_k = \mathcal{K}_k(r_0, A)$ defines the Full Orthogonalization Method (FOM), which for spd–matrices is equivalent to the conjugate gradient method.

2. $\mathcal{L} = A\mathcal{K}$, minimum residual approach. In this case (1) implies[1]

$$\begin{aligned} \|b - Ax_k\| &= \|r_0 - A(x_k - x_0)\| \\ &= \min_{x \in x_0 + \mathcal{K}_k} \|r_0 - A(x - x_0)\| = \min_{y \in \mathcal{L}_k} \|r_0 - y\| \,. \end{aligned} \tag{3}$$

   Hence the norm $\|b - Ax_k\|$ is minimal over $x_0 + \mathcal{K}_k$. The most important examples are GMRES with $\mathcal{K}_k = \mathcal{K}_k(r_0, A)$ and CGNR with $\mathcal{K}_k = \mathcal{K}_k(A^T r_0, A^T A)$.

3. $A^T\mathcal{L} = \mathcal{K}$, minimal error methods. From (1) and with x^* as the exact solution of the linear system we get

$$\begin{aligned} W_k^T(b - Ax_k) &= W_k^T A A^{-1}(b - Ax_k) \\ &= (A^T W_k)^T(x^* - x_k) = V_k^T(x^* - x_k) = 0 \end{aligned} \tag{4}$$

   and hence

$$\|x^* - x_k\| = \min_{x \in x_0 + \mathcal{K}_k} \|x^* - x\| \,,$$

   which proves the error minimizing property.

The present paper focusses on this third class of iterative methods.

Remark. It is important to remark that the relation between the Krylov subspaces is sufficient for the residual resp. error minimizing properties, but not necessary. Indeed the CGNR method, which does not satisfy $A^T\mathcal{L} = \mathcal{K}$, is error reducing – as proven in Appendix 1. In recent times there is a renewed interest in the appropriately preconditioned CGNR method, see for example Benzi and Tuma [1], which has been underrated for many years due to the "squaring the condition number argument", see e.g. Nachtigal et al. [6]. Perhaps the somewhat surprising fact that CGNR simultaneously reduces the residuals *and* errors of the approximate solutions may motivate further research on robust implementations.

Until now the only commonly known genuine error–minimizing method is CGNE or Craig's method defined by $\mathcal{K}_k = \mathcal{K}_k(A^T r_0, A^T A)$, which can

[1] Throughout this paper $\|\cdot\|$ is the 2–norm and $\langle \cdot, \cdot \rangle$ the Euclidean inner product.

be derived applying the CG algorithm to the normal equations in the form $AA^T y = b$, $x = A^T y$.
A second algorithm, named GMERR by Weiss [9] in analogy to GMRES, can be obtained through the specifications

$$\begin{aligned} \mathcal{K}_k &= span\{A^T r_0, (A^T)^2 r_0, \ldots, (A^T)^k r_0\} , \\ \mathcal{L}_k &= span\{r_0, A^T r_0, \ldots, (A^T)^{k-1} r_0\} . \end{aligned}$$

GMERR and CGNE share some common properties. First, from the Galerkin condition (4), one may derive

$$\langle x_i - x_0, x^* - x_k \rangle = 0, \ i \leq k$$

and therefore $\langle x_i - x_0, x^* \rangle = \langle x_i - x_0, x_k \rangle$ for $i \leq k$. Since for the errors e_i hold $\|e_k\| \leq \|e_i\|$, if $i \leq k$, one obtains

$$0 \leq \langle x_i - x_0, x_i - x_0 \rangle \leq \langle x_k - x_0, x_k - x_0 \rangle \leq \langle x^* - x_0, x^* - x_0 \rangle, \ i \leq k ,$$

which, with $\delta x_k := x_{k+1} - x_k$, implies

$$\begin{aligned} \langle \delta x_i, \delta x_k \rangle = 0, \ i < k , \\ \langle \delta x_k, \delta x_k \rangle = \langle x_{k+1} - x_0, x_{k+1} - x_0 \rangle - \langle x_k - x_0, x_k - x_0 \rangle . \end{aligned}$$

Therefore the iterative errors satisfy

$$\begin{aligned} \|e_k\|^2 &= \|e_0\|^2 - \sum_{i=0}^{k-1} \|\delta x_i\|^2 , \\ \|e_k\|^2 &= \|e_{k-1}\|^2 - \|\delta x_{k-1}\|^2 . \end{aligned} \tag{5}$$

These relations will be helpful for a control of the convergence of the GMERR algorithm.

3 Convergence analysis of GMERR

In this section, the theoretical properties of GMERR and GMRES are synoptically compared, independent of any algorithmic implementation. First, we analyze the method without considering restarts. For simplicity we set $x_0 = 0$ throughout this section. Let $q(A)$ be the minimal polynomial of A defined as the unique monic polynomial of minimal degree with $q(A) = 0$. Herewith one easily derives the following characterization of the convergence of GMRES.

Lemma 1. *GMRES converges in at most k steps to the exact solution for every right–hand side b, if and only if the minimal polynomial of A is of degree k.*

Proof. If GMRES converges for every b in at most k iterations, then for every b the minimal polynomials $q_b(t)$ with the property $q_b(A)b = 0$ are of degree less or equal k. Since then the minimal polynomial of A is identical to one of the q_b's, see HOUSEHOLDER [4, p. 18], it has the desired property. The converse follows from $\|r_k\| = \min_{p\in P_k,\, p(0)=1} \|p(A)b\|$. □

Due to the Cayley–Hamilton theorem the degree of the minimal polynomial is always less or equal n, thus GMRES converges always in at most n steps. Next we try to derive the corresponding property for GMERR. Since the Krylov subspaces are generated by repeated applications of A^T, we obtain for the approximate solutions

$$\|e_k\| = \|x^* - x_k\| = \min_{p\in P_k,\, p(0)=0} \|x^* - p(A^T)b\| \tag{6}$$

and accordingly to Lemma 1 the following statement.

Lemma 2. *GMERR converges in at most k steps to the exact solution for every right–hand side b, if and only if it exists a polynomial q of degree k with $q(A^T) = A^{-1}$ and $q(0) = 0$.*

Proof. If GMERR converges for every b in at most k iterations, then for every b we have a polynomial $q_b(t)$ of degree less or equal k with $q_b(A^T)b = A^{-1}b$ and $q_b(0) = 0$. For an eigenvector v of A^T with eigenvalue λ we obtain $q_v(\lambda)v = q_v(A^T)v = A^{-1}v$. Thus each eigenvector of A^T is also an eigenvector of A^{-1} resp. A. This is equivalent to the normality of A, see e.g. SAAD [7, p. 22]. Let $v_1, \ldots, v_n$ now be the orthonormal set of eigenvectors of A^T. Then with $b = \sum_{i=1}^n v_i$ one obtains

$$\sum_{i=1}^{n} q_b(\lambda_i)v_i = q_b(A^T)b = A^{-1}b = \sum_{i=1}^{n} \bar{\lambda}_i^{-1} v_i \ .$$

Therefore we have $q_b(A^T)e_i = A^{-1}e_i$ with the unit vectors e_i and hence q_b is a polynomial with $q_b(A^T) = A^{-1}$ and $q_b(0) = 0$. The converse follows similar as in Lemma 1 from (15). □

As a very important consequence of this Lemma we state

Theorem 3. *GMERR without restarts converges for every right–hand side, if and only if A is a normal matrix.*

Proof. Due to Lemma 1 it remains to show that for a normal matrix A exists a polynomial q with $q(A^T) = A^{-1}$ and $q(0) = 0$. Let $A = U\Lambda U^*$ the unitary diagonalization of A. Using Lagrange interpolation one can construct a real polynomial q of degree less or equal n with $q(\bar{\Lambda}) = \Lambda^{-1}$ and $q(0) = 0$. Then we have

$$q(A^T) = \bar{U}q(\Lambda)U^T = Uq(\bar{\Lambda})U^* = U\Lambda^{-1}U^* = A^{-1} , \tag{7}$$

thus q has the desired properties. □

At a first glance, this theorem states a serious limitation of the applicability of GMERR, but we will see that GMERR with restarts converges for a larger class of matrices. Considering the Schur decomposition $U^*AU = \Lambda + N$ of an arbitrary matrix A one can interpret the strictly upper diagonal matrix N as the "non-normal" part of A. Indeed $(\Lambda + N)^{-1}$ can not be represented as a polynomial in $(\Lambda + N)^T$, since the inverse of an upper triangular matrix is also upper triangular, this proofs again one direction of Theorem 1.
Now we discuss more practical conditions, which guarantee convergence of GMRES resp. GMERR in at most k, usually $k \ll n$, steps. These considerations are important even for the study of suitable preconditioning techniques. We begin again for iterative methods with with $x_k \in \mathcal{K}(r_0, A)$. The following lemma states a sufficient condition for the convergence of GMRES for every right–hand side in at most $k + 1$ iterations.

Lemma 4. *GMRES converges for every right–hand side b in at most $k + 1$ steps if $I - A$ has rank k.*

Proof. If $I - A$ has rank k, A is a k–rank modification of the identity. Thus we can write $A = I + VW^T$ with $V, W \in \mathbb{R}^{n\times k}$. Now the Sherman–Morrison–Woodbury formula gives

$$(I + VW^T)^{-1} = I - V(I + W^TV)^{-1}W^T . \tag{8}$$

Since $I + W^TV \in \mathbb{R}^{k\times k}$ it exists a polynomial q_1 of degree at most $k-1$ with $q_1(I + W^TV) = (I + W^TV)^{-1}$. Inserting this in (8) one obtains

$$\begin{aligned}
A^{-1} &= I - Vq_1(I + W^TV)W^T \\
&= I - Vq_2(W^TV)W^T , && q_2 \text{ of degree } k-1 \\
&= I - q_3(VW^T) , && q_3 \text{ of degree } k ,\ q_3(0) = 0 \\
&= q_4(I + VW^T) , && q_4 \text{ of degree } k \\
&= q_4(A) ,
\end{aligned}$$

thus the minimal polynomial of A is of degree at most $k + 1$. Therefore GMRES converges always in at most $k + 1$ steps. □

An alternative proof is based on the fact that the minimal polynomial of a matrix with rank $k < n$ is of a degree at most $k+1$. Hence we have $q(I-A) = 0$ for a polynomial q of degree less or equal $k+1$, and obviously even the minimal polynomial of A is of degree at most $k+1$.
The corresponding statement for GMERR bases upon the normality of A.

Lemma 5. *GMERR converges for every right–hand side b in at most $k+1$ steps if $I-A$ has rank k and A is normal.*

Proof. Since $I-A$ has rank k we can write $A = U\Lambda U^*$ with U unitary, $\Lambda = diag(1,\dots,1,\lambda_1,\dots,\lambda_k)$ and $\lambda_i \neq 1, i = 1,\dots,k$. As in Theorem 1 we can construct a polynomial q of degree $k+1$ with $q(\bar{\lambda_i}) = \lambda_i^{-1}$, $q(1) = 1$ and $q(0) = 0$. Now again equation (7) holds and hence due to Lemma 2 GMERR converges always in at most $k+1$ iterations. □

With similar arguments one obviously proofs likewise the convergence of the GMERR algorithm, if the normal matrix A has not more than $k+1$ different eigenvalues. For GMRES such a generalization holds only for diagonalizable matrices.
Both Lemma 3 and 4 are even useful to estimate the convergence of left or right preconditioned GMRES resp. GMERR iterations. With P as a preconditioner $P-A$ has rank k, if and only if $P^{-1}A$ or AP^{-1} is a k–rank modification of the identity and we can conclude the following statements.

Lemma 6. *The preconditioned GMRES algorithm converges for every right–hand side b in at most k steps if $P-A$ has rank k.*

Lemma 7. *The preconditioned GMERR algorithm converges for every right–hand side b in at most k steps if $P-A$ has rank k and $P^{-1}A$ resp. AP^{-1} is normal.*

These lemmas explain, why some modern and successful approaches for the construction of preconditioners are obtained as low–rank modifications of the matrix A, see for example Bramley and Meñkov [2].
In order to substantiate the importance of these properties, we give an example involving block–Jacobi preconditioning, which is the basis for many parallel preconditioning schemes, see e.g. Ehrig, Nowak and Deuflhard [3]. Usually large sparse matrices are related to some grid, which lead to banded systems. Now we analyze the structure of the left preconditioned matrix, i.e. of the matrix product $P_{Jac}^{-1}A$, see Fig. 1, using exact subblock solutions. In Fig. 1 the block–column matrices E_i and F_i are defined by $E_i = A_i^{-1}B_i$ and $F_i = A_i^{-1}C_i$, the diagonal elements of $P_{Jac}^{-1}A$ are equal 1. We assume for

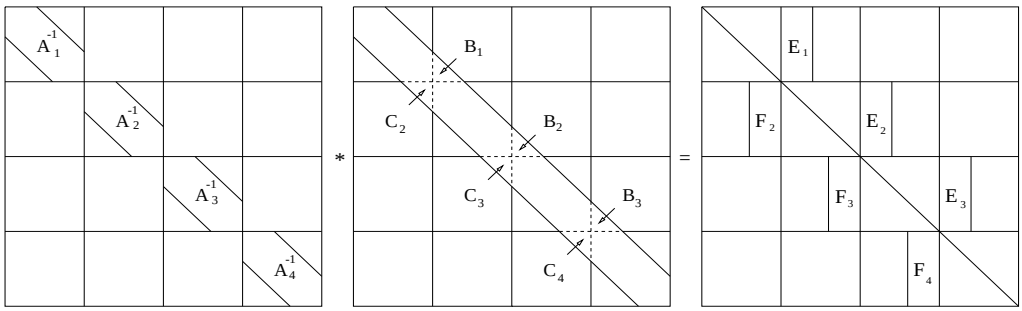

Figure 1: Schematic representation of the matrix product $P_{Jac}^{-1}A$ with 4 subblocks

clarity that the dimension n of A is a multiple of p and A has m lower and upper diagonals. Then each of the matrices E_i and F_i covers m vectors.
Now Lemma 5 tells us that preconditioned GMRES ends with the exact solution after at most $2m(p-1)$ iterations, since $P_{Jac}^{-1}A$ is a $2m(p-1)$–rank update of the identity matrix. The corresponding preconditioned GMERR algorithm would converge with the same number of iterations, if and only if $P^{-1}A$ would be normal, which is not necessarily true, even if A is symmetric.
The convergence of GMRES can furthermore be analyzed in terms of the eigenvalue distribution if A is diagonalizable, i.e. $A = X\Lambda X^{-1}$. Then the following result holds, Saad [7, p. 195],

$$\|r_k\| \leq \|r_0\| \, \|X\| \, \|X^{-1}\| \min_{p \in P_k, p(0)=1} \max_{\lambda \in \sigma(A)} |p(\lambda)| \, .$$

For GMERR and normal matrices A one can easily derive the relation

$$\|e_k\| \leq \|e_0\| \min_{p \in P_k, \, p(0)=1, \, p'(0)=0} \max_{\lambda \in \sigma(A)} |p(\lambda)| \, ,$$

which suggests fast convergence for clustered eigenvalues and a similar convergence behavior of GMRES and GMERR for normal matrices.
The convergence of the restarted GMERR algorithm is rather intricate to analyze. The involved polynomials consist of mixed terms in A and A^T. The optimal restarted GMERR method should select this polynomial, which is of minimal degree in A resp. A^T, but this is very difficult to achieve. Note, however, that for the restarted variants of GMRES the solutions are in the same Krylov subspace as for the exact variant, whereas GMERR begins with a new Krylov subspace at each restart. Thus the restarted GMRES converges always slower than full GMRES, assuming exact arithmetics. In contrast to this the convergence of restarted GMERR may be faster compared to the full variant. In general we can expect fast convergence for matrices with

an essential normal or orthogonal component or clustered eigenvalues. Any appropriate preconditioner therefore should "move" the iteration matrix in these directions.
Preconditioners can be applied from the left or/and the right. With right preconditioning, GMERR minimizes the preconditioned errors. So if one is interested in minimizing the true error norms, one should use left preconditioners, which only affect the rate of convergence.

4 Algorithmic realization

The most successful implementation of GMRES is based on a Gram–Schmidt orthogonalization. As in turns out, a similar technique helps in the implementation of GMERR. To introduce our approach we refer to some algorithmic details of GMRES.
From (2) and $AV_k = W_k$ GMRES constructs approximate solutions as

$$x_k = x_0 + V_k \left[(AV_k)^T (AV_k)\right]^{-1} (AV_k)^T r_0 \ . \tag{9}$$

With the orthonormal basis V_k of $\mathcal{K}_k$ then one defines the $(k+1,k)$–Hessenberg matrices $\bar{H}_k$ through $AV_k = V_{k+1}\bar{H}_k$. Then we obtain from (9) with $\beta = r_0$ and the unit vector $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^{k+1}$

$$x_k = x_0 + V_k \left[\bar{H}_k^T \bar{H}_k\right]^{-1} \bar{H}_k^T \beta e_1 =: x_0 + V_k y_k \ , \tag{10}$$

where $y_k \in \mathbb{R}^k$ minimizes the *overdetermined* linear system $\bar{H}_k y_k = \beta e_1$. Thus we have $y_k = \bar{H}_k^+ \beta e_1$, $\bar{H}_k^+$ being the pseudoinverse von $\bar{H}_k$.
Proceeding equivalently for GMERR we obtain from (2) and $V_k = A^T W_k$

$$x_k = x_0 + A^T W_k \left[(A^T W_k)^T (A^T W_k)\right]^{-1} W_k^T r_0 \ .$$

Now we have to use an orthonormal basis W_k of $\mathcal{L}_k$ to define the Hessenberg matrices by $A^T W_k = W_{k+1}\bar{H}_k$ and we obtain correspondingly to (10) with $e_1 \in \mathbb{R}^k$

$$x_k = x_0 + W_{k+1}\bar{H}_k \left[\bar{H}_k^T \bar{H}_k\right]^{-1} \beta e_1 \ .$$

With $y_k := \bar{H}_k \left[\bar{H}_k^T \bar{H}_k\right]^{-1} \beta e_1$ we can write $\bar{H}_k^T y_k = \beta e_1$, therefore $y_k \in \mathbb{R}^{k+1}$ now is the norm–minimal solution of the *underdetermined* linear system

$$\bar{H}_k^T y_k = \beta e_1 \ , \tag{11}$$

i.e. $y_k = (\bar{H}_k^T)^+ \beta e_1$.

In GMRES the overdetermined least squares problems are solved by successive factorization

$$\bar{H}_k = Q_k \begin{bmatrix} R_k \\ 0 \end{bmatrix}$$

with $Q_k \in \mathbf{O}(k+1)$, $R_k \in \mathbb{R}^{k\times k}$ upper triangular. The optimal solution y_k of the system $\bar{H}_k y_k = \beta e_1$ is then given by

$$y_k = R_k^{-1} b_1 \quad \text{with} \quad Q_k^T \beta e_1 = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} .$$

Within GMERR we can use the same QR decomposition as for GMRES to solve the underdetermined least squares problem (11). Its minimal solution is

$$y_k = Q_k \begin{pmatrix} R_k^{-T} \beta e_1 \\ 0 \end{pmatrix} .$$

The factorization of the $\bar{H}_k$ can be done successively just as in GMRES, see [7], so that only the actual column of $\bar{H}_k$ needs to be stored. The progressive decomposition yields a new column of the upper triangular matrices R_k per each iteration or equivalently a new row of the lower triangular matrices R_k^T. Now assume we have already solved $R_k^T z_k = e_1$. Then in the next iteration we need the solution of $R_{k+1}^T z_{k+1} = e_1$, which can be written as

$$\begin{pmatrix} & R_k^T & & \\ & & & \\ r_{1,k+1} & \dots & r_{k,k+1} & r_{k+1,k+1} \end{pmatrix} z_{k+1} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Obviously, the first k components of the vector $z_{k+1} \in \mathbb{R}^{k+1}$ are identical to $z_k \in \mathbb{R}^k$ and for the last component we obtain

$$(z_{k+1})_{k+1} = -\sum_{i=1}^{k} r_{i,k+1} (z_k)_i / r_{k+1,k+1} .$$

Therefore we do not need to store the triangular matrices R_k and the computation of the minimal solutions $R_k^{-T} \beta e_1$ can be done successively as well.

The algorithmic realization of GMERR can now be sketched as follows.

1. $r_0 = b - Ax_0,\ \beta = \|r_0\|,\ w_1 = r_0/\beta.$
2. For $j = 1, \ldots, k$
3. $\quad w_{j+1} = A^T w_j$
4. $\quad$ For $i = 1, \ldots, j$: $h_{ij} = \langle w_{j+1}, w_i \rangle$, $w_{j+1} = w_{j+1} - h_{ij} w_j$
5. $\quad h_{j+1,j} = \|w_{j+1}\|$, $w_{j+1} = w_{j+1}/h_{j+1,j}$
6. $\quad$ Extend the QR decomposition of $\bar{H}_{j-1}$ to those of $\bar{H}_j$
7. $\quad$ Compute $R_j^{-T} \beta e_1$ using $R_{j-1}^{-T} \beta e_1$
8. $y_k = Q_k \binom{R_k^{-T} \beta e_1}{0}$
9. $x_k = x_0 + W_{k+1} y_k$

We remark that step 8 and 9 ar only needed to compute the approximations themselves, but not for further iterations. So, if one is sure to need more iterations, these steps can be omitted.
In Appendix 2 we have included a detailed derivation of the algorithmic realization together with a pseudocode presentation.

5 Convergence control of GMERR

It is well known that the convergence of the GMRES iteration can be cheaply monitored by means of

$$\|r_k\| = \|r_{k-1}\| s_k = \beta \prod_{i=1}^{k} s_i \ ,$$

with s_i the Givens coefficients of the QR–factorization. Since GMERR is error minimizing, we would like to have a similarly cheap error monitor at hand. Unfortunately we could not find a comparably cheap method to estimate the errors $\|x^* - x_k\|$. The most promising approach to control the progress of the iteration seems to be exploiting the contributions δx_k to the error e_0 as suggested by equation (5). This enables a definite detection of a stagnation of the iterative process. The computation of the terms $\|\delta x_{k-1}\|$ is not expensive, as can be shown by the following equations

$$\begin{aligned} \|\delta x_{k-1}\| &= \|x_k - x_{k-1}\| = \|W_{k+1} y_k - W_k y_{k-1}\| \\ &= \left\| W_{k+1} y_k - W_{k+1} \begin{pmatrix} y_{k-1} \\ 0 \end{pmatrix} \right\| = \left\| y_k - \begin{pmatrix} y_{k-1} \\ 0 \end{pmatrix} \right\| \end{aligned}$$

Obviously, for the computation of $\|\delta x_{k-1}\|$ we need the minimal solutions of the underdetermined systems $\bar{H}_k^T y_k = \beta e_1$, but not the approximate solutions x_k themselves. The evaluation of y_k requires only about $4k$ additional floating point operations.
In the practical implementation we will compute the relative error contributions $\|\delta x_k\|/\|\delta x_1\|$. If this term is smaller than a predefined minimal threshold δ_{min}, then the convergence is regarded as "too slow" and the iteration is restarted. To inhibit numerical instabilities due to Krylov subspaces of large dimensions, we define furthermore (as in the usual GMRES realizations) a maximal dimension k_{max} of the Krylov subspace. Thus our implementation depends on the two parameter δ_{min} and k_{max}. Appropriate settings for both parameters will be discussed in the next section.

6 Numerical tests

In this section some results are presented to show the efficiency of the GMERR approach compared to GMRES. We took a large number of matrices from the Matrix Market collection [5]. As preconditioner we used an incomplete LU–factorization with an appropriate selected fill–in and threshold parameters, see SAAD [7]. In each case, a bunch of tests with varying parameters k_{max} and δ_{min} were done to find "optimal" values. Since all results in general are very similar, we only show the "best" results, which exhibit all typical phenomena. We selected the matrix SHERMAN5, which is well known in the numerical analysis community and arises from oil reservoir modeling. It is

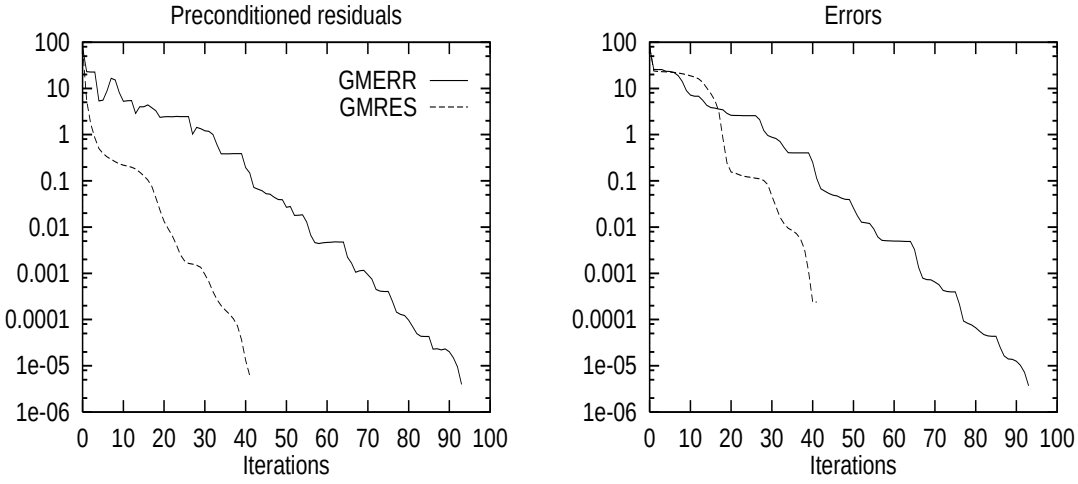

Figure 2: Convergence of GMERR and GMRES for the matrix SHERMAN5.

a real unsymmetric matrix of size $n = 3312$ with 20793 nonzero entries. As right–hand side we used a random vector. Rows and columns of the matrix were scaled by their 1–norms.
Fig. 2 compares the convergence behavior of left preconditioned GMERR and GMRES(20). The ILU preconditioner has a fill–in parameter 10 and a threshold for the drop tolerance of 10^{-4}. k_{max} was set to 15 and δ_{min} to 0.01. The results clearly show the superiority of GMRES, even in this nearly optimal example.
Nevertheless several features are to be mentioned. First, GMERR reduces the (preconditioned) residuals and errors in a very similar fashion, whereas GMRES sometimes reaches a satisfactory residual norm with an error significantly larger than the final error of GMERR. Next we demonstrate, how the progress of the GMERR iteration can be controlled via δ_{min}. Fig. 3 shows the size of $\|\delta x_k\|/\|\delta x_1\|$ during the iteration. This figure demonstrates that

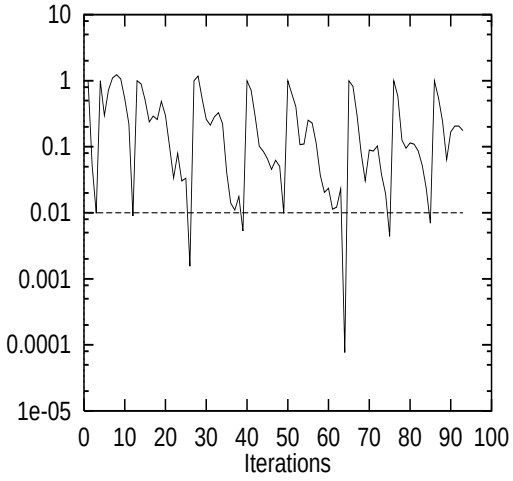

Figure 3: $\|\delta x_k\|/\|\delta x_1\|$ during the GMERR iteration with $\delta_{min} = 0.01$. (Note that $\|\delta x_1\|$ is redefined whenever $\|\delta x_k\|/\|\delta x_1\| < 0.001$)

whenever the GMERR iteration stagnates, the control of $\|\delta x_k\|$ enables an effective technique for timely restarts. The dimension of the Krylov spaces built by GMERR is in most cases small. In most of our test cases the optimal value of k_{max} is between 5 and 15, which implies that many restarts are needed. Also the optimal δ_{min} was not very different within our test set, optimal values were always found between 0.1 and 0.001, but we could not recognize any systematic trend in these parameters.

References

[1] M. Benzi, M. Tuma: *A Comparison of Some Preconditioning Techniques for General Sparse Matrices.* in: Iterative Methods in Linear Algebra, II, S.D. Margenov, P.S. Vassilevski, eds., IMACS Series in Computational and Applied Mathematics, **3**, pp. 191–203 (1996).

[2] R. Bramley, V. Meñkov: *Low Rank Off–Diagonal Block Preconditioners for Solving Sparse Linear Systems on Parallel Computers.* Tech. Rep. 446, Department of Computer Science, Indiana University, Bloomington (1996).

[3] R. Ehrig; U. Nowak; P. Deuflhard: *Massively Parallel Linearly–Implicit Extrapolation Algorithms as a Powerful Tool in Process Simulation.* Preprint SC 97–43 Konrad–Zuse–Zentrum Berlin 1997, accepted for publication in Advances in Parallel Computing.

[4] A.S. Householder: *The Theory of Matrices in Numerical Analysis.* Blaisdell Publishing Co.: New York, Toronto, London (1964).

[5] *Matrix Market, a visual repository of test data for use in comparative studies of algorithms for numerical linear algebra.*
URL: http://math.nist.gov/MatrixMarket/index.html

[6] N.M. Nachtigal, S.C. Reddy, L.N. Trefethen: *How fast are Nonsymmetric Matrix Iterations?* SIAM J. Matrix Anal. Appl. **13**, pp. 778–795 (1992).

[7] Y. Saad: *Iterative Methods for Sparse Linear Systems.* PWS Publishing Co.: Boston (1996).

[8] Y. Saad, M.H. Schultz: *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems.*
SIAM J. Sci. Stat. Comp. **7**, pp. 856–869 (1986).

[9] R. Weiss: *Error–Minimizing Krylov Subspace Methods.*
SIAM J. Sci. Comp. **15**, pp. 511–527 (1994)

Appendix 1: Iterative error reduction by CGNR

Here we prove the error–reducing property of CGNR, which seems to be not generally known.

Theorem 8. *CGNR is both residual minimizing and error reducing.*

Proof. For clarity we assume $x_0 = 0$. Then we have for CGNR

$$\begin{aligned} \mathcal{K}_k &= span\{A^T b, (A^T A)A^T b, \ldots, (A^T A)^{k-1} A^T b\} \\ \mathcal{L}_k &= span\{AA^T b, (AA^T)^2 b, \ldots, (AA^T)^k b\} \ . \end{aligned}$$

Thus by $\mathcal{L}_k = A\mathcal{K}_k$ and (3) CGNR is identified as a residual minimizing method, i.e. $\|r_k\| \leq \|r_i\|$, $i \leq k$. Furthermore follows from the Galerkin condition $\langle Ax_i, b - Ax_k\rangle = 0$. This gives together for $i \leq k$

$$0 \leq \langle Ax_i, b\rangle \leq \langle Ax_k, b\rangle \leq \langle b, b\rangle \ . \tag{12}$$

The condition $A^T\mathcal{L}_k \subseteq \mathcal{K}_{k+1}$ suggests the following rewriting of the Galerkin condition (1)

$$W_{k-1}^T \left(b - \langle x_k, v_1\rangle Av_1 - A(x_k - \langle x_k, v_1\rangle v_1)\right) = 0, \ k > 1 \ ,$$

with v_1 the first Krylov vector in $\mathcal{K}$. This equation can be converted to

$$(A^T W_{k-1})^T \left(x^* - \langle x_k, v_1\rangle v_1 - (x_k - \langle x_k, v_1\rangle v_1)\right) = 0 \ .$$

Since $x_k - \langle x_k, v_1\rangle v_1 \in A^T W_{k-1}$, the last equation is equivalent to

$$\|x^* - \langle x_k, v_1\rangle v_1 - (x_k - \langle x_k, v_1\rangle v_1)\| = \min_{y \in A^T W_{k-1}} \|x^* - \langle x_k, v_1\rangle v_1 - y\| \ .$$

Therefore we have with $v_1 = A^T b/\|A^T b\|$

$$\|e_k\| = \|x^* - x_k\| = \min_{y \in A^T W_{k-1}} \|x^* - \langle Ax_k, b\rangle \frac{A^T b}{\langle A^T b, A^T b\rangle} - y\|$$

Obviously the minimum is $y_k = x_k - \langle b, Ax_k\rangle A^T b/\langle A^T b, A^T b\rangle$. In order to prove now $\|e_{k+1}\| \leq \|e_k\|$, we have to construct $y \in A^T W_k$ with

$$\|x^* - \langle Ax_{k+1}, b\rangle \frac{A^T b}{\langle A^T b, A^T b\rangle} - y\| \leq \|x^* - x_k\| \ .$$

As the simplest choice we set $y = y_k$. Then we have to show

$$\|x^* - x_k - (\langle Ax_{k+1}, b\rangle - \langle Ax_k, b\rangle)\frac{A^T b}{\langle A^T b, A^T b\rangle}\| \leq \|x^* - x_k\| \ ,$$

which is equivalent to

$$\left(\langle Ax_{k+1}, b\rangle - \langle Ax_k, b\rangle\right)^2 \leq 2\left(\langle Ax_{k+1}, b\rangle - \langle Ax_k, b\rangle\right)\langle x^* - x_k, A^T b\rangle \ .$$

With the inequality (12) it remains to show that

$$\left(\langle Ax_{k+1}, b\rangle - \langle Ax_k, b\rangle\right) \leq 2\left(\langle b, b\rangle - \langle b, Ax_k\rangle\right)$$

holds. But this condition is, again by help of (12), always fulfilled.
It remains to show $\|e_1\| \leq \|e_0\|$. For CGNR we have

$$x_1 = \langle A^T b, A^T b\rangle / \langle AA^T b, AA^T b\rangle A^T b \ .$$

We have to prove $\|x^* - x_1\| \leq \|x^*\|$ or equivalently

$$\langle A^T b, A^T b\rangle^2 \leq 2\langle b, b\rangle\langle AA^T b, AA^T b\rangle \ .$$

which follows immediately from the Schwarz inequality. □

Appendix 2: Implementation details of GMERR

Here we give a detailed derivation of the GMERR implementation. In each iteration we have to compute firstly $A^T w_j$ and then via Gram–Schmidt orthogonalization a new column of the Hessenberg matrix $\bar{H}_k$. Now we show in detail how the factorization and computation of the minimal solution can be carried out.
In the first iteration we obtain the 2×1–Hessenberg matrix $\bar{H}_1$. Then we have to compute the coefficients c_1, s_1 of the Givens rotation $Q_1 = G_1^1$ with $\bar{H}_1 = Q_1 \left[\begin{smallmatrix} R_1 \\ 0 \end{smallmatrix}\right]$. This leads to

$$Q_1^T \bar{H}_1 = (G_1^1)^T \bar{H}_1 \begin{pmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{21} \end{pmatrix} = \begin{pmatrix} r_{11} \\ \end{pmatrix} \ .$$

with

$$r_{11} = \sqrt{h_{11}^2 + h_{21}^2} \ , \quad c_1 = \frac{h_{11}}{r_{11}} \quad s_1 = \frac{h_{21}}{r_{11}} \ .$$

The minimal solution of $\bar{H}_1^T y_1 = \beta e_1$ is now simply given by

$$y_1 = Q_1 \begin{pmatrix} R_1^{-T} \beta e_1 \\ 0 \end{pmatrix} = \beta G_1^1 \begin{pmatrix} r_{11}^{-1} \\ 0 \end{pmatrix} \ .$$

We define $z_{11} := r_{11}^{-1}$.
The next step begins with the computation of the new column for the Hessenberg matrix $\bar{H}_2$, followed by the QR–decomposition $\bar{H}_2 = Q_2 \left[\begin{smallmatrix} R_2 \\ 0 \end{smallmatrix}\right]$ with $Q_2 = G_1^2 G_2^2$. Therefore we have

$$(G_1^2)^T \bar{H}_2 = \begin{pmatrix} c_1 & s_1 & \\ -s_1 & c_1 & \\ & & 1 \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ & h_{32} \end{pmatrix} = \left(\begin{array}{c|cc} r_{11} & h'_{12} \\ & h'_{22} \\ & h_{32} \end{array}\right) .$$

with r_{11}, c_1 and s_1 as above and

$$h'_{12} = c_1 h_{12} + s_1 h_{22} , \quad h'_{22} = -s_1 h_{12} + c_1 h_{22} .$$

Then we compute G_2^2 via

$$Q_2^T \bar{H}_2 = (G_2^2)^T (G_1^2)^T \bar{H}_2 = \begin{pmatrix} 1 & & \\ & c_2 & s_2 \\ & -s_2 & c_2 \end{pmatrix} \left(\begin{array}{c|cc} r_{11} & h'_{12} \\ & h'_{22} \\ & h_{32} \end{array}\right) = \begin{pmatrix} r_{11} & r_{12} \\ & r_{22} \\ & \end{pmatrix}$$

with

$$r_{12} = h'_{12} , \quad r_{22} = \sqrt{h'^2_{22} + h^2_{32}} , \quad c_2 = \frac{h'_{22}}{r_{22}} \quad s_1 = \frac{h_{32}}{r_{22}} .$$

The minimal solution of $\bar{H}_2^T y_2 = \beta e_1$ is then

$$y_2 = Q_2 \begin{pmatrix} R_2^{-T} \beta e_1 \\ 0 \end{pmatrix} = \beta G_1^2 G_2^2 \begin{pmatrix} z_{21} \\ z_{22} \\ 0 \end{pmatrix}$$

with $z_{21} = z_{11}$, $z_{22} = -z_1 r_{12}/r_{22}$. Obviously we can reuse the results of the first iteration, namely c_1, s_1, r_{11} and z_1. Furthermore in the second iteration we do not need explicitely the values of h_{11} and h_{21}.
The next, third step now is as follows. Gram–Schmidt orthogonalization yields the third column of the Hessenberg matrix $\bar{H}_3$, which is then factorized equivalently to the preceeding steps as $\bar{H}_3 = Q_3 \left[\begin{smallmatrix} R_3 \\ 0 \end{smallmatrix}\right]$ with $Q_3 = G_1^3 G_2^3 G_3^3$.
This gives first

$$(G_1^3)^T \bar{H}_3 = \begin{pmatrix} c_1 & s_1 & & \\ -s_1 & c_1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ & h_{32} & h_{33} \\ & & h_{23} \end{pmatrix} = \left(\begin{array}{c|cc} r_{11} & h'_{12} & h'_{13} \\ & h'_{22} & h'_{23} \\ & h_{32} & h_{33} \\ & & h_{43} \end{array}\right)$$

with r_{11}, c_1, s_1, h'_{12}, h'_{22} as above and

$$h'_{13} = c_1 h_{13} + s_1 h_{23} \ , \quad h'_{23} = -s_1 h_{13} + c_1 h_{23} \ .$$

Then we have to calculate

$$(G_2^3)^T (G_1^3)^T \bar{H}_3 = \begin{pmatrix} 1 & & & \\ & c_2 & s_2 & \\ & -s_2 & c_2 & \\ & & & 1 \end{pmatrix} \left(\begin{array}{c|cc} r_{11} & h'_{12} & h'_{13} \\ & h'_{22} & h'_{23} \\ & h_{32} & h_{33} \\ & & h_{43} \end{array} \right) = \left(\begin{array}{c|cc} r_{11} & r_{12} & h'_{13} \\ & r_{22} & h''_{23} \\ & & h''_{33} \\ & & h_{43} \end{array} \right)$$

with r_{12}, r_{22}, c_2, s_2 as in step 2 and

$$h''_{23} = c_2 h'_{23} + s_2 h_{33} \ , \quad h''_{33} = -s_2 h'_{23} + c_2 h_{33} \ .$$

The factorization is finished by

$$Q_3^T \bar{H}_3 = (G_3^3)^T (G_2^3)^T (G_1^3)^T \bar{H}_3 \quad =$$

$$\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & c_3 & s_3 \\ & & -s_3 & c_3 \end{pmatrix} \left(\begin{array}{cc|c} r_{11} & r_{12} & h'_{13} \\ & r_{22} & h''_{23} \\ & & h''_{33} \\ & & h_{43} \end{array} \right) = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ & r_{22} & r_{23} \\ & & r_{33} \\ & & \end{pmatrix}$$

with

$$r_{13} = h'_{13} \ , \quad r_{23} = h''_{23} \ , \quad r_{33} = \sqrt{h''^2_{33} + h^2_{43}} \ , \quad c_3 = \frac{h''_{33}}{r_{33}} \ , \quad s_3 = \frac{h_{43}}{r_{33}} \ .$$

Therefore the minimal solution of $\bar{H}_3^T y_3 = \beta e_1$ is then

$$y_3 = Q_3 \begin{pmatrix} R_3^{-T} \beta e_1 \\ 0 \end{pmatrix} = \beta G_1^3 G_2^3 G_3^3 \begin{pmatrix} z_{31} \\ z_{32} \\ z_{33} \end{pmatrix}$$

with $z_{31} = z_{11}$, $z_{32} = z_{22}$ and $z_{33} = -(r_{13} z_{11} + r_{23} z_{22})/r_{33}$. To compute y_3 we require neither the first colums of the Hessenberg matrix nor the first columns of the triangular matrix R_3. Given the coefficients c_i, s_i, $i = 1, 2$ and y_2 we need for the new minimal solution y_3 only the values of the actual column in the Hessenberg matrix, which can then be discarded. The generalization of this step by step derivation leads to the following pseudocode representation of the GMERR algorithm.

Pseudocode for left preconditioned GMERR

Given: an initial guess x_0 and a residual tolerance tol.
Define k_{max} and δ_{min}.
Compute preconditioned initial residual $P^{-1}(b - Ax_0)$
Compute first Krylov vector w_1.
    $w_1 = r_0$
    $\beta = \|w_1\|$
    $w_1 = w_1/\beta$
$k = 0$
Perform GMERR iteration.
    $k = k + 1$
    Perform modified Gram-Schmidt process.
        $w_{k+1} = A^T P^{-T} w_k$
        For $i = 1, \ldots, k$
            $h_i = \langle w_{k+1}, w_i \rangle$
            $w_{k+1} = w_{k+1} - h_i w_j$
        $h_{k+1} = \|w_{k+1}\|$
        $w_{k+1} = w_{k+1}/h_{k+1}$
    Update the QR decomposition of H.
        For $i = 1, \ldots, k-1$
            $t = c_i h_i + s_i h_{i+1}$
            $h_{i+1} = -s_i h_i + c_i h_{i+1}$
            $h_i = t$
        $r = 1/\sqrt{h_k h_k + h_{k+1} h_{k+1}}$
        $c_k = r h_k$
        $s_k = r h_{k+1}$
    Update $z = R^{-T}\|r_0\| e_1$.
        If $k = 1$: $z_1 = \beta r$
        If $k > 1$: $z_k = -r \prod_{i=1}^{k-1} y_i h_i$
    Compute minimal solution y_k.
        $y_{k,k+1} = s_k z_k$
        $t = c_k z_k$
        For $j = k-1, \ldots, 1$
            $y_{k,j+1} = s_j z_j + c_j t$
            $t = c_j z_j - s_j t$
        $y_{k,1} = t$
    Compute $\|\delta x_k\|/\|\delta x_1\| = \|y_k - (y_{k-1,1}, \ldots, y_{k-1,k}, 0)^T\|/\|\delta x_1\|$.
    Convergence monitor.
        If $\|\delta x_k\|/\|\delta x_1\| \leq \delta_{min}$ or $k = k_{max}$ THEN perform restart:
            Compute approximate solution $x_k = x_0 + W_{k+1} y$.

Compute preconditioned residual $r_k = P^{-1}(b - Ax_k)$.
Check residual: If $\|r_k\| \leq tol$ stop.
Compute new first Krylov vector w_1.
$w_1 = r_k$
$\beta = \|w_1\|$
$w_1 = w_1/\beta$
$k = 0$
Perform new GMERR iteration.
ELSE continue GMERR iteration.