

AUGMENTED GMRES-TYPE VERSUS CGNE METHODS FOR THE SOLUTION OF LINEAR ILL-POSED PROBLEMS*

ANDREAS NEUBAUER[†]

Abstract. In this paper we compare (augmented) GMRES-type methods and (augmented) CGNE methods. The numerical results show that the CGNE method is more robust and suitable for ill-posed problems with a much higher degree of ill-posedness. GMRES-type methods only yield useful results for very moderately ill-posed problems.

Key words. (augmented) CGNE and GMRES-type methods

AMS subject classifications. 65J20, 65R20, 65R30

1. Introduction. In this paper we deal with linear ill-posed problems

$$Tx = y,$$

where $T : X \rightarrow X$ is a bounded linear operator and X is a Hilbert space, i.e., we assume that the best approximate solution x^\dagger does not depend continuously on the right-hand side y . Since, in general, only noisy data y^δ are available, where we assume that a bound δ for

$$\|y - y^\delta\|_X \leq \delta$$

is known, we have to approximate x^\dagger by so-called regularized solutions; see, e.g., [9]. One of the best-known regularization method is Tikhonov regularization. However, for large scale problems, iterative regularization methods are better suited since one avoids inverting large linear systems: there we look for regularized solutions x_k^δ , where the iteration index k has to be stopped in dependence on δ and the data y^δ in order to obtain good approximations to x^\dagger . One such stopping rule is the discrepancy principle, where the iteration is stopped when for the first time

$$(1.1) \quad \|Tx_k^\delta - y^\delta\|_X \leq \tau\delta$$

for some $\tau > 1$. This index is denoted by k_*^δ .

A very fast method (concerning the number of iterations needed) is the Conjugate Gradient (CG) method. However, this method only works for self-adjoint positive definite operators T . If T is not self-adjoint, then there are two alternatives: the first one is the CGNE method, where the CG method is applied to the normal equation

$$T^*Tx = T^*y.$$

Thus, x_k^δ minimizes the residual in a special Krylov subspace, i.e.,

$$(1.2) \quad \|Tx_k^\delta - y^\delta\|_X = \min_{x \in K_k^C} \|Tx - y^\delta\|_X,$$

$$(1.3) \quad K_k^C := \mathcal{K}_k(T^*T, T^*y^\delta).$$

Note that $\mathcal{K}_k(A, z) := \text{span}\{z, Az, \dots, A^{k-1}z\}$. It follows from the definition of this method that among all iterative regularization methods in the Krylov subspaces $\mathcal{K}_k(T^*T, T^*y^\delta)$

*Received September 5, 2019. Accepted November 5, 2019. Published online on November 12, 2019. Recommended by L. Reichel.

[†]Industrial Mathematics Institute, Johannes Kepler University, A-4040 Linz, Austria (neubauer@indmath.uni-linz.ac.at).

combined with the discrepancy principle as stopping rule, CGNE requires the smallest number of iteration steps. The calculation of the unique minimizer of (1.2) is possible via a fast iterative algorithm (see, e.g., [9, Algorithm 7.1] for the CGLS implementation) going back to Hestenes and Stiefel [11].

The second alternative is the Generalized Minimal Residual (GMRES) method, where one looks for the minimizer

$$(1.4) \quad \|Tx_k^\delta - y^\delta\|_X = \min_{x \in K_k^G} \|Tx - y^\delta\|_X,$$

$$(1.5) \quad K_k^G := \mathcal{K}_k(T, y^\delta).$$

If this minimizer is not unique, then we choose the one of minimal norm, i.e.,

$$x_k^\delta := T_k^\dagger y^\delta \in \mathcal{N}(T_k)^\perp, \quad T_k := TP_k,$$

where P_k is the orthogonal projector from X onto K_k^G . All least-squares solutions in K_k are given by

$$x_k^\delta + z \quad \text{with} \quad z \in \mathcal{N}(T_k) \cap K_k^G.$$

Noting that

$$\mathcal{N}(T_k) = (K_k^G)^\perp + (\mathcal{N}(T) \cap K_k^G) \quad \text{and} \quad \mathcal{N}(T_k)^\perp = K_k^G \cap (\mathcal{N}(T) \cap K_k^G)^\perp,$$

there is only one least-squares solution in K_k^G if and only if $\mathcal{N}(T) \cap K_k^G = \{0\}$.

This method was developed for finite-dimensional linear systems. However, it directly carries over to linear problems in Hilbert spaces (see Section 2). Also an efficient implementation based on the Arnoldi algorithm (see [13]) can be used in Hilbert spaces. As will be seen below, this method needs much more storage than the CGNE method.

It was shown in [3] that the range restricted GMRES (RRGMRES), where in the minimization problem (1.4) K_k^G is replaced by $\mathcal{K}_k(T, Ty^\delta)$, often gives higher accuracy than GMRES when applied to the solution of linear discrete ill-posed problems.

Of course, for each regularization method the question arises if it is convergent, i.e., we want to know if the regularized solutions $x_{k_*^\delta}^\delta$, where k_*^δ is the stopping index from the discrepancy principle (1.1), converge to x^\dagger as δ tends to zero.

Results about convergence and even convergence rates are well-known for the CGNE method; see, e.g., [9]. However, almost no results are known for the GMRES method. Therefore, we discuss convergence aspects in Section 3.

If the operators T and T^* are very smooth, then for CGNE and RRGMRRES, the regularized solutions will be also smooth. For GMRES this will be true for all solution parts except the first one. Therefore, it will not be easy to find very good approximations in Krylov subspaces when we expect discontinuities in the exact solution x^\dagger . An improvement can be achieved if one augments the Krylov subspaces by some small space that represents non-smooth features of the desired solution. This approach will be discussed in Section 4.

In Section 5 we deal with integral equations and their discretization to obtain finite-dimensional problems. Numerical results are presented in Section 6. Finally, a short conclusion is given in the last section.

2. The GMRES method in Hilbert spaces. As already mentioned in the introduction, in this section we discuss the efficient computation of the minimizers x_k^δ from (1.4).

The aim is to inductively construct orthonormal functions v_i such that

$$K_k^G := \text{span}\{v_1, \dots, v_k\}.$$

This is possible, as long as $\dim K_k^G = k$ holds. Assuming that $Ty^\delta \neq 0$, the induction is started with

$$v_1 := \frac{y^\delta}{\|y^\delta\|_X} \quad \text{for GMRES} \quad \text{and} \quad v_1 := \frac{T y^\delta}{\|T y^\delta\|_X} \quad \text{for RRGMRRES}.$$

Assume that v_1, \dots, v_k have already been computed. Then the next function v_{k+1} is found as follows: calculate

$$(2.1) \quad h_{ik} := \langle v_i, T v_k \rangle_X, \quad i = 1, \dots, k, \quad \text{and} \quad z_{k+1} := T v_k - \sum_{i=1}^k h_{ik} v_i.$$

Note that, due to the orthonormality of the functions v_i , the calculation of z_{k+1} can be done recursively: starting with $z_{k+1} := T v_k$, the update is calculated via

$$h_{ik} = \langle v_i, z_{k+1} \rangle_X, \quad z_{k+1} \leftarrow z_{k+1} - h_{ik} v_i, \quad i = 1, \dots, k.$$

If $z_{k+1} = 0$, then $T v_k$ can be written as a linear combination of the functions v_1, \dots, v_k . Since $K_{k+1}^G = K_k^G \cup T K_k^G = \text{span}\{v_1, \dots, v_k, T v_k\}$ (see, e.g., [1]), this can only happen if $\dim K_{k+1}^G = k$. The algorithm breaks down (see Proposition 3.1 below), and we set $v_{k+1} := 0$.

If $z_{k+1} \neq 0$, then by construction $\langle z_{k+1}, v_i \rangle_X = 0$ for all $i = 1, \dots, k$. Therefore, we may set

$$(2.2) \quad h_{k+1,k} := \|z_{k+1}\|_X \quad \text{and} \quad v_{k+1} := \frac{z_{k+1}}{h_{k+1,k}}.$$

Combining (2.1) and (2.2) immediately yields

$$(2.3) \quad T v_k = \sum_{i=1}^{k+1} h_{ik} v_i.$$

Note that this formula is also meaningful for the case where $z_{k+1} = v_{k+1} = 0$.

Assuming that this algorithm does not break down until k , i.e., that $\dim K_k^G = k$, and noting that

$$(2.4) \quad P_{k+1} y = \sum_{i=1}^{k+1} \hat{y}_i v_i, \quad \text{with} \quad \hat{y}_i := \langle y, v_i \rangle_X,$$

is the orthogonal projector onto K_{k+1}^G , using formula (2.3), we may rewrite the residuum $\|T x - y^\delta\|_X$ for any

$$x = \sum_{j=1}^k \xi_j v_j \in K_k^G$$

via

$$\begin{aligned} \|T x - y^\delta\|_X^2 &= \|T x - P_{k+1} y^\delta\|_X^2 + \|(I - P_{k+1}) y^\delta\|_X^2 \\ &= \left\| \sum_{i=1}^{k+1} \left(\sum_{j=\max\{i-1, 1\}}^k h_{ij} \xi_j - \hat{y}_i \right) v_i \right\|_X^2 + \|(I - P_{k+1}) y^\delta\|_X^2. \end{aligned}$$

Defining the $(k+1) \times k$ Hessenbergmatrix H_k by

$$(2.5) \quad (H_k)_{ij} := \begin{cases} h_{ij}, & 1 \leq i \leq j+1, \\ 0, & \text{else,} \end{cases}$$

with h_{ij} as in (2.1) ($k = j$) and noting that

$$\|(I - P_{k+1})y^\delta\|_X^2 = \|y^\delta\|_X^2 - \sum_{i=1}^{k+1} \hat{y}_i^2,$$

we finally obtain that

$$(2.6) \quad \|Tx - y^\delta\|_X^2 = \|H_k \xi - \hat{y}\|_{k+1}^2 + \|y^\delta\|_X^2 - \|\hat{y}\|_{k+1}^2,$$

where $\xi := (\xi_1, \dots, \xi_k) \in \mathbb{R}^k$, $\hat{y} := (\hat{y}_1, \dots, \hat{y}_{k+1}) \in \mathbb{R}^{k+1}$, and $\|\cdot\|_{k+1}$ denotes the Euclidean norm in \mathbb{R}^{k+1} . Note that for the GMRES method $\hat{y} = \|y^\delta\|_X (1, 0, \dots, 0)$ and $\|(I - P_{k+1})y^\delta\|_X = 0$.

The minimizer ξ of (2.6) may be easily calculated by transforming H_k via Givens rotations to an upper triangular matrix. One may calculate the solutions in K_k^G recursively by updating the matrix H_k in every step and then applying the orthonormal transforms to the new column. Combined with the discrepancy principle (1.1) this yields the following algorithm:

ALGORITHM 2.1. (GMRES and RRGMRRES + discrepancy principle.)

Let $Ty^\delta \neq 0$, $\delta > 0$, $\tau > 1$, $\varepsilon > 0$ be a small tolerance, and set gm:=true if GMRES and gm:=false if RRGMRRES.

```

1:  $r := \|y^\delta\|_X; k := 0;$ 
2: if (gm) {
3:    $z := y^\delta; h := r; \xi_1 := r;$ 
4: } else {
5:    $z := Ty^\delta; h = \|z\|_X; \xi_1 = \langle y^\delta, z \rangle_X / h; u := r^2;$ 
6: }
7: while ( $r \geq \tau\delta$ ) {
8:    $k := k + 1; v_k = z/h; z := Tv_k;$ 
9:   for ( $i = 1, \dots, k$ ): {
10:     $R_{ik} := \langle v_i, z \rangle_X; z := z - R_{ik}v_i;$ 
11:   }
12:   for ( $i = 1, \dots, k-1$ ): {
13:     $\beta := R_{i+1,k}; R_{i+1,k} := c_i\beta - s_iR_{ik}; R_{ik} := c_iR_{ik} + s_i\beta;$ 
14:   }
15:    $h := \|z\|_X;$ 
16:   if ( $h < \varepsilon$ ) {
17:     $r := 0;$ 
18:    if ( $|R_{kk}| < \varepsilon$ ) {  $k := k - 1;$  }
19:   } else {
20:     $\beta := \sqrt{R_{kk}^2 + h^2}; c_k := R_{kk}/\beta; s_k := h/\beta; R_{kk} := \beta;$ 
21:    if (gm) {
22:      $\xi_{k+1} := -s_k\xi_k; \xi_k := c_k\xi_k; r := |\xi_{k+1}|;$ 
23:    } else {
24:      $\beta := \langle y^\delta, z \rangle_X / h; \xi_{k+1} := c_k\beta - s_k\xi_k; \xi_k := c_k\xi_k + s_k\beta;$ 
25:      $u := u - \xi_k^2; r := \sqrt{u};$ 
26:    }

```

27: }
 28: }
 29: for $(i = k, \dots, 1)$: {
 30: for $(l = i + 1, \dots, k)$ { $\xi_l := \xi_l - R_{il}\xi_l$ };
 31: $\xi_i := \xi_i / R_{ii}$;
 32: }
 33: $x_k^\delta := \sum_{i=1}^k \xi_i v_i$;

3. Convergence aspects of the GMRES method. In the following proposition we collect some results concerning the dimension of K_k^G . We use y instead of y^δ since the results are independent from the noise aspect.

PROPOSITION 3.1. *Let $T \in L(X, X)$ and $K_k^G := \mathcal{K}_k(T, y)$. Then the following assertions hold:*

(i) *If $\dim K_k^G = k$, then $\dim K_j^G = j$ for all $j < k$. Moreover,*

$$(3.1) \quad \mathcal{N}(T) \cap K_j^G = \{0\} \quad \text{for all } j < k,$$

$$(3.2) \quad \dim(\mathcal{N}(T) \cap K_k^G) \leq 1.$$

(ii) *If the condition*

$$(3.3) \quad \dim K_k^G = k = \dim K_{k+1}^G$$

holds, then $K_j^G = K_k^G$ for all $j > k$, and there is a unique $\xi \in \mathbb{R}^k$ such that

$$(3.4) \quad T^k y = \sum_{i=1}^k \xi_i T^{i-1} y.$$

Moreover, if $\xi_1 \neq 0$, then $\mathcal{N}(T) \cap K_k^G = \{0\}$ and $y = Tx$ for $x = x_k = T_k^\dagger y \in K_k^G$. Note that x_k need not be equal to the best approximate solution x^\dagger .

If $\xi_1 = 0$, then $\dim(\mathcal{N}(T) \cap K_k^G) = 1$, and $x_{k-1} = T_{k-1}^\dagger y$ is a least-squares solution of $Tx = y$ in K_k^G that is not necessarily equal to $T_k^\dagger y$.

Proof.

(i) Due to the definition of Krylov subspaces, it immediately follows that

$$\dim K_{j+l}^G \leq \dim K_j^G + l \quad \text{for all } l \geq 0.$$

This implies that, if $\dim K_j^G < j < k$, then also $\dim K_k^G < j + k - j = k$, in contradiction to the assumption that $\dim K_k^G = k$.

Assume that a $j < k$ exists with $\mathcal{N}(T) \cap K_j^G \neq \{0\}$. Then a vector $\xi \in \mathbb{R}^j \setminus \{0\}$ exists such that

$$0 = T \left(\sum_{i=1}^j \xi_i T^{i-1} y \right) = \sum_{i=1}^j \xi_i T^i y.$$

Thus, $\dim K_{j+1}^G < j + 1$, which is again a contradiction.

We now prove (3.2): assume that $x_i \in \mathcal{N}(T) \cap K_k^G$ with $x_i \neq 0$, $i = 1, 2$. Due to (3.1), x_i can be written as $x_i = c_i (T^{k-1} y - \bar{x}_i)$ with $\bar{x}_i \in K_{k-1}^G$ and $c_i \neq 0$. Since $Tx_i = 0$, we get that

$$(3.5) \quad T^k y = T\bar{x}_i.$$

Moreover, $T(\bar{x}_1 - \bar{x}_2) = 0$ and (3.1) imply that $\bar{x}_1 = \bar{x}_2$ and hence that $c_2x_1 = c_1x_2$. This proves (3.2).

- (ii) Let us assume that condition (3.3) holds. Then $K_{k+1}^G = K_k^G$ and $T^k y \in K_k^G$ yielding (3.4). The uniqueness of the vector $\xi \in \mathbb{R}^k$ follows from the fact that the elements $y, Ty, \dots, T^{k-1}y$ are linearly independent. It immediately follows by induction that $T^j y \in K_k^G$ for all $j \geq k$ and hence that $K_j^G = K_k^G$ for all $j > k$. (3.4) implies that

$$\xi_1 y = T\bar{x}_k \quad \text{with} \quad \bar{x}_k := T^{k-1}y - \sum_{i=1}^{k-1} \xi_{i+1} T^{i-1}y \in K_k^G \setminus K_{k-1}^G.$$

Furthermore, it follows from (3.4) and (3.5) that for some $\bar{x} \in K_{k-1}^G$

$$\dim(\mathcal{N}(T) \cap K_k^G) = 1 \Leftrightarrow \sum_{i=1}^k \xi_i T^{i-1}y = T\bar{x} \Leftrightarrow \xi_1 = 0.$$

If $\xi_1 \neq 0$, then $\mathcal{N}(T) \cap K_k^G = \{0\}$, and $\xi_1^{-1}\bar{x}_k = x_k$ is an exact solution of $Tx = y$. If $\mathcal{N}(T) \neq \{0\}$, then x_k need not be equal to x^\dagger .

If $\xi_1 = 0$, then $\bar{x}_k \in \mathcal{N}(T) \cap K_k^G$ since for all $x \in K_k^G$ it holds that $x = \bar{x} + c\bar{x}_k$ for some $\bar{x} \in K_{k-1}^G$ and $c \in \mathbb{R}$, and since $\|Tx - y\|_X = \|T\bar{x} - y\|_X$, x_{k-1} is a least-squares solution of $Tx = y$ in K_k^G . Note that it might differ from

$$x_k = x_{k-1} - \frac{\langle x_{k-1}, \bar{x}_k \rangle_X}{\|\bar{x}_k\|_X^2} \bar{x}_k. \quad \square$$

Almost all results of Proposition 3.1 remain valid also for the case of RRGMRRES, where K_k^G is replaced by $\mathcal{K}_k(T, Ty)$. However, in the case that $\xi_1 \neq 0$, it is not guaranteed that an exact solution of the equation $Tx = y$ is found.

If $\mathcal{R}(T)$ is finite-dimensional, then $\dim K_{k+1}^G = k$ will occur for some $k \leq \dim \mathcal{R}(T)$. The case $\xi_1 = 0$ in Proposition 3.1 (ii) above can never occur for injective operators T . This means that for finite-dimensional injective operators and exact data, the algorithm will always converge to the exact solution.

The question of course is if we get convergence for noisy data when the iteration is stopped according to (1.1) and δ is going to zero. So far we only found one result addressing this question in [4]. Stability and convergence were proven under the very strong condition that condition (3.3) holds uniformly for all data. This practically means that the problem has to be finite-dimensional, i.e., the problem may be ill-conditioned but not ill-posed. Unfortunately, no results are yet available for the general case.

The following examples give an indication why it might be complicated to find conditions that guarantee convergence in the general case.

EXAMPLE 3.2. Let $X := l^2$, and let $T : l^2 \rightarrow l^2$ be defined as follows:

$$x := (x_1, x_2, \dots) \mapsto Tx := (\eta_1, \eta_2, \dots)$$

with

$$\eta_{2j-1} := 0 \quad \text{and} \quad \eta_{2j} := a_j x_{2j-1}, \quad j \in \mathbb{N},$$

where (a_j) is a sequence satisfying $a_j \neq 0$ and $|a_j| \leq c$ for all $j \in \mathbb{N}$ and some $c > 0$. Then the following assertions obviously hold:

- $\mathcal{N}(T) = \{x \in l^2 : x_{2j-1} = 0 \text{ for all } j \in \mathbb{N}\}$.
- $T^2 = 0$.
- $\mathcal{R}(T)$ is dense in $\mathcal{N}(T)$.
- $\mathcal{R}(T) = \mathcal{N}(T)$ if and only if $|a_j| \geq \underline{c}$ for all $j \in \mathbb{N}$ and some $\underline{c} > 0$.
- If $\lambda \neq 0$, then $x = -\frac{1}{\lambda}y - \frac{1}{\lambda^2}Ty$ is the unique solution of $Tx - \lambda x = y$ depending continuously on y . Thus, $\sigma(T) = \{0\}$.
- T is compact if and only if $a_j \rightarrow 0$ as $j \rightarrow \infty$.
- T is a Hilbert-Schmidt operator if and only if $(a_j) \in l^2$.

Note that if there exists a subsequence of a_j tending to zero and another one being bounded from below by a positive constant, then T is not compact and $\mathcal{R}(T)$ is not closed. Thus, the problem of solving $Tx = y$ is still ill-posed.

Let us assume that $y^\delta \in \mathcal{N}(T) \setminus \{0\}$. Using GMRES, we obtain that

$$K_k^G := \mathcal{K}_k(T, y^\delta) = \text{span}\{y^\delta\}, \quad k \in \mathbb{N}.$$

Thus, $x_k^\delta = 0$, and we never get convergence except for the trivial case that $y = 0$.

If $y^\delta \notin \mathcal{N}(T)$, then we obtain that

$$K_1^G = \overline{\text{span}\{y^\delta\}} \quad \text{and} \quad K_k^G = \text{span}\{y^\delta, Ty^\delta\}, \quad k \geq 2.$$

Then

$$x_1^\delta = \lambda y^\delta \quad \text{and} \quad x_k^\delta = \lambda y^\delta - \lambda^2 Ty^\delta, \quad k \geq 2 \quad \text{with} \quad \lambda = \frac{\sum_{l=0}^{\infty} a_l y_{2l-1}^\delta y_{2l}^\delta}{\sum_{l=0}^{\infty} a_l^2 (y_{2l-1}^\delta)^2}.$$

If, for instance,

$$x^\dagger = (1, 0, 1, 0, 0, \dots), \quad y = Tx^\dagger = (0, a_1, 0, a_2, 0, \dots), \quad y^\delta = y + (\delta, 0, \dots),$$

then

$$\|y - y^\delta\|_{l^2} = \delta, \quad x_1^\delta = (1, a_1/\delta, 0, a_2/\delta, 0, \dots), \quad \text{and} \quad x_k^\delta = (1, 0, 0, a_2/\delta, 0, \dots)$$

for $k \geq 2$. None of these approximations converges towards x^\dagger if $\delta \rightarrow 0$.

EXAMPLE 3.3. Let $X := l^2$, and let $T : l^2 \rightarrow l^2$ be defined as follows:

$$x := (x_1, x_2, \dots) \mapsto Tx := (\eta_1, \eta_2, \dots)$$

with

$$\eta_1 := 0 \quad \text{and} \quad \eta_{j+1} := a_j x_j, \quad j \in \mathbb{N},$$

where (a_j) is a sequence satisfying $a_j \neq 0$ and $|a_j| \leq c$ for all $j \in \mathbb{N}$ and some $c > 0$. Then the following assertions obviously hold:

- $\mathcal{N}(T) = \{0\}$ and $\overline{\mathcal{R}(T)} = \{y \in l^2 : y_1 = 0\}$.
- $\mathcal{R}(T) = \overline{\mathcal{R}(T)}$ if and only if $|a_j| \geq \underline{c}$ for all $j \in \mathbb{N}$ and some $\underline{c} > 0$.
- T has no eigenvalues.
- T is compact if and only if $a_j \rightarrow 0$ as $j \rightarrow \infty$.
- T is a Hilbert-Schmidt operator if and only if $(a_j) \in l^2$.

As in Example 3.2, it holds that, if there exists a subsequence of a_j tending to zero and another one being bounded from below by a positive constant, then T is not compact and $\mathcal{R}(T)$ is not closed. Thus, the problem of solving $Tx = y$ is still ill-posed.

Since $(T^i x)_j = 0$, for all $j \leq i$, it will always hold that $\dim K_k^G = k$. Thus, the GMRES algorithm will never break down.

We first consider the case of exact data ($y^\delta = y = Tx^\dagger$). Let us assume that $x^\dagger = (x_k)$ is such that $x_j = 0$ for all $1 \leq j < l$ and $x_l \neq 0$, then, due to the definition of T , all elements of the spaces K_k^G have the property that the first l components are zero. Therefore, the iterates x_k can never converge towards x^\dagger .

Let us now assume that x^\dagger is as above and that $y^\delta = Tx^\dagger + \delta u$ with $\|u\|_{l^2} = 1$ and $u_j = 0$ for all $j \leq l$. Then

$$\|Tx - y^\delta\|_{l^2} \geq |a_l x_l + \delta u_{l+1}| \quad \text{for all } x \in K_k^G.$$

This means that for $\delta > 0$ sufficiently small, the discrepancy principle will not stop.

Let us consider the following concrete case: $a_j := j^{-p}$, $j \in \mathbb{N}$, for some $p \geq 0$. If $p = 0$, then T is the down-shift operator with T^\dagger being bounded. If $p > 0$, then T is compact. If $p > \frac{1}{2}$, then T is even a Hilbert-Schmidt operator. Moreover, let

$$\begin{aligned} x^\dagger &:= (1, 1, 0, \dots), & y &:= Tx^\dagger = (0, 1, 2^{-p}, 0, \dots), \\ y^\delta &:= (1 + c_p \delta)y, & c_p &:= 1/\sqrt{1 + 4^{-p}}. \end{aligned}$$

Then $\|y^\delta - y\|_{l^2} = \delta$, and all elements $x \in K_k^G$ satisfy that

$$(3.6) \quad \begin{aligned} x &= (1 + c_p \delta)(0, \xi_1, \dots, \xi_k, \xi_{k+1}, 0 \dots) \quad \text{with} \\ \xi_{k+1} &= \sum_{i=1}^k (-1)^{k-i} \left(\frac{i!}{(k+1)!} \right)^p \xi_i. \end{aligned}$$

A straightforward calculation yields that x_k^δ is as in (3.6) with

$$\begin{aligned} \xi_1 &= 2^{-p} + (-1)^k \frac{1}{((k+1)!)^p} \xi_{k+1}, \\ \xi_i &= (-1)^{k+1-i} \left(\frac{i!}{(k+1)!} \right)^p \xi_{k+1}, \quad i = 2, \dots, k, \\ \xi_{k+1} &= (-1)^{k+1} \frac{((k+1)!)^p}{2^p \sum_{i=1}^{k+1} (i!)^{2p}}. \end{aligned}$$

In addition, $x_k^\delta \rightarrow \bar{x} := (0, 2^{-p}, 0, \dots) \neq x^\dagger$ as $\delta \rightarrow 0$ and $k \rightarrow \infty$ since

$$\|x_k^\delta - \bar{x}\|_{l^2}^2 = (1 - c_p^2 \delta^2) \frac{|\xi_{k+1}|}{(2(k+1)!)^p} + c_p^2 \delta^2 4^{-p}.$$

Both examples show that convergence towards x^\dagger can not be guaranteed if T is not injective and also not for infinite-dimensional injective operators.

4. Augmented CGNE and GMRES methods. As already mentioned in the introduction one would like to improve the quality of the approximations when the functions in the Krylov spaces are much smoother than the expected exact solution by augmenting the Krylov subspace.

The augmented (or also called enriched) CGNE method was discussed in [5], where problem (1.2) is replaced by the following one: find x_k^δ as the minimizer of

$$(4.1) \quad \|Tx_k^\delta - y^\delta\|_X = \min_{x \in K_k^C + W_n} \|Tx - y^\delta\|_X,$$

where K_k^C is as in (1.3) and

$$(4.2) \quad W_n := \text{span}\{w_1, \dots, w_n\}$$

with n small. We always assume that the functions Tw_i , $i = 1, \dots, n$, are linearly independent.

Let us assume that $W_{n,k}$ is such that $K_k^C + W_n = K_k^C + W_{n,k}$ with $TK_k^C \perp TW_k$. Then an efficient computation is possible noting that

$$\|T(x+z) - y^\delta\|_X = \|Q_k y^\delta - Tx\|_X + \|(I - Q_k)y^\delta - Tz\|_X$$

with $x \in K_k^C$ and $z \in W_{n,k}$, where Q_k is the orthogonal projector onto TK_k . Thus, we compute the standard CGNE iterate $\bar{x}_k^\delta \in K_k^C$, and then we look for the minimizer z_k^δ of

$$(4.3) \quad \min_{z \in W_k} \|d_k^\delta - Tz\|_X$$

with $d_k^\delta := y^\delta - T\bar{x}_k^\delta = (I - Q_k)y^\delta$. Then $x_k^\delta = \bar{x}_k^\delta + z_k^\delta$. Note that this minimizer is not unique if $TK_k^C \cap TW_n \supset \{0\}$. Let $W_{n,k} = \text{span}\{w_{1,k}, \dots, w_{n,k}\}$. Then

$$w_{j,k} := w_{j,k-1} - \frac{\langle Tw_{j,k-1}, q_k \rangle_X}{\|q_k\|_X^2} p_k, \quad w_{j,0} := w_j,$$

where p_k and q_k are the usual functions produced by the CGNE algorithm; see, e.g., [9, Algorithm 7.1]. This guarantees that $\langle Tw_{j,k}, Tp_i \rangle_X = 0$ for all $1 \leq i \leq k$.

The minimizer of (4.3) is then given by $z_k = \sum_{j=1}^n \eta_j w_{j,k}$, where $\eta = (\eta_1, \dots, \eta_n)$ solves the linear system

$$[\langle Tw_{i,k}, Tw_{j,k} \rangle_X] \eta = [\langle Tw_{i,k}, d_k^\delta \rangle_X].$$

The same can be done for the GMRES method (and also for the RRGMRES method), i.e., (1.4) is replaced by the problem of finding x_k^δ such that

$$(4.4) \quad \|Tx_k^\delta - y^\delta\|_X = \min_{x \in K_k^C + W_n} \|Tx - y^\delta\|_X$$

with K_k^C as in (1.5) and W_n as in (4.2).

The authors of [1] wanted to solve this problem. However, their presented algorithm solves the problem of finding x_k^δ such that

$$\|Tx_k^\delta - y^\delta\|_X = \min_{x \in \tilde{K}_{n,k} + W_n} \|Tx - y^\delta\|_X,$$

where $\tilde{K}_{n,k} := \mathcal{K}_k((I - Q_n)T, (I - Q_n)y^\delta)$ and Q_n is the orthogonal projector onto TW_n . The advantage of this approach is that one can first solve the problem of finding $\tilde{x}_{n,k}^\delta \in \tilde{K}_{n,k}$ such that

$$\|(I - Q_n)(T\tilde{x}_{n,k}^\delta - y^\delta)\|_X = \min_{x \in \tilde{K}_k} \|(I - Q_n)(Tx - y)\|_X,$$

and then in a second step one looks for $w_k^\delta \in W_n$ as the (unique) solution of

$$Tw_k^\delta = Q_n(y^\delta - T\tilde{x}_{n,k}^\delta).$$

The solution of (4.4) is a little bit more involved. A nice fast algorithm was presented in [7]: an efficient solution is possible if after running the Arnoldi algorithm to find the orthonormal basis v_1, \dots, v_{k+1} of $K_{k+1}^G = K_k^G \cup TK_k^G$, one recursively computes an orthonormal basis $\bar{v}_1, \dots, \bar{v}_n$ of $P_{k+1}TW_n$ with P_{k+1} as in (2.4):

ALGORITHM 4.1. (Computation of an orthonormal basis for $P_{k+1}TW_n$.)

- 1: for $(j = 1, \dots, n)$ {
- 2: $\bar{v}_j := Tw_j$;
- 3: for $(i = 1, \dots, k + 1)$ { $G_{ij} := \langle v_i, \bar{v}_j \rangle_X$; $\bar{v}_j := \bar{v}_j - G_{ij}v_i$; }
- 4: for $(i = 1, \dots, j - 1)$ { $\bar{G}_{ij}^k := \langle \bar{v}_i, \bar{v}_j \rangle_X$; $\bar{v}_j := \bar{v}_j - \bar{G}_{ij}^k \bar{v}_i$; }
- 5: $\bar{G}_{jj}^k := \|\bar{v}_j\|_X$; $\bar{v}_j := \bar{v}_j / \bar{G}_{jj}^k$;
- 6: }

It then holds that $G_{ij} := \langle v_i, Tw_j \rangle_X$, for $i = 1, \dots, k + 1$, and that $\bar{G}_{ij}^k := \langle \bar{v}_i, Tw_j \rangle_X$ for $i \leq j$. Moreover,

$$Tw_j = \sum_{i=1}^{k+1} G_{ij}v_i + \sum_{i=1}^j \bar{G}_{ij}^k \bar{v}_i.$$

For any

$$x = \sum_{j=1}^k \xi_j v_j + \sum_{i=1}^n \eta_i w_i,$$

we may rewrite the residuum as follows (compare (2.6)):

$$\|Tx - y\|_X^2 = \left\| \begin{pmatrix} H_k & G_k \\ 0 & \bar{G}_k \end{pmatrix} \begin{pmatrix} \xi \\ \eta \end{pmatrix} - \begin{pmatrix} \hat{y} \\ \bar{y} \end{pmatrix} \right\|_{k+1+p}^2 + \|y\|_X^2 - \|\hat{y}\|_{k+1}^2 - \|\bar{y}\|_n^2$$

with H_k as in (2.5),

$$(G_k)_{ij} := G_{ij}, \quad (\bar{G}_k)_{ij} := \begin{cases} \bar{G}_{ij}^k, & i \leq j, \\ 0, & i > j, \end{cases}$$

$\xi \in \mathbb{R}^k$, $\eta \in \mathbb{R}^n$, $\hat{y} \in \mathbb{R}^{k+1}$, and $\bar{y} \in \mathbb{R}^n$, where $\hat{y}_i := \langle y, v_i \rangle_X$ and $\bar{y}_i := \langle y, \bar{v}_i \rangle_X$.

Note that during the iteration process, the matrix \bar{G}_k has to be recalculated in each step. In case of RRGMRES this also holds for the vector \bar{y} . However, this can be done recursively since

$$(4.5) \quad \bar{G}_{ii}^k = \left(A_{ii}^k - \sum_{l=1}^{i-1} (\bar{G}_{li}^k)^2 \right)^{\frac{1}{2}}, \quad \bar{G}_{ij} = (\bar{G}_{ii}^k)^{-1} \left(A_{ij}^k - \sum_{l=1}^{i-1} \bar{G}_{li}^k \bar{G}_{lj}^k \right),$$

$$\bar{y}_i = (\bar{G}_{ii}^k)^{-1} \left(b_i^k - \sum_{l=1}^{i-1} \bar{G}_{li}^k \bar{y}_l \right),$$

for $1 \leq i \leq j \leq n$, and A_k and b_k are defined by induction via

$$(4.6) \quad A_{ij}^1 := \langle Tw_i, Tw_j \rangle_X - \sum_{l=1}^2 G_{li} G_{lj} \quad \text{and} \quad A_{ij}^{k+1} := A_{ij}^k - G_{k+2,i} G_{k+2,j},$$

$$b_j^1 := \langle y^\delta, Tw_j \rangle_X - \sum_{l=1}^2 G_{lj} \hat{y}_l \quad \text{and} \quad b_j^{k+1} := b_j^k - G_{k+2,j} \hat{y}_{k+2}.$$

Since n is usually very small, (4.5) and (4.6) are a very fast alternative for the computation of the matrix \bar{G}_k . The functions \bar{v}_j need not be calculated. This approach is even slightly faster than the one presented in [7].

A comment on the dimension of $P_{k+1}TW_n$: above we always assumed that it is equal to n . However, although the functions Tw_1, \dots, Tw_n are assumed to be linearly independent, it might happen that $K_{k+1}^G \cap TW_n \neq \{0\}$. This will rarely occur for infinite-dimensional operators since TK_k^G consists usually of much smoother functions than TW_n . However, if $\mathcal{R}(T)$ is finite-dimensional, which is the case for any discretization of infinite-dimensional problems, then this will occur for k large enough. Whenever $\bar{G}_{ii}^k = 0$, the function Tw_i is a linear combination of the other ones and, therefore, we can drop this function and reduce n by 1. Implementation details will be shown in the next section.

5. Discretization of integral equations. In our numerical examples we deal with linear Fredholm integral equations of the first kind, i.e.,

$$(Tx)(s) = \int_0^1 k(s,t)x(t) dt$$

with an L^2 -kernel k . Then $T : X \rightarrow X$ with $X := L^2[0,1]$ is compact. For the numerical calculations, we approximate this operator by the finite-dimensional operator $Q_m T$, where Q_m is the orthogonal projector onto $X_m := \text{span}\{\varphi_0, \dots, \varphi_m\}$ and the φ_i are the well-known hat functions, i.e., linear splines with $\varphi_i(s_j) = \delta_{ij}$, $s_j := j/m$.

All integrals that can not be computed exactly are approximated using a four-point Gaussian quadrature rule on each subinterval $[s_j, s_{j+1}]$, $j = 0, \dots, m-1$.

All functions $x \in X_m$ are identified by their representation vectors, i.e., the function

$$x = \sum_{i=0}^m x_i \varphi_i \in X_m$$

corresponds to the vector

$$x := (x_0, \dots, x_m) \in \mathbb{R}^{m+1}$$

and vice versa. Thus, we use the same letter for the function in X_m and its representation vector in \mathbb{R}^{m+1} . The inner product of $u, v \in X_m$ is given by

$$(5.1) \quad \langle u, v \rangle_X = u^\top H v \quad \text{with} \quad H_{ij} := \langle \varphi_i, \varphi_j \rangle_X.$$

The matrix H is a symmetric tridiagonal matrix.

Noisy data are created via

$$y_i^\delta = \langle y, \varphi_i \rangle + \kappa \rho_i,$$

with ρ_i uniformly distributed in $[-1, 1]$ and κ such that $\|Q_m(y^\delta - y)\|_X = \delta$, i.e.,

$$\kappa = \delta / \sqrt{\rho^\top z} \quad \text{with} \quad H z = \rho.$$

For the numerical calculation of the augmented CGNE iterates with T in (4.1) replaced by $Q_m T$, we need the matrix

$$(5.2) \quad Q := [\langle T^* \varphi_i, T^* \varphi_j \rangle]_{i,j=0}^m.$$

The approximation of Q with quadrature rules usually costs $O(m^3)$. However, for our numerical examples we could calculate Q exactly using a computer algebra package. This reduces the effort to $O(m^2)$ operations. Using this matrix we get the following algorithm:

ALGORITHM 5.1. (augmented CGNE.)

Let $\delta > 0, \tau > 1$. Moreover, let H be defined as in (5.1), and let Q be as in (5.2).

```

1:  $x := 0; k := 0;$ 
2: Solve  $H z = y^\delta; r := (y^\delta)^\top z;$ 
3: for  $(j = 1, \dots, n)$  {
4:    $\bar{w}_j := [\langle T w_j, \varphi_i \rangle]_{i=0, \dots, m}; \rho_j := 0;$ 
5:   Solve  $H v = \bar{w}_j;$ 
6:   for  $(i = 1, \dots, j)$  {  $A_{ij} := \bar{w}_i^\top v;$  }
7:    $b_j := (y^\delta)^\top v;$ 
8: }
9: Solve  $A \eta = b$  with  $A = (A_{ij}), A_{ji} = A_{ij}, 1 \leq i, j \leq n, b := (b_1, \dots, b_n);$ 
10:  $\hat{r} := r - \eta^\top b;$ 
11: while  $(\hat{r} \geq (\tau \delta)^2)$  {
12:   if  $(k = 0)$  {
13:      $w := Q z; p := z; \gamma := z^\top w;$ 
14:   } else {
15:      $z := z - \alpha q;$ 
16:      $v := Q z; \kappa := z^\top v; \beta := \kappa / \gamma; \gamma := \kappa;$ 
17:      $p := z + \beta p; w := v + \beta w;$ 
18:   }
19:   Solve  $H q = w; \kappa := q^\top w; \alpha := \gamma / \kappa;$ 
20:    $x := x + \alpha p; r := r - \alpha \gamma;$ 
21:   for  $(j = 1, \dots, n)$  {
22:      $\beta := \bar{w}_j^\top q / \kappa; \bar{w}_j := \bar{w}_j - \beta w; \rho_j := \rho_j + \beta p;$ 
23:      $b_j := \bar{w}_j^\top z;$  Solve  $H v := \bar{w}_j;$ 
24:     for  $(i = 1, \dots, j)$  {  $A_{ij} := \bar{w}_i^\top v;$  }
25:   }
26:   Solve  $A \eta = b;$ 
27:    $\hat{r} := r - \eta^\top b;$ 
28:    $k := k + 1;$ 
29: }
30:  $x_k^\delta := \sum_{i=0}^m \left( x_i - \sum_{j=1}^n \rho_{ji} \eta_j \right) T^* \varphi_i + \sum_{j=1}^n \eta_j w_j;$ 

```

For the numerical calculation of the augmented GMRES or RRGMRES iterates with T in (4.4) replaced by $Q_m T$, we need the matrix

$$(5.3) \quad B := [\langle \varphi_i, T \varphi_j \rangle]_{i,j=0}^m.$$

Contrary to the matrix Q , the approximation of B with quadrature rules only costs $O(m^2)$ operations. Again we could calculate it exactly.

If $v \in X_m$, then $Q_m T v$ may be identified with the vector z solving the linear system $B v = H z$. This leads to the following algorithm:

ALGORITHM 5.2. (augmented GMRES and RRGMRES.)

Let $\delta > 0$, $\tau > 1$, $\varepsilon > 0$ be a small tolerance, and set $\text{gm}:=\text{true}$ if GMRES and $\text{gm}:=\text{false}$ if RRGMRES. Moreover, let H be defined as in (5.1), and let B be as in (5.3).

```

1: for ( $j = 1, \dots, n$ ) {
2:    $\bar{w}_j := [\langle Tw_j, \varphi_i \rangle]_{i=0, \dots, m}$ ;
3:   Solve  $H z = \bar{w}_j$ ;
4:   for ( $i = 1, \dots, j$ ) {  $A_{ij} := \bar{w}_i^\top z$ ; }
5:    $b_j := (y^\delta)^\top z$ ;
6: }
7: Solve  $H z = y^\delta$ ;
8:  $r := \sqrt{(y^\delta)^\top z}$ ;  $k := 0$ ;
9: if (gm) {
10:   $h := r$ ;  $\xi_1 := r$ ;
11:  for ( $j = 1, \dots, n$ ) {
12:     $G_{1j} := b_j/h$ ;
13:    for ( $i = 1, \dots, j$ ) {  $A_{ij} := A_{ij} - G_{1i}G_{1j}$ ; }
14:  }
15: } else {
16:   $w := Bz$ ; Solve  $H z = w$ ;  $h = \sqrt{z^\top w}$ ;  $\xi_1 = (y^\delta)^\top z/h$ ;  $u := r^2$ ;
17:  for ( $j = 1, \dots, n$ ) {
18:     $G_{1j} := z^\top \bar{w}_j/h$ ;  $b_j := b_j - G_{1j}\xi_1$ ;
19:    for ( $i = 1, \dots, j$ ) {  $A_{ij} := A_{ij} - G_{1i}G_{1j}$ ; }
20:  }
21: }
22: while ( $r \geq \tau\delta$ ) {
23:   $k := k + 1$ ;  $v_k = z/h$ ;  $w := Bv_k$ ; Solve  $H z = w$ ;
24:  for ( $i = 1, \dots, k$ ): {
25:     $R_{ik} := w^\top v_i$ ;  $z := z - R_{ik}v_i$ ;
26:  }
27:  for ( $i = 1, \dots, k - 1$ ): {
28:     $\beta := R_{i+1,k}$ ;  $R_{i+1,k} := c_i\beta - s_iR_{ik}$ ;  $R_{ik} := c_iR_{ik} + s_i\beta$ ;
29:  }
30:   $h := z^\top w$ ;
31:  if ( $h < \varepsilon$ ) {
32:     $r := 0$ ;
33:    if ( $|R_{kk}| < \varepsilon$ ) {  $k = k - 1$ ;  $\eta := 0$ ; goto 102; }
34:  } else {
35:     $h := \sqrt{h}$ ;
36:    if ( $\neg\text{gm}$ ) {  $\xi_{k+1} = (y^\delta)^\top z/h$ ; }
37:    for ( $j = 1, \dots, n$ ) {
38:       $G_{k+1,j} := z^\top \bar{w}_j/h$ ; if ( $\neg\text{gm}$ ) {  $b_j := b_j - G_{k+1,j}\xi_{k+1}$ ; }
39:      for ( $i = 1, \dots, j$ ) {  $A_{ij} := A_{ij} - G_{k+1,i}G_{k+1,j}$ ; }
40:    }
41:     $\beta := \sqrt{R_{kk}^2 + h^2}$ ;  $c_k := R_{kk}/\beta$ ;  $s_k := h/\beta$ ;  $R_{kk} := \beta$ ;
42:    for ( $j = 1, \dots, n$ ) {
43:       $\beta := G_{k+1,j}$ ;  $G_{k+1,j} := c_k\beta - s_kG_{k,j}$ ;  $G_{k,j} := c_kG_{k,j} + s_k\beta$ ;
44:    }
45:    if (gm) {
46:       $\xi_{k+1} := -s_k\xi_k$ ;  $\xi_k := c_k\xi_k$ ;

```

```

47:   } else {
48:      $\beta := \xi_{k+1}; \xi_{k+1} := c_k \beta - s_k \xi_k; \xi_k := c_k \xi_k + s_k \beta; u := u - \xi_k^2;$ 
49:   }
50:   for ( $i = 1, \dots, n$ ) {
51:      $\beta := A_{ii};$ 
52:     for ( $l = 1, \dots, i - 1$ ) {  $\beta := \beta - \bar{G}_{li}^2;$  }
53:     if ( $\beta < \varepsilon$ ) {
54:       drop column  $i$ ;
55:     } else {
56:        $\bar{G}_{ii} := \sqrt{\beta};$ 
57:       for ( $j = i + 1, \dots, n$ ) {
58:          $\beta := A_{ij};$ 
59:         for ( $l = 1, \dots, i - 1$ ) {  $\beta := \beta - \bar{G}_{li} \bar{G}_{lj};$  }
60:          $\bar{G}_{ij} := \beta / \bar{G}_{ii};$ 
61:       }
62:       if (gm) {
63:          $\beta := b_i;$ 
64:         for ( $l = 1, \dots, i - 1$ ) {  $\beta := \beta - \bar{G}_{li} \bar{y}_l;$  }
65:          $\bar{y}_i := \beta / \bar{G}_{ii};$ 
66:       }
67:     }
68:   }
69:    $\kappa_1 := \sqrt{G_{k+1,1}^2 + \bar{G}_{11}^2}; c := G_{k+1,1} / \kappa_1; s := \bar{G}_{11} / \kappa_1;$ 
70:   for ( $j = 2, \dots, n$ ) {  $\kappa_j := c G_{k+1,j} + s \bar{G}_{1j}; \bar{G}_{1j} := c \bar{G}_{1j} - s G_{k+1,j};$  }
71:   if (gm) {
72:      $\kappa_{n+1} := c \xi_{k+1}; \bar{y}_1 := -s \xi_{k+1};$ 
73:   } else {
74:      $\kappa_{n+1} := c \xi_{k+1} + s \bar{y}_1; \bar{y}_1 := c \bar{y}_1 - s \xi_{k+1}; \bar{u} := u - \kappa_{n+1}^2;$ 
75:   }
76:   for ( $i = 2, \dots, n$ ) {
77:      $\beta := \sqrt{\bar{G}_{i-1,i}^2 + \bar{G}_{ii}^2}; c := \bar{G}_{i-1,i} / \beta; s := \bar{G}_{ii} / \beta; \bar{G}_{i-1,i} := \beta;$ 
78:     for ( $j = i + 1, \dots, n$ ) {
79:        $\beta := \bar{G}_{i-1,j}; \bar{G}_{i-1,j} := c \beta + s \bar{G}_{ij}; \bar{G}_{ij} := c \bar{G}_{ij} - s \beta;$ 
80:     }
81:     if (gm) {
82:        $\bar{y}_i := -s \bar{y}_{i-1}; \bar{y}_{i-1} := c \bar{y}_{i-1};$ 
83:     } else {
84:        $\beta := \bar{y}_{i-1}; \bar{y}_{i-1} := c \beta + s \bar{y}_i; \bar{y}_i := c \bar{y}_i - s \beta; \bar{u} := \bar{u} - \bar{y}_{i-1}^2;$ 
85:     }
86:   }
87:   if (gm) {
88:      $r := |\bar{y}_n|;$ 
89:   } else {
90:      $r := \sqrt{\bar{u}};$ 
91:   }
92: }
93: }
94: for ( $j = n, \dots, 2$ ): {

```

```

95:   $\eta_j = \bar{y}_{j-1}$ ;
96:  for  $(l = j + 1, \dots, n)$  {  $\eta_j := \eta_j - \bar{G}_{j-1,l}\eta_l$  }
97:   $\eta_j := \eta_j / \bar{G}_{j-1,j}$ ;
98:  }
99:   $\eta_1 = \kappa_{n+1}$ ;
100: for  $(l = 2, \dots, n)$  {  $\eta_1 := \eta_1 - \kappa_l\eta_l$  }
101:  $\eta_1 := \eta_1 / \kappa_1$ ;
102: for  $(i = k, \dots, 1)$ : {
103:   for  $(l = 1, \dots, n)$  {  $\xi_i := \xi_i - G_{il}\eta_l$  };
104:   for  $(l = i + 1, \dots, k)$  {  $\xi_i := \xi_i - R_{il}\xi_l$  };
105:    $\xi_i := \xi_i / R_{ii}$ ;
106: }
107:  $x_k^\delta := \sum_{i=1}^k \xi_i v_i + \sum_{j=1}^n \eta_j w_j$ ;

```

It turns out that, due to numerical errors, one should include a reorthogonalization step after line 26 of the algorithm above, especially for RRGMRES; see, e.g., [1]. An implementation of RRGMRES that does not require reorthogonalization is described in [12]. The implementations there are particularly well suited for the use with the discrepancy principle.

6. Numerical results. In our numerical examples, we choose kernels

$$k(s, t) := \begin{cases} (s - t)^p, & t \leq s, \\ 0, & t > s, \end{cases} \quad p \in \mathbb{N}.$$

The corresponding Volterra integral operator T is not selfadjoint,

$$\mathcal{R}(T) = \{y \in H^{p+1}[0, 1] : y^{(l)}(0) = 0, l = 0, \dots, p\}.$$

The problem of solving $Tx = y$ is modestly ill-posed. Note that

$$Tx = y \Leftrightarrow y^{(p+1)} = p!x.$$

We present examples for $p = 0, 1, 4$. In all cases we test discontinuous solutions

$$x^\dagger(t) := \begin{cases} 1 - \frac{t}{a}, & 0 \leq t \leq a, \\ \frac{1}{2} + \frac{t-a}{2(1-a)}, & a < t \leq 1. \end{cases}$$

Here a is the location of the discontinuity. In one case we assume that $a = \frac{1}{2}$ and in another one that $a = \frac{11}{20}$. For both solutions we chose the following augmented functions:

$$w_1(t) := \begin{cases} 1, & 0 \leq t \leq \frac{1}{2}, \\ 0, & \frac{1}{2} < t \leq 1, \end{cases} \quad w_2(t) := \begin{cases} 0, & 0 \leq t \leq \frac{1}{2}, \\ 1, & \frac{1}{2} < t \leq 1. \end{cases}$$

This means that for $a = \frac{1}{2}$ one guesses the location of the discontinuity correctly, but for the other a value, one is only close to it.

For all examples, the exact right-hand sides y , the functions Tw_j , and the matrices B and Q were calculated exactly.

Since the results for GMRES and RRGMRES were almost identical, we only present results for CGNE and GMRES.

For all cases, we computed x_k^δ for $m = 20$ and $m = 200$. The noise was chosen smaller for larger m and larger p to still obtain useful results. In the stopping rule, $\tau = 1.1$. All figures display the graphs of the exact solution in black and of the final iterates $x_{k_*^\delta}^\delta$ stopped according to the discrepancy principle (see Algorithms 5.1 and 5.2); the iterations with augmentation are displayed in red and those without augmentation in blue.

The results for the case $p = 0$ are given in Table 7.1 and Figures 7.1 and 7.2: δ -% means percent of $\|y\|_X$, C stands for CGNE, aC for augmented CGNE, G for GMRES, and aG for augmented GMRES.

The results for the case $p = 1$ are presented in Table 7.2 and Figures 7.3 and 7.4. The results for GMRES without augmentation are complete nonsense and are, therefore, omitted. The reason is that the matrix B is already almost singular, and we know from Section 3 that no convergence may be expected in this case. Luckily, augmentation fixes the problem.

The results for the case $p = 4$ are presented in Table 7.3 and Figures 7.5. In that case even augmented GMRES does not yield any useful results. Therefore, only the results for CGNE are shown, which are much better for $m = 200$ than for $m = 20$.

7. Conclusion. The numerical results show that for Volterra integral equations with discontinuous solutions, the GMRES and the RRGMRES (even with reorthogonalization) methods implemented as in Algorithm 5.2 should only be used when the matrix is not too close to a singular one. That means it can be used only for very moderately ill-posed problems. The augmentation of the GMRES method improves the results quite a bit.

The advantage of the CGNE method is that it also converges for inverse problems with a higher degree of ill-posedness. Also for the CGNE method, augmentation improves the results. In all cases the results were better for the CGNE method than for the GMRES method.

The number of CGNE iterations is about half of the number of GMRES iterations. Since the CGNE method needs two matrix vector multiplications with the matrix Q in each iteration step while the GMRES method only needs one with the matrix B , the numerical effort for computing the regularized solutions is almost the same. As mentioned in Section 5, the effort for the compilation of the matrix Q is higher than the one for B if the integrals involved have to be approximated by quadrature rules.

We want to mention that one can improve the implementations of GMRES and RRGMRES using preconditioning so that the results are compatible to CGNE and sometimes yield even better results. In [2], GMRES is used for deconvolution in image restoration. It turns out that GMRES compared to CGNE requires less computational work and gives restored images of higher quality. In [8], GMRES converges well for the test problem baart from Hansen's Regularization Tools, which is a severely ill-posed problem. In [6] it was shown that right preconditioning in conjunction with methods based on the Arnoldi process are found to be robust and give high-quality restorations. In [10], the authors try to shed some light on reasons for the poor performance of GMRES in certain situations, especially discretizations of linear ill-posed problems, and discusses some remedies based on specific kinds of preconditioning.

Nevertheless, whether one uses GMRES with preconditioning or CGNE, sometimes CGNE and sometimes GMRES yields better results. Unfortunately, no rule was yet found that allows to decide for which classes of problems one can expect GMRES-like methods to perform better or worse than CGNE.

Acknowledgments. The author wants to thank the referees for comments and for pointing out several new references concerning GMRES methods with preconditioning. He also wants to thank Lothar Reichel for stimulating discussions and providing a copy of the new paper [10].

TABLE 7.1
Case $p = 0$.

a	m	δ -%	k_*^δ -C	k_*^δ -aC	k_*^δ -G	k_*^δ -aG
1/2	20	1	5	3	9	4
1/2	200	0.1	15	14	25	14
11/20	20	1	5	5	9	7
11/20	200	0.1	16	16	24	21

TABLE 7.2
Case $p = 1$.

a	m	δ -%	k_*^δ -C	k_*^δ -aC	k_*^δ -G	k_*^δ -aG
1/2	20	0.1	9	4	19	3
1/2	200	0.1	11	4	46	7
11/20	20	0.1	9	8	19	7
11/20	200	0.1	9	8	36	8

TABLE 7.3
Case $p = 4$.

a	m	δ -%	k_*^δ -C	k_*^δ -aC	k_*^δ -G	k_*^δ -aG
1/2	20	0.01	6	5	10	8
1/2	200	0.001	15	5	21	21
11/20	20	0.01	6	2	12	7
11/20	200	0.001	15	9	12	12

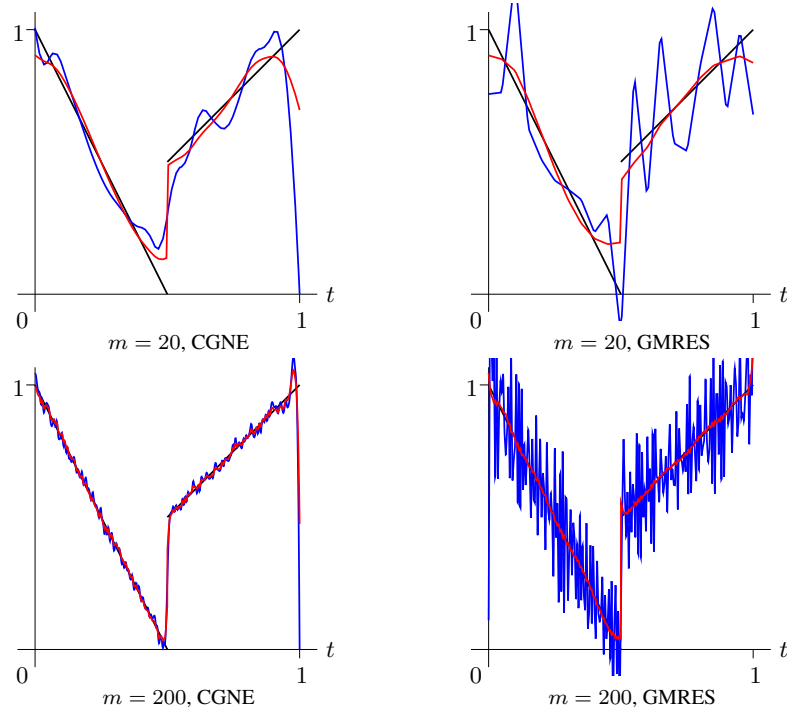


FIG. 7.1. Case $p = 0$, $a = 1/2$.

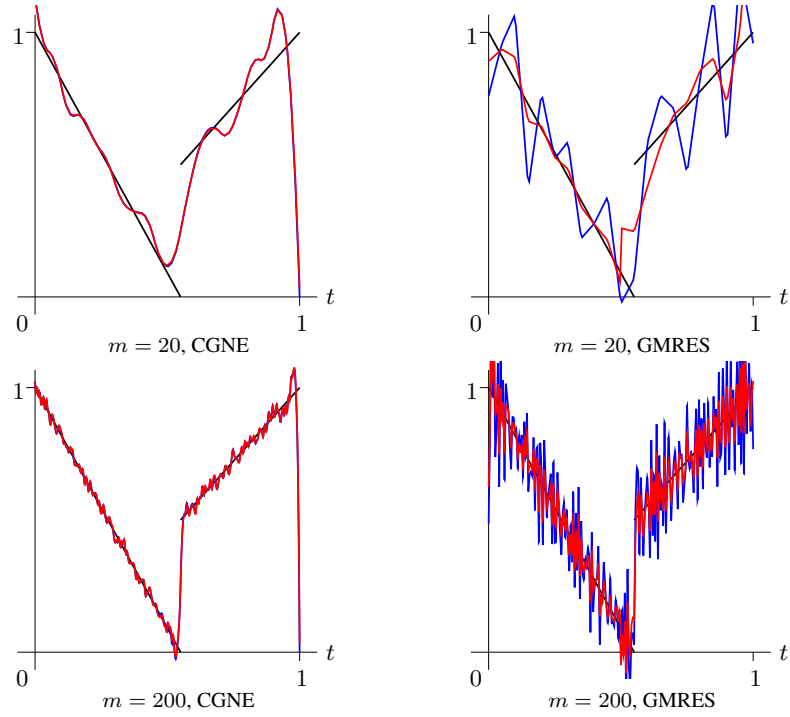


FIG. 7.2. Case $p = 0, a = 11/20$.

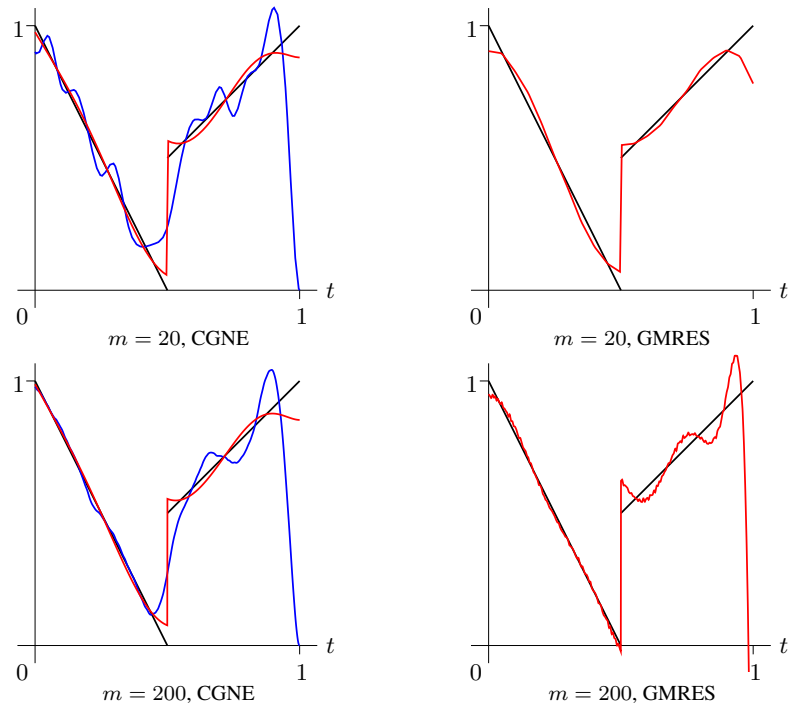


FIG. 7.3. Case $p = 1, a = 1/2$.

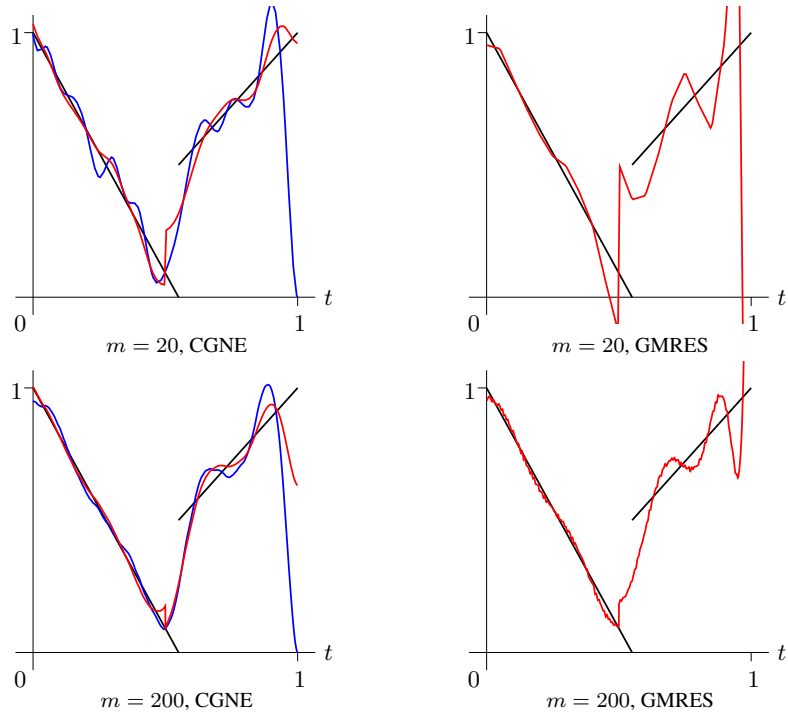


FIG. 7.4. Case $p = 1$, $a = 11/20$.

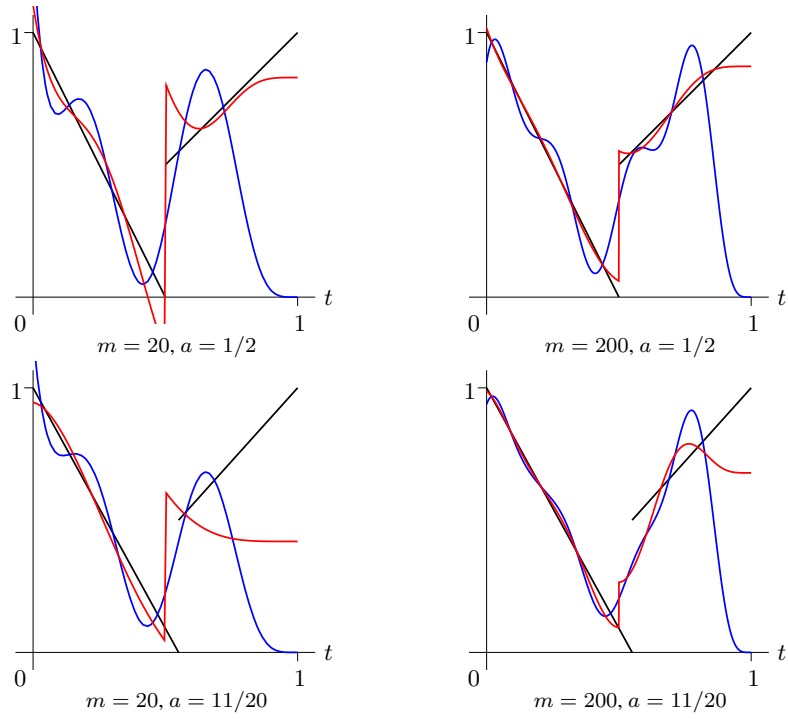


FIG. 7.5. Case $p = 4$, CGNE.

REFERENCES

- [1] J. BAGLAMA AND L. REICHEL, *Augmented GMRES-type methods*, Numer. Linear Algebra Appl., 14 (2007), pp. 337–350.
- [2] D. CALVETTI, B. LEWIS, AND L. REICHEL, *Restoration of images with spatially variant blur by the GMRES method*, in Advanced Signal Processing Algorithms, Architectures, and Implementations X., F. T. Luk, ed., Proceedings of SPIE 4116, SPIE, Bellingham, 2000, pp. 385–395.
- [3] ———, *On the choice of subspace for iterative methods for linear discrete ill-posed problems*, Int. J. Appl. Math. Comput. Sci., 11 (2001), pp. 1069–1092.
- [4] ———, *On the regularizing properties of the GMRES method*, Numer. Math., 91 (2002), pp. 605–625.
- [5] D. CALVETTI, L. REICHEL, AND A. SHUIBI, *Enriched Krylov subspace methods for ill-posed problems*, Linear Algebra Appl., 362 (2003), pp. 257–273.
- [6] M. DONATELLI, D. MARTIN, AND L. REICHEL, *Arnoldi methods for image deblurring with anti-reflective boundary conditions*, Appl. Math. Comput., 253 (2015), pp. 135–150.
- [7] Y. DONG, H. GARDE, AND P. C. HANSEN, *R^3 GMRES: Including prior information in GMRES-type methods for discrete inverse problems*, Electron. Trans. Numer. Anal., 42 (2014), pp. 136–146.
<http://etna.ricam.oeaw.ac.at/vol.42.2014/pp136-146.dir/pp136-146.pdf>
- [8] L. ELDEN AND V. SIMONCINI, *Solving ill-posed linear systems with GMRES and a singular preconditioner*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1369–1394.
- [9] H. W. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 1996.
- [10] S. GAZZOLA, S. NOSCHESI, P. NOVATI, AND L. REICHEL, *Arnoldi decomposition, GMRES, and preconditioning for linear discrete ill-posed problems*, Appl. Numer. Math., 142 (2019), pp. 102–121.
- [11] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.
- [12] A. NEUMAN, L. REICHEL, AND H. SADOK, *Implementations of range restricted iterative methods for linear discrete ill-posed problems*, Linear Algebra Appl., 436 (2012), pp. 3974–3990.
- [13] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.