AN INNER-OUTER REGULARIZING METHOD
FOR ILL-POSED PROBLEMS

PAOLA FAVATI
IIT - CNR Via G. Moruzzi 1, 56124 Pisa, Italy

GRAZIA LOTTI
Dipart. di Matematica e Informatica, University of Parma
Viale G. Usberti 53/A, 43100 Parma, Italy

ORNELLA MENCHI AND FRANCESCO ROMANI
Dipart. di Informatica, University of Pisa
Largo B. Pontecorvo 3, 56127 Pisa, Italy

(Communicated by Jun Zou)

ABSTRACT. Conjugate Gradient is widely used as a regularizing technique for solving linear systems with ill-conditioned coefficient matrix and right-hand side vector perturbed by noise. It enjoys a good convergence rate and computes quickly an iterate, say $x_{k_{\text{opt}}}$, which minimizes the error with respect to the exact solution. This behavior can be a disadvantage in the regularization context, because also the high-frequency components of the noise enter quickly the computed solution, leading to a difficult detection of $k_{\text{opt}}$ and to a sharp increase of the error after the $k_{\text{opt}}$th iteration. In this paper we propose an inner-outer algorithm based on a sequence of restarted Conjugate Gradients, with the aim of overcoming this drawback. A numerical experimentation validates the effectiveness of the proposed algorithm.

1. Introduction. Given a matrix $A \in \mathbb{R}^{n \times n}$ and a vector $b \in \mathbb{R}^n$, we consider the system

$$Ax = b.$$ (1)

In many applications the available right-hand side of the system is contaminated by a noise $\eta$, i.e. $b = b^* + \eta$. The vectors $b^*$ and $x^*$, such that $Ax^* = b^*$, are considered the exact right-hand side and the exact solution of the system. Classical examples are problems arising from the discretization of Fredholm integral equations of the first kind [7], as for instance in the imaging deconvolution.

We assume $A$ to be a large ill-conditioned matrix, with singular values gradually decaying to zero. Because of the presence of $\eta$, here assumed to be an uncorrelated Gaussian white noise with distribution $\mathcal{N}(0, \sigma^2 I)$, the solution $\bar{x} = A^T b$ is often a poor approximation of $x^*$ even if the magnitude of $\eta$ is small. Better approximations to $x^*$ than $\bar{x}$ can be found by looking for regularized solutions of (1). Special regularization techniques have been devised to deal with this kind of problems exploiting both direct methods (as Tikhonov method) and iterative methods [1, 7]. The iterative methods, which are suggested for large matrices $A$ without particular

2010 Mathematics Subject Classification. 65F22, 65F10.

Key words and phrases. Regularization problems, iterative methods, conjugate gradient, generalized cross validation, inner-outer algorithms.
structure properties, must enjoy the semi-convergence property, i.e. in presence of the noise they must reconstruct first the low-frequency components of the solution, in such a way that the iteration can be stopped before the high-frequency components of the noise start to enter the computed solution. Among the classical semi-convergent methods a major role is played by the conjugate gradient method (CG), whose regularizing properties are well known (see for example [6, 7, 11]). When $A$ is not a symmetric positive definite matrix, CG can be applied to the normal equations

$$A^T A x = A^T b.$$  

The optimal number of iterations of CG is very sensitive to the perturbation of the right-hand side [2] and often the noise enters very quickly in the iterates following the optimal one. As a consequence, the regularizing efficiency of CG depends heavily on the effectiveness of the stopping rule employed. The discrepancy principle, widely used as a stopping rule, does not produce sufficiently accurate estimates of the stopping index. The use of the Generalized Cross Validation rule (GCV) [5, 12, 14] gives better results, even if in some cases it is not fully reliable.

In this paper we propose an inner-outer algorithm based on both Tikhonov method and CG, with the aim of overcoming the problems related to the detection of the best regularizing Tikhonov parameter and to the too fast entry of the noise after the optimal iteration of CG. This algorithm consists of two loops: in the outer loop a decreasing sequence of regularization parameters $\lambda_h$, $h \geq 0$, is generated. The aim of the $h$th outer step is to compute a regularizing solution of the system

$$(A^T A + \lambda_h I)x = A^T b,$$  

which minimizes the norm of the error with respect to $x^*$. Namely, at the $h$th outer step a restarted CG is applied to system (3), starting with the last computed vector of the previous outer step until the stopping rule based on GCV is satisfied. The choice of the sequence $\lambda_h$ may be critical. We suggest a geometrically decreasing sequence suitably stopped. More precisely the outer loop is stopped when the results of two subsequent outer steps are equal. The proposed approach gains in reliability at the price of a computational cost greater than the cost of CG applied directly to (2).

A similar approach has been proposed in [15] for estimating the Tikhonov parameter. Indeed the method of [15] consists of a restarted CG applied to a sequence of Tikhonov equations. The main differences reside in the stopping criteria both for the inner and the outer loop. Namely in [15] the inner loop goes on until convergence is achieved and the outer loop is stopped by using the damped Morozov’s discrepancy. Also in [17] an algorithm with a similar structure has been described which produces a sequence of couples of approximate solutions and regularization parameters with the aim of minimizing the value of the Tikhonov functional instead of the norm of the error with respect to $x^*$. Other approaches to adaptively generate a sequence of Tikhonov regularization parameters have been studied in literature. In [10, 16, 9] (see also the references therein) improved versions of the model function method, based on the damped Morozov’s discrepancy, have been introduced. Unlike the methods based on the Morozov’s discrepancy, our algorithm does not require the knowledge of the noise level.

The outline of our paper is the following: first, in Section 2 we compare the behaviors of Tikhonov method and CG. In Sections 3 and 4 we recall the properties...
of CG as a regularization method and of GCV as a stopping rule for CG. In Sections 5 the proposed inner-outer method is described and a sketch of its code is given. Finally, in Section 6 the results of an experimentation on the collection of test problems by Hansen [8] are presented and analyzed.

**Notation.** Throughout the paper, $\|v\|$ denotes the Euclidean norm of a vector $v$, i.e. $\|v\|^2 = v^Tv$.

2. Regularization. Let $A = U\Sigma V^T$ be the singular value decomposition of $A$, where $U = [u_1, \ldots, u_n] \in \mathbb{R}^{n \times n}$ and $V = [v_1, \ldots, v_n] \in \mathbb{R}^{n \times n}$ have orthonormal columns, and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$, where the $\sigma_i$ for $i = 1, \ldots, n$ are the singular values of $A$. We have assumed that the $\sigma_i$ gradually decay toward zero, but in practice this means that they settle to values of the same magnitude of the machine precision. The condition number of $A$, given by $\kappa = \sigma_1/\sigma_n$ is very large.

We consider the expansions of $b^*$ and $\eta$ in the basis $U$

$$b^* = \sum_i b_i^* u_i, \quad \eta = \sum_i \eta_i u_i,$$

where $b_i^* = u_i^T b^*$, $\eta_i = u_i^T \eta$, and the expansions of $x^*$ in the basis $V$

$$x^* = \sum_i x_i^* v_i, \quad \bar{x} = \sum_i \bar{x}_i v_i,$$

where $x_i^* = b_i^*/\sigma_i$, $\bar{x}_i = x_i^* + \eta_i/\sigma_i$.

The vectors $v_i$ are associated to frequencies which increase with $i$. Typically, the low-frequency components describe the overall features of $x^*$ and the high-frequency components represent its details.

The high-frequency components of $\bar{x}$ can be dominated by the high-frequency components of the noise. An acceptable approximation can be obtained only if the quantities $|u_i^T b^*|$ decay to zero with $i$ faster than the corresponding $\sigma_i$ (this condition is known as Picard discrete condition), at least until the numerical levelling of the singular values.

The prototype of direct regularization methods is Tikhonov method, which solves a penalized least-squares problem of the form

$$\min_x \phi(x, \lambda), \quad \text{with} \quad \phi(x, \lambda) = \|b - Ax\|^2 + \lambda \left(\|L x\|^2 - \gamma^2\right),$$

where $\lambda$, called the regularization parameter, is a suitable positive scalar, $L$ is a well-conditioned matrix often related to the discretization of a differential operator and $\gamma$ is related to the level of the solution smoothness. The assumption that $\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}$ guarantees that (5) has a unique solution. To avoid an excessive computational effort, the structure of $L$ should be fairly simple, for example with a small bandwidth, or even diagonal. If $L = I$ a solution of (5) having minimal norm is sought and the problem is said to be in standard form. We consider here the problem in standard form with $\gamma = 0$, which is equivalent to solving the normal equations (3).

Let $x^{(\lambda)}$ be the solution of (3). Tikhonov method aims at detecting the value $\lambda_T$ which minimizes the error $E_T^{(\lambda)} = \|x^* - x^{(\lambda)}\|$. The solution $x^{(\lambda)}$ is given by

$$x^{(\lambda)} = \sum_{i=1}^n x_i^{(\lambda)} v_i = \sum_{i=1}^n \varphi_i^{(\lambda)} \bar{x}_i v_i, \quad \text{with} \quad \varphi_i^{(\lambda)} = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}.$$
It follows that \( \lambda_T \) is linked to the singular values of \( A \) and to the magnitude \( \| \eta \| \) of the noise. This fact can be easily seen assuming the coefficients \( \eta_i \) to be all of the same order, i.e. \( |\eta_i| \sim \| \eta \| / \sqrt{n} \) for \( i = 1, \ldots, n \). In this case, from (4) and (6) it follows that \( \lambda_T \sim \sigma_k^2 \), where the index \( k \) is such that \( b_k^2 \sim \| \eta \| / \sqrt{n} \). Of course, finding \( \lambda_T \) in the case of a large matrix \( A \) is not feasible. Then, to detect a sufficiently good approximation of \( \lambda_T \), one should solve (3) for many values of \( \lambda \) and have a reliable estimator of the error of \( x^{(\lambda)} \) with respect to the unknown \( x^* \).

A different approach consists in applying to system (2) an iterative semi-convergent method to compute a sequence \( x_k \), \( k = 1, 2, \ldots \), stopping the iteration before the high-frequency components of the noise start to enter the computed solution. In this sense the iteration index \( k \) plays the role of a regularization parameter. We consider here the conjugate gradient method (CG) for its good convergence rate. Let \( k_{opt} \) be the optimal index, i.e. the index satisfying

\[
\| x_{k_{opt}} - x^* \| = \min_k \| x_k - x^* \|.
\]

CG converges quickly, but this behavior can be a disadvantage in the regularization context, because also the high-frequency components enter quickly the computed solution, leading to a difficult detection of \( k_{opt} \) and to a sharp increase of the error after the \( k_{opt} \)th iteration.

At this point, a natural question arises: what happens if CG is applied to system (3) with \( \lambda \) varying on the set of nonnegative real numbers? Let \( x_{k}^{(\lambda)} \) be the sequence computed by CG applied to (3) for a selected value of \( \lambda \) and let \( k_{opt}^{(\lambda)} \) be the index minimizing \( \| x_{k}^{(\lambda)} - x^* \| \). For \( \lambda = 0 \) the sequence \( x_k^{(0)} \) is the one computed by CG for system (2), hence \( k_{opt}^{(0)} = k_{opt} \). For \( \lambda > 0 \) the sequence \( x_{k}^{(\lambda)} \) would converge to the Tikhonov solution \( x_{\infty}^{(\lambda)} = x^{(\lambda)} \). Due to the double dependence on \( k \) and \( \lambda \), it follows that

\[
\min_{k, \lambda} \| x_{k}^{(\lambda)} - x^* \| \leq \left\{ \begin{array}{ll}
\min_{\lambda} \| x_{\infty}^{(\lambda)} - x^* \|, \\
\min_{k} \| x_{k}^{(0)} - x^* \|.
\end{array} \right.
\]

To see what happen, we consider a problem randomly generated following the suggestions given in [7], Section 4.7. Matrix \( A \) has size \( n = 64 \), condition number \( \kappa = 10^{20} \), with singular values decaying as \( \sigma_i \approx 0.7^i \). A white Gaussian noise \( \eta \) of magnitude 0.01 is added to the right-hand side \( b^* \). We apply Tikhonov method to this problem with \( \lambda \) varying in the interval \([0.001, 0.04]\) and for each \( \lambda \) we compute the error \( E_T^{(\lambda)} \) (black points in Fig. 1 left). Then we apply CG to system (3) starting with \( x_0 = 0 \) and compute the errors \( E_{k_{opt}}^{(\lambda)} = \| x_{k_{opt}}^{(\lambda)} - x^* \| \) (gray points in Fig. 1 left).

Both the error functions decrease until their minimum points \( \lambda_T \) of \( E_T^{(\lambda)} \) and \( \lambda_C \) of \( E_{k_{opt}}^{(\lambda)} \), then increase. The two minimum points differ and \( \lambda_C < \lambda_T \) (for an explanation of this fact, see Section 3). As expected, the best \( E_{k_{opt}}^{(\lambda)} \) is smaller than \( E_T^{(\lambda)} \), meaning that a better result could be obtained by applying CG to (3) if the correct value \( \lambda_C \) was known and a good stopping rule was available. This fact appears interesting, but we think that an equally important issue concerns the error behavior of CG applied to (3) varying \( \lambda \). The better conditioning of \( A^T A + \lambda I \) with respect to \( A^T A \) slows down the increase of the error after the \( k_{opt} \)th iteration. Fig. 1 (on the right) shows the error histories \( E_k^{(0)} = \| x^* - x_k^{(0)} \| \) (dashed line), and
Figure 1. On the left: errors of Tikhonov method (black points) and of CG applied to system (3) (grey points). On the right: error history of CG applied to system (3) with $\lambda = 0$ (dashed line) and with $\lambda = \lambda_C = 0.005$ (solid line).

$$E_k^{(\lambda_C)} = \|x^* - x_k^{(\lambda_C)}\|$$ (solid line). The different behavior of the errors after the minimum is evident: in the first case the error increases sharply, in the second case the error remains nearly constant for a long while. This advantageous behavior is obtained at the expense of a moderate slowing down of the convergence rate.

3. CG applied to system (3). We examine first how CG applies to solve (3) with different values of the parameter $\lambda > 0$. To avoid a too cumbersome notation, we omit an explicit indication of $\lambda$ in the vectors computed by CG. Denoting $M(\lambda) = A^T A + \lambda I$ and $g = A^T b$, system (3) becomes

(7) $$M^{(\lambda)} x = g.$$ 

Let $x_0$ be a starting point and $q_0 = g - M^{(\lambda)} x_0$ the corresponding residual in system (7). CG is a projection method on the Krylov subspace

$$K_k^{(\lambda)} = \text{span}(q_0, M^{(\lambda)} q_0, (M^{(\lambda)})^2 q_0, \ldots, (M^{(\lambda)})^{k-1} q_0).$$

It builds a sequence of vectors $x_k \in x_0 + K_k^{(\lambda)}$, for $k = 1, 2, \ldots$, such that $x_k - x_{k-1} \in K_k^{(\lambda)}$, and the residual $q_k = g - M^{(\lambda)} x_k$ is orthogonal to $K_k^{(\lambda)}$, i.e. $\langle q_k, w \rangle = 0$, for any $w \in K_k^{(\lambda)}$. In details, setting $p_0 = q_0$, at the $k$th step the method computes

function CGstep($\lambda, k$)
\begin{align*}
\alpha_k &= \|q_k\|^2 / (p_k^T M^{(\lambda)} p_k); \\
x_{k+1} &= x_k + \alpha_k p_k; \\
q_{k+1} &= g - M^{(\lambda)} x_{k+1}; \\
\beta_k &= \|q_{k+1}\|^2 / \|q_k\|^2; \\
p_{k+1} &= q_{k+1} + \beta_k p_k;
\end{align*}

The regular behavior of CG should be monitored, by checking for example breakdowns and evident instability. For the sake of simplicity these controls are not included in the code shown above.

An orthonormal basis $W_k = [w_1, \ldots, w_k]$ of $K_k^{(\lambda)}$ can be constructed by applying $k$ steps of the symmetric Lanczos algorithm to the vector $w_1 = q_0 / \|q_0\|$. The $k \times k$
tridiagonal symmetric matrix
\[ T_k^{(\lambda)} = \begin{bmatrix} \gamma_1 & \delta_2 & & \\ \delta_2 & \gamma_2 & \ddots & \\ & \ddots & \ddots & \delta_k \\ & & \delta_k & \gamma_k \end{bmatrix}, \]
such that \( W_k^T M^{(\lambda)} W_k = T_k^{(\lambda)} \) is produced. The matrix \( T_k^{(\lambda)} \) can be viewed as the representation of \( M^{(\lambda)} \) projected on \( \mathcal{K}_k^{(\lambda)} \). Its entries are \( \gamma_1 = 1/\alpha_0 \) and
\[ \gamma_j = \frac{1}{\alpha_{j-1}} + \frac{\beta_{j-2}}{\alpha_{j-2}}, \quad \delta_j = \frac{\beta_{j-2}}{(\alpha_{j-2})^2}, \quad \text{for } 2 \leq j \leq k. \]
If \( x_0 \) is chosen as the null vector, then \( q_0 = g \) does not depend on \( \lambda \) and
\[ \mathcal{K}_k^{(\lambda)} = \text{span} \left\{ g, (A^T A + \lambda I)g, \ldots, (A^T A + \lambda I)^{k-1}g \right\} \]
\[ = \text{span} \left\{ g, A^T A g, \ldots, (A^T A)^{k-1}g \right\}. \]
Hence \( \mathcal{K}_k^{(\lambda)} \) does not depend on \( \lambda \) and we can drop the superscript. This remains true also if \( x_0 \) is not the null vector, as long as it belongs to \( \mathcal{K}_h \) for a \( h < k \). On the contrary, the matrix \( T_k^{(\lambda)} \) depends on \( \lambda \) and it is easy to see that
\[ T_k^{(\lambda)} = T_k^{(0)} + \lambda I. \]
The eigenvalues \( \theta_{k,j}^{(\lambda)} \) of \( T_k^{(\lambda)} \) are the Ritz values of \( M^{(\lambda)} \) at the \( k \)th iteration. From (10) it follows that
\[ \theta_{k,j}^{(\lambda)} = \theta_{k,j}^{(0)} + \lambda. \]
The regularization properties of CG applied to (7), whose solution is \( x^{(\lambda)} \), can be derived by using the expression of its filter factors given in [7], p.146. If \( x_0 = 0 \),
\[ x_k = \sum_i \psi_{k,i}^{(\lambda)} x_i^{(\lambda)} v_i, \quad \text{where} \quad \psi_{k,i}^{(\lambda)} = 1 - \prod_{j=1}^k \left( 1 - \frac{\sigma_i^2 + \lambda}{\theta_{k,j}^{(\lambda)}} \right), \]
and \( x_i^{(\lambda)} = \varphi_i^{(\lambda)} \tilde{x}_i \) are defined in (6). Then
\[ x_k = \sum_i \psi_{k,i}^{(\lambda)} \varphi_i^{(\lambda)} \tilde{x}_i v_i. \]
In general \( \psi_{k,i}^{(\lambda)} \varphi_i^{(\lambda)} < \varphi_i^{(\lambda)} \). To obtain the same regularizing effect we would get by applying Tikhonov method with the optimal parameter \( \lambda_T \), a value for \( \lambda \) in (12) should verify \( \psi_{k,i}^{(\lambda)} \varphi_i^{(\lambda)} \sim \varphi_i^{(\lambda_T)} \), implying that \( \varphi_i^{(\lambda)} > \varphi_i^{(\lambda_T)} \). It follows that the optimal value \( \lambda_C \) for (12) must verify \( \lambda_C < \lambda_T \). The combined effect of the CG filter factors \( \psi_{k,i}^{(\lambda)} \) and of the Tikhonov filter factors entails a slowing down in the growth of the filter factors of CG applied to system (7), allowing more iterations before the noise sets in, and, more important from our point of view, avoiding that the filter factors corresponding to the smallest \( \sigma_i \) grow too much. Hence more low-frequency components can be reconstructed (see Fig. 1 on the left) and the error growth after the optimal iteration is reduced. The different error history growths for \( \lambda = 0 \) and \( \lambda \neq 0 \) in Fig.1 (on the right) point out this behavior.

**Remark.** From the sequence of parameters \( \alpha_k \) and \( \beta_k \) computed by function CGstep the entries \( \gamma_k \) and \( \delta_k \) of the matrix \( T_k^{(\lambda)} \) are obtained using (8). Low-cost
methods are available for computing the eigenvalues of this symmetric tridiagonal matrix. If we subtract $\lambda$, according to (11), we find the Ritz values of $M^{(0)} = A^T A$.

It is known that in infinite precision arithmetic, when $k$ increases, the Ritz values of the matrix $M^{(0)}$ approximate some eigenvalues of $M^{(0)}$, starting from the largest ones. Unfortunately in finite precision arithmetic, artifacts are generated due to the loss of orthogonality of the vectors $w_j$, $j \leq k$ for large $k$. These spurious Ritz values appear as multiple quasi-approximations of the same eigenvalue of $M^{(0)}$. If our aim is to get some information on the behavior of the largest eigenvalues of $M^{(0)}$ through its Ritz values, the spurious values should be removed. A fit of the remaining values can give an insight on the behavior of the eigenvalues of $M^{(0)}$, which are the singular values $\sigma_i^2$ of $A^T A$.

4. GCV as a stopping rule. The choice of the index at which the iteration should be stopped is critical. Ideally, this should happen when the error $\|x^* - x_k\|$ is minimum, but this error is not directly computable. An acceptable approximation of the optimal index could be sought through the predictive error $\pi_k = Ax^* - Ax_k$, but also this error is not directly computable. However it is possible to get a reliable estimate of $\pi_k$ stochastically by using the Generalized Cross Validation method (GCV) described in [5, 14].

Let $x_k = R_k(b)$ be the vector computed at the $k$th step by an iterative method. The operator $A_k(b) = AR_k(b)$ is called influence operator. For certain iterative regularization methods (for example Landweber method) the operator $A_k$ depends linearly on $b$ and an explicit simple form is available, but this is not the case of CG. To overcome this difficulty, if the function $R_k(b)$ is continuously differentiable in the neighborhood of $b$ (and this is true for CG applied to (7)), in [12] it is suggested to replace $R_k(b)$ by its first order approximation $J_k b$, where $J_k$ is the Jacobian matrix of $x_k$ with respect to $b$, thus

$$x_k \sim J_k b.$$  

With this approximation $Ax_k \sim A_k b$, where $A_k = AJ_k$ is the approximate influence matrix.

The GCV functional for CG is defined as

$$V_k = \frac{n \|r_k\|^2}{\left(\text{tr} (I - A_k)\right)^2},$$  

where $r_k = b - Ax_k$ is the residual of $x_k$ in system (1) and tr($B$) indicates the trace of the matrix $B$. If initially $x_0 = 0$, we have $V_0 = n \|r_0\|^2 / (\text{tr} (I))^2 = \|b\|^2 / n$.

The minimizer $k_V$ of $V_k$ is the GCV estimate for the optimal index. The quantity tr($I - A_k$) in the denominator of (13) is considered as the effective number of degrees of freedom.

To compute $V_k$ a technique for approximating the trace of $A_k$ is required. For large dimensions, the Trace Lemma [13] is generally exploited: if $v$ is any vector of $n$ independent samples of a normal random variable with zero mean and unit variance, then $v^T A_k v$ is an unbiased estimator for tr $A_k$, i.e.

$$\text{tr} A_k \sim v^T A_k v = v^T A J_k v.$$  

For a practical implementation of (14), we need:

(a) one or more vectors $v \sim \mathcal{N}(0, I)$ [13]. According to [4], in most of the current problems, when matrix $A_k$ is symmetric and $n$ is large enough, only one vector $v$ is
needed to get a reliable estimate. From now on, by \( v \) we indicate always a vector of this type.

(b) a way to compute the vector \( s_k = J_k v \). The \((j,h)\)th component of \( J_k \) is given by \( \partial(x_k)_j \) and, following [3] the product \( J_k v \) can be computed recursively differentiating the recursive definition of \( x_k \) given in the function \( \text{CGstep} \). The computation of the quantities

\[
(s_k)_j = \sum_h \partial(x_k)_j \partial(bh) v_h, \quad (u_k)_j = \sum_h \partial(p_k)_j \partial(bh) v_h, \quad (t_k)_j = \sum_h \partial(q_k)_j \partial(bh) v_h,
\]

\[
\zeta_k = \sum_h \partial(\alpha_k) \partial(bh) v_h, \quad \xi_k = \sum_h \partial(\beta_k) \partial(bh) v_h,
\]

at the \( k \)th step is then performed in the following way

\[
\text{function } \text{CGstep}(\lambda, k) \\
\zeta_k = 2 \left( t_k^T q_k - \alpha_k u_k^T M(\lambda) p_k \right) / \left( p_k^T M(\lambda) p_k \right);
\]

\[
s_{k+1} = s_k + \alpha_k u_k + \zeta_k p_k;
\]

\[
t_{k+1} = A^T v - M(\lambda) s_{k+1};
\]

\[
\xi_k = 2 \left( t_{k+1}^T q_{k+1} - \beta_k t_k^T q_k \right) / \left\| q_k \right\|^2;
\]

\[
u_{k+1} = t_{k+1} + \beta_k u_k + \xi_k p_k;
\]

Initially, if \( x_0 = J_0 b \), then \( s_0 = J_0 v \) and \( u_0 = t_0 = A^T v - M(\lambda) s_0 \). This procedure for evaluating \( \text{tr} A_k \) doubles the computational cost of CG in terms of the number of matrix-vector products.

5. The inner-outer algorithm. In Section 2 we have seen that applying CG to system (3) can produce better results than both CG applied to system (2) and Tikhonov method. However this is true only for a limited choice of values for the parameter \( \lambda \). To avoid such a critical issue, we propose the following ACG algorithm:

(1) Initialize \( \lambda_0, x_0^{(\lambda_0)} \) and \( \delta \);

(2) set \( h = 0 \);

(3) while an outer stopping condition is not satisfied do:

(a) compute a regularized solution \( x_k^{(\lambda_h)} \) of system \((A^T A + \lambda_h I)x = A^T b\);

(b) set \( \lambda_{h+1} = \lambda_h \delta \);

(c) set \( h = h + 1 \);

Here are some details about the steps (1) and (3) of ACG.

**Step (1).** We have already noted in Section 2 that the optimal value \( \lambda_T \) for Tikhonov method is not far from a singular value \( \sigma_k^2 \) of \( A^T A \), so we assume that the optimal value for \( \lambda \) when CG is applied to (3) should also be close to a singular value of \( A^T A \). This assumption suggests to consider an exponential decreasing sequence of the form \( \lambda_k = \lambda_0 \delta^k \), with suitable \( \lambda_0 \) and \( \delta < 1 \). At the beginning, we consider \( \lambda_0 = \gamma_1 = \|A g\|^2 / \| g \|^2 \) which is the first value of the sequence given in (8) when the CG is applied to system (2) and represents an approximation to the first eigenvalue of \( A^T A \). For what concerns the value of \( \delta \) which rules the decreasing of...
the parameter $\lambda$, we suggest $\delta = 0.1$ since, in our experimentations, it represents a good tradeoff between the regularization capability and the convergence rate of the algorithm. Other choices for $\lambda_0$ and $\delta$, ruled by information about the spectral structure of the matrix $A^T A$ and the magnitude of $\eta$, could be more suitable. In particular an idea on the decreasing rate of the spectrum of $A^T A$ could be acquired according to the Remark at the end of Section 3.

**Step (3).** To find $x_h^{(\lambda_h)}$ we apply a restarted CG method starting from $x_0^{(\lambda_h)} = x_{h-1}^{(\lambda_{h-1})}$. CG iteration is stopped through a stopping rule (implemented by means of the Boolean `inner_stop_cond` in the following code) based on GCV, i.e. it is stopped when the GCV functional increases, detecting in this way a regularized solution of system (3) for $\lambda = \lambda_h$. It is true that the regularized solutions $x_k^{(\lambda_h)}$, for $h = 0, 1, \ldots$, correspond to different values of the parameter $\lambda$, but as noted in Section 3 they belong all to the same Krylov space $K_k$ defined in (9). For this reason, the sequence obtained by joining the sequences of vectors computed in subsequent steps of the while loop can be seen as produced by a unique iterative projection method. It follows that also the sequences of the GCV functionals $V_k$ and of the derivatives $s_k$ can be linked together. The outer stopping condition is satisfied for the first $h$ such that $k_h = 0$, meaning that two subsequent steps of the while loop produce the same vector, i.e. $x_h^{(\lambda_h)} = x_{h-1}^{(\lambda_{h-1})}$. The condition is monitored in the following code by the Boolean `outer_stop_cond`. The number of iterations is kept under control by means of the condition $h < h_{\text{max}}$, where $h_{\text{max}}$ is a given constant.

To avoid long lists of arguments, all the constructed vectors are assumed to belong to a global working space.

```matlab
function x_reg = ACG(A, b);
g = A^T * b;
v = randn(n, 1); % normally distributed pseudorandom vector
w = A^T * v;
h = 0;
x_0 = s_0 = 0;
V_0 = norm(b)^2 / n;
lambda_0 = norm(A * g)^2 / norm(g)^2; % approximation of $\sigma_1^2$
\delta = 0.1;
outer_stop_cond = true;
while outer_stop_cond
    CGinner(\lambda_h, h);
    outer_stop_cond = (k_h > 0) & (h < h_{\text{max}});
    \lambda_{h+1} = \delta \lambda_h;
    h = h + 1;
end
x_reg = x_0;

function CGinner(\lambda, h) % inner loop, $k_h$ is the GCV minimizer
M^{(\lambda)} = A^T A + \lambda I;
p_0 = q_0 = g - M^{(\lambda)} x_0;
u_0 = t_0 = w - M^{(\lambda)} s_0;
k = 0;
```

---

**AN INNER-OUTER REGULARIZING METHOD FOR ILL-POSED PROBLEMS**

**Inverse Problems and Imaging** Volume 8, No. 2 (2014), 409–420
inner_stop_cond = true;
while inner_stop_cond
    CGstep(λ, k)
    CGVstep(λ, k)
    k = k + 1;
    r_k = b - Ax_k;
    V_k = n\|r_k\|^2/((n - w^T s_k)^2);
    inner_stop_cond = (V_k < V_k-1) & (k < k_{max});
end
k_h = k - 1;
x_0 = x_{k_h};
s_0 = s_{k_h};
V_0 = V_{k_h};

Note that a decreasing sequence of parameters λ_h has been considered also in algorithm Descent-TCG of [17], together with a specific stopping rule, with the aim of minimizing the functional \( \phi(x, λ) \) defined in (5). Both the updating of the parameters and the stopping rule of [17] are not suitable to our algorithm, which aims at minimizing \( \|x^{(λ)}_k - \mathbf{x}^*\| \).

6. Numerical experiments. The numerical experimentation has been carried out in Mathematica with machine-epsilon \( 2^{-53} \) arithmetic on 11 problems of [8] at size \( n = 64 \) and \( n = 256 \). These problems are obtained from the discretization of Fredholm integral equations of the first kind. For each problem the matrix \( A \) and the solution \( \mathbf{x}^* \) were given, then the vector \( \mathbf{b}^* = Ax^* \) was computed and six white Gaussian noises were generated, with relative magnitudes \( NSR = \|\eta\|/\|\mathbf{b}^*\| \) ranging from 0.001 to 0.05. To each sample, i.e. a triple \( (A, \mathbf{b}^*, \eta) \), both simple CG with GCV stopping rule (CG+GCV in the following) and ACG have been applied. The solutions obtained by CG+GCV and by ACG are denoted \( x_{CG+} \) and \( x_{ACG} \) respectively. Also the theoretical optimal solution \( x_{k_{opt}} \) has been computed with CG as a reference. The corresponding errors with respect to the theoretical solution are denoted

\[
E_{k_{opt}} = \|\mathbf{x}^* - x_{k_{opt}}\|, \quad E_{CG+} = \|\mathbf{x}^* - x_{CG+}\|, \quad E_{ACG} = \|\mathbf{x}^* - x_{ACG}\|.
\]

In order to highlight the behavior of ACG, two problems of size \( n = 64 \) have been selected, namely shaw with \( NSR = 0.01 \) and ilaplace with the solution \( \mathbf{x}^* \) corresponding to the forth example and \( NSR = 0.001 \). In Figure 2 we show the error histories for GC (solid and dotted line) and ACG (dashed line) applied to problems shaw (on the left) and ilaplace (on the right). The transition point between solid line and dots indicates where CGV stops the iteration of CG.

For both problems, \( x_{ACG} \) outperforms \( x_{CG+} \) and is even better than \( x_{k_{opt}} \) as can be seen in Table 1.

|          | \( E_{k_{opt}} \) | \( E_{CG+} \) | \( E_{ACG} \) |
|----------|-----------------|----------------|----------------|
| shaw     | 1.054           | 1.054          | 0.879          |
| ilaplace | 0.949           | 1.109          | 0.934          |

Table 1. Errors obtained by applying CG and ACG to the two selected problems.
These problems illustrate two different possible cases where the use of ACG can be worthwhile: in the first case (shaw) the error of CG+GCV is very close to the optimal error $E_{k_{opt}}$, but ACG gets a substantial reduction; in the second case (ilaplace) CG+GCV has a poor performance because GCV does not provide a good stopping index, while ACG reaches an error lower than $E_{k_{opt}}$.

However ACG does not guarantee always an improvement in the accuracy of the approximate solution with respect to CG+GCV. Considering the whole set of test problems, for each sample the quantity

$$e = \frac{E_{ACG}}{E_{CG}} - 1$$

is used as an indicator of the performance of the algorithm. We judge that the two methods produce approximations with comparable accuracy when $|e| \leq \ell$ for $\ell = 0.02$. The behavior of ACG on the two test sets, each consisting of 66 samples, for $n = 64$ and $n = 256$ is shown in Table 2.

| $e < -10\ell$ | $-10\ell \leq e < -\ell$ | $|e| \leq \ell$ | $\ell < e \leq 10\ell$ | $10\ell < e$ |
|---|---|---|---|---|
| $n = 64$ | 19.7% | 30.3% | 27.3% | 15.2% | 7.5% |
| $n = 256$ | 9.2% | 31.8% | 30.3% | 21.2% | 7.5% |

**Table 2.** Analysis of the behavior of ACG compared with CG+GCV on the whole test set for $n = 64$ and $n = 256$.

The columns of the table give the percentages of cases for which ACG strongly outperforms CG+GCV ($e < -10\ell$), outperforms CG+GCV ($-10\ell \leq e < -\ell$), is comparable to CG+GCV, is outperformed by CG+GCV ($\ell < e \leq 10\ell$) and is strongly outperformed by CG+GCV ($10\ell < e$). The above results show that, in a high percentage of cases, ACG produces approximations more accurate than CG+GCV, even if at a greater cost. More precisely the analysis of the computational costs of the two algorithms, in terms of the number of iterations, shows that the average cost of ACG on the two test sets is four/five times that of the usual CG+GCV method. Such multiplicative factor in the cost of ACG is not so heavy when compared to the number of trials required by the most popular methods for the detection of the regularization Tikhonov parameter.

7. **Conclusions.** The regularization problems are intrinsically difficult, because all the heuristics suggested for estimating the optimal regularizing parameter may...
turn out sometimes unreliable. Our proposal joins the advantage of using an iterative method (essential with large dimensions matrices) with a lower criticality in detecting the stopping point.

REFERENCES

[1] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*, Institute of Physics Publishing, Bristol, 1998.

[2] B. Eicke, A. K. Louis and R. Plato, The instability of some gradient methods for ill-posed problems, *Numer. Math.*, 58 (1990), 129–134.

[3] P. Favati, G. Lotti, O. Menchi and F. Romani, Generalized Cross-Validation Applied to Conjugate Gradient for Discrete Ill-Posed Problems, Technical Report IIT TR-09/2013. Available from: http://www.iit.cnr.it/sites/default/files/TR-09-2013.pdf

[4] D. A. Girard, A fast ‘Monte-Carlo Cross-Validation’ procedure for large least squares problems with noisy data, *Numer. Math.*, 56 (1989), 1–23.

[5] G. H. Golub, M. Heath and G. Wahba, Generalized cross-validation as a method for choosing a good ridge parameter, *Technometrics*, 21 (1979), 215–223.

[6] M. Hanke, *Conjugate Gradient Type Methods for Ill-Posed Problems*, Pitman Research Notes in Mathematics, Longman, Harlow, 1995.

[7] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM Monographs on Mathematical Modeling and Computation, Philadelphia, 1998.

[8] P. C. Hansen, Regularization tools, *Numer. Algo.*, 46 (2007), 189–194.

[9] B. Jin, Y. Zhao and J. Zou, Iterative parameter choice by discrepancy principle, *IMA J. Numer. Anal.*, 32 (2012), 1714–1732.

[10] K. Kunisch and J. Zou, Iterative choices of regularization parameters in linear inverse problems, *Inverse Problems*, 14 (1998), 1247–1264.

[11] R. Plato, Optimal algorithms for linear ill-posed problems yield regularization methods, *Num. Funct. Anal. and Optimiz.*, 11 (1990), 111–118.

[12] R. J. Santos and A. R. De Pierro, The effect of the nonlinearity on GCV applied to Conjugate Gradients in computerized tomography, *Comput. Appl. Math.*, 25 (2006), 111–128.

[13] C. Vogel, *Computational Methods for Inverse Problems*, SIAM Frontier in Applied Mathematics, Philadelphia, 2002.

[14] G. Wahba, Practical approximate solutions to linear operator equations when the data are noisy, *SIAM J. Numer. Anal.*, 14 (1977), 651–667.

[15] Y. Wang, A restarted conjugate gradient method for ill-posed problems, *Acta Mathematicae Applicatae Sinica, English Series*, 19 (2003), 31–40.

[16] J. Xie and J. Zou, An improved model function method for choosing regularization parameters in linear inverse problems, *Inverse Problems*, 18 (2002), 631–643.

[17] F. Zama and E. Loli Piccolomini, A descent method for regularization of ill-posed problems, *Optimization Methods and Software*, 20 (2005), 615–628.

Received September 2013; revised February 2014.

E-mail address: paola.favati@iit.cnr.it
E-mail address: grazia.lotti@unipr.it
E-mail address: menchi@di.unipi.it
E-mail address: romani@di.unipi.it