Algorithms for the regularization of ill-conditioned least squares problems

Article *in* BIT. Numerical mathematics · June 1977

DOI: 10.1007/BF01932285

L. Eldén
Linköping University

BIT 17 (1977), 134–145

ALGORITHMS FOR THE REGULARIZATION OF ILL-CONDITIONED LEAST SQUARES PROBLEMS

LARS ELDÉN

Abstract.

Two regularization methods for ill-conditioned least squares problems are studied from the point of view of numerical efficiency. The regularization methods are formulated as quadratically constrained least squares problems, and it is shown that if they are transformed into a certain standard form, very efficient algorithms can be used for their solution. New algorithms are given, both for the transformation and for the regularization methods in standard form. A comparison to previous algorithms is made and it is shown that the overall efficiency (in terms of the number of arithmetic operations) of the new algorithms is better.

1. Introduction.

The problem of solving a Fredholm integral equation of the first kind

$$(1.1) \qquad \int_a^b K(x,y)f(y)dy = g(x), \quad c \leqq x \leqq d, \quad -\infty < a < b < +\infty ,$$

where K is continuous, is ill-posed in the sense that the solution f does not depend continuously on the data g. Equation (1.1) can be discretized in various ways, e.g. by moment discretization (see [14]), or expansion of f in a basis of piece-wise polynomials (see [9]), giving a system of linear equations,

$$(1.2) \qquad Kf = g ,$$

where K is an $m \times n$ matrix and f and g are vectors.

Due to the ill-posedness of (1.1) the condition number of K increases rapidly with n. Therefore any attempt to solve (1.2) e.g. in the least squares sense for large values of n will give a meaningless result, and in this sense the problem of solving (1.2) is ill-posed too. To make it well-posed one can introduce some a priori information about the solution, e.g. in the form of a bound of the norm of Lf, where L is some $p \times n$ matrix. This leads to a constrained least squares problem

$$(1.3) \qquad R1 : \min_{f \in B_1} \|Kf-g\|_2, \quad B_1 = \{f : \|Lf\|_2 \leqq \omega\} .$$

Received Oct. 1. 1976. Revised Feb. 18, 1977.

In practice g often contains measurement errors and then it is not meaningful to try to satisfy (1.2) exactly. If we know that the solution f is smooth in some sense we are led to the following minimization problem

$$R2:\ \min_{f \in B_2} \|Lf\|_2, \quad B_2 = \{f:\ \|Kf-g\|_2 \leqq \varepsilon\}, \tag{1.4}$$

for some value of ε related to the statistical distribution of the errors in g (see [9] and [1]).

In [3] $R1$ and $R2$ are studied and it is proved that in all interesting cases the minima are attained on the boundaries of B_1 and B_2. Using the terminology of Tihonov [17] we call $R1$ and $R2$ *regularization methods* for the ill-conditioned least squares problem $\min \|Kf-g\|_2$. $R1$ is sometimes called the method of quasisolution [10]. $R2$ was suggested by Phillips [15] and Cook [1].

In $R1$ and $R2$ the matrix L is the discretization of a differentiation operator and thus in most cases a band matrix. For simplicity we assume that L has full row rank (i.e. $\operatorname{rank}(L) = p$). This is no restriction, since otherwise L can be transformed into a full row rank matrix by premultiplication by an orthogonal matrix [13], Theorem 3.15. We also assume that the nullspaces of K and L intersect trivially:

$$N(K) \cap N(L) = \{0\}. \tag{1.5}$$

This is a necessary condition for (1.3) and (1.4) to have a unique solution.

Using the method of Lagrange multipliers we see that (1.3) and (1.4) are equivalent to solving (for the multiplier μ, $\mu > 0$) the equations

$$h_1(\mu) = \|Lf_\mu\|_2^2 - \omega^2 = 0, \tag{1.6}$$

$$h_2(\mu) = \|Kf_\mu - g\|_2^2 - \varepsilon^2 = 0, \tag{1.7}$$

respectively, where f_μ is the solution of

$$\min_f \{\|Kf-g\|_2^2 + \mu\|Lf\|_2^2\}. \tag{1.8}$$

Thus we note that the repeated solution of the unconstrained least squares problem (1.8) for different values of μ is common to $R1$ and $R2$. (We also remark that (1.8) in itself has been suggested as a regularization method by Tihonov [17], [18]).

We shall say that $R1$ and $R2$ (and also (1.8)) are in *standard form* if $L = I$.

Although there is an extensive literature about regularization methods, not very much has been written about numerical algorithms for $R1$ and $R2$, except when they are in standard form, ([7], [8]). In this paper we develop efficient algorithms for $R1$ and $R2$ in the general case $L \neq I$.

In section 2 we first show that if a straightforward approach is made then the cost for the repeated solution of (1.8) is prohibitively high. However, for $R1$, $R2$ and (1.8) in standard form there are very efficient algorithms. We give an algorithm for the transformation into standard form, which is based on a decomposition of L. If L is a band matrix the transformation can be computed cheaply.

Previous algorithms ([7], [8]) for $R1$ and $R2$ in standard form use a singular value decomposition (*SVD*) of the matrix K. The *SVD* algorithm starts with a bidiagonalization of K, followed by an iterative reduction to diagonal form. In section 3 we show that the iterative part can be avoided, thus saving the main part of the work in the *SVD* algorithm, and we give algorithms for $R1$ and $R2$ based on the bidiagonalization of K. A comparison is made between our algorithms and those based on the *SVD*, and it is shown that the overall efficiency (in terms of the number of arithmetic operations) of our algorithms is better.

The algorithms presented in this paper are discussed in more detail in [2]. The implementation of the algorithms in a program for interactive regularization will be discussed in a forthcoming report [5], where also Algol procedures will be given.

2. Transformation into standard form.

When an iterative method is used for the solution of the non-linear equations (1.6) and (1.7), the least squares problem (1.8) must be solved for several values of μ. Note that the assumption $N(K) \cap N(L) = \{0\}$ implies that (1.8) has a unique solution for all positive μ. We first consider a straightforward approach for the solution of (1.8), and here we assume that L is a band matrix. It is easily seen that (1.8) is equivalent to

$$\min_f \left\| \begin{pmatrix} R \\ \sqrt{\mu} L \end{pmatrix} f - \begin{pmatrix} g_1 \\ 0 \end{pmatrix} \right\|_2, \tag{2.1}$$

where $K = QR$ is the QR-decomposition of K and $Q^T g = g_1$. By making a further orthogonal transformation we can reduce $\begin{pmatrix} R \\ \sqrt{\mu} L \end{pmatrix}$ to upper triangular form, from which the solution is easily obtained (the details of the procedure can be found in [6], pp 423-424). Even if we take advantage of the band structure of L this reduction requires $O(n^3)$ operations, which is highly unsatisfactory since it must be performed for several values of μ. It seems that we can do no better as long as $L \neq I$, for then only left hand orthogonal transformations can be used for the decomposition of K (transformations from the right would destroy the band structure of L). When $L=I$, (i.e. when (1.8) is in standard form), right hand transformations can be used and in this case there are very efficient algorithms for the repeated solution of (1.8) (see [7], [8] and section 3 of this paper).

We also see that when L is not a band matrix the approach (2.1) is unsatisfactory, because then we must annihilate a $p \times n$ matrix for every value of μ (it does not seem to be possible to exploit any sparseness structure of L).

We now give an algorithm for the reduction of (1.8) into standard form. Problem (1.8) is equivalent to minimizing $\|r\|_2$, where r is the residual vector

$$r = \begin{pmatrix} K \\ \sqrt{\mu}L \end{pmatrix} f - \begin{pmatrix} g \\ 0 \end{pmatrix}. \tag{2.2}$$

We start by making a decomposition of L^T

$$V^T L^T = \begin{pmatrix} R \\ 0 \end{pmatrix} \begin{matrix} \}p \\ \}n-p \end{matrix}, \quad V = (\underbrace{V_1}_{p} \; \underbrace{V_2}_{n-p}), \tag{2.3}$$

where V and R are non-singular (note that we have assumed that L has full row rank (see section 1)). We can take (2.3) to be the QR-decomposition of L^T, in which case V is orthogonal and R upper triangular. If L is a band matrix this decomposition can be computed efficiently by a sequence of plane rotations. Alternatively (2.3) can be computed by Gaussian elimination.

If $n=p$ ($=\operatorname{rank}(L)$) we can now put $f=VR^{-T}\tilde{f}$ and it is easily seen that the problem is in standard form in the variable $\tilde{f}$. Otherwise, if $p<n$, we make the change of variables

$$f = Vy = V_1 y_1 + V_2 y_2, \tag{2.4}$$

by which (2.2) is transformed into

$$r = \begin{pmatrix} KV_1 & KV_2 \\ \sqrt{\mu}R^T & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} g \\ 0 \end{pmatrix}. \tag{2.5}$$

We then perform a QR-decomposition of KV_2,

$$KV_2 = Q \begin{pmatrix} U \\ 0 \end{pmatrix} \begin{matrix} \}n-p \\ \}m-n+p \end{matrix}, \quad Q = (\underbrace{Q_1}_{n-p} \quad \underbrace{Q_2}_{m-n+p}), \tag{2.6}$$

where Q is orthogonal and U is upper triangular. Note that the nullspace of L is spanned by the columns of V_2 and therefore by the assumption (1.5) we must have $\operatorname{rank}(KV_2)=n-p$. This implies that U is nonsingular. By multiplying the first m components in (2.5) by Q^T we get

$$\bar{r} = \begin{pmatrix}\bar{r}_1\\ \bar{r}_2\\ \bar{r}_3\end{pmatrix} = \begin{pmatrix} Q_1{}^TKV_1 & U\\ Q_2{}^TKV_1 & 0\\ \sqrt{\mu}R^T & 0\end{pmatrix}\begin{pmatrix}y_1\\ y_2\end{pmatrix} - \begin{pmatrix}Q_1{}^Tg\\ Q_2{}^Tg\\ 0\end{pmatrix}.$$

Here $\bar{r}_2$ and $\bar{r}_3$ are independent of y_2, and y_2 can always be chosen so that $\bar{r}_1=0$. Therefore the problem (1.8) splits into

$$\min_{y_1}\left\|\begin{pmatrix}Q_2{}^TKV_1\\ \sqrt{\mu}R^T\end{pmatrix} y_1 - \begin{pmatrix}Q_2{}^Tg\\ 0\end{pmatrix}\right\|_2, \tag{2.7}$$

and

$$y_2 = U^{-1}Q_1{}^T(g-KV_1y_1)\,. \tag{2.8}$$

By the final change of variables

$$\tilde{f} = R^Ty_1\,, \tag{2.9}$$

(2.7) is reduced to standard form

$$\min_{\tilde{f}}\,\{\|\tilde{K}\tilde{f}-\tilde{g}\|_2^2+\mu\|\tilde{f}\|_2^2\}\,, \tag{2.10}$$

where

$$\tilde{K} = Q_2{}^TKV_1R^{-T}, \quad \tilde{g} = Q_2{}^Tg, \quad \tilde{K} \text{ is } (m-n+p)\times p\,. \tag{2.11}$$

From (2.4), (2.8) and (2.9) we see that the solutions of (1.8) and (2.10) are related by

$$\begin{cases} \tilde{f} = Lf, & \text{(2.12a)}\\ f = V_1R^{-T}\tilde{f} + V_2U^{-1}Q_1{}^T(g-KV_1R^{-T}\tilde{f}\,)\,. & \text{(2.12b)}\end{cases}$$

If ω and ε are small enough so that the constraints are active, it can be proved (see [3]) that $R1$ and $R2$ are equivalent to

$$\tilde{R}1:\ \min_{\tilde{f}\in\tilde{B}_1}\|\tilde{K}\tilde{f}-\tilde{g}\|_2, \quad \tilde{B}_1 = \{\tilde{f}:\ \|\tilde{f}\|_2 = \omega\}\,,$$
$$\tilde{R}2:\ \min_{\tilde{f}\in\tilde{B}_2}\|\tilde{f}\|_2, \quad \tilde{B}_2 = \{\tilde{f}:\ \|\tilde{K}\tilde{f}-\tilde{g}\|_2 = \varepsilon\}$$

where $\tilde{K}$ and $\tilde{g}$ are defined by (2.11) and the solutions are related by (2.12). Thus, using the above technique, $R1$ and $R2$ are transformed into standard form.

Instead of computing the QR-decomposition (2.6) one can use projections, see [2].

It can be shown that in a certain sense the above transformation is a discrete analogue of the reproducing kernel Hilbert space technique developed by Wahba [22], [23].

The number of operations for the transformation into standard form depends very much on the actual implementation. As an example we assume that $m=n$ and that the bandwidth of L is $q+1$ where $q=n-p$. If the decomposition of L is computed by a series of rotations, then the whole transformation requires approximately $6npq+(q+1)p^2$ operations. The backtransformation (2.12b) can be performed in $6pq+q^2/2$ operations.

Van Loan [19] and Köckler [12] derive a transformation similar to the one described above (see also Varah [20]). Their approach is based on a simultaneous diagonalization of K^TK and L^TL, and therefore requires more arithmetic operations than our transformation. A different transformation is made by Jennings [11], who uses the singular value decomposition of K. Since K is usually very ill-conditioned and L often quite well-conditioned, it is more natural to base the transformation on a decomposition of L. The case $\mathrm{rank}(L)=n=p$ has been considered by Voevodin [21].

3. Algorithms for *R*1 and *R*2 in standard form.

In this section we discuss how to compute function values and derivatives of $h_1(\mu)$ and $h_2(\mu)$ ((1.6) and (1.7)) for a sequence of values of μ. From section 2 we see that we can restrict ourselves to the case when $R1$ and $R2$ are in standard form. Then for each μ we must first solve the least squares problem

$$\min_f\{\|Kf-g\|_2^2+\mu\|f\|_2^2\}\,. \tag{3.1}$$

Here K is assumed to be an $m\times n$ matrix (if the transformation of section 2 has been made on an original $m\times n$ matrix, then of course the matrix in (3.1) is $(m-n+p)\times p$). For simplicity we assume that $m\geqq n$ (see [2] for the case $m<n$).

3.1 *Efficient solution of the least squares problem.*

In [7], [8] formulas are given for the solution of (3.1) when a singular value decomposition of the matrix K has been performed. Here we give algorithms based on a bidiagonalization of K. Consider the decomposition

$$K = W\begin{pmatrix}B\\0\end{pmatrix}Y^T\,, \tag{3.2}$$

where W and Y are orthogonal and B is $n\times n$ upper bidiagonal. It is shown in [24], p. 135-136, how this decomposition can be computed in $2(mn^2-n^3/3)$ operations taking W and Y to be products of Householder transformations. We also remark that when $m > (5/3)n$ it is more efficient to perform the decomposition in two steps:

(i) Reduction to triangular form $K = W_1 \begin{pmatrix} R \\ 0 \end{pmatrix}$ with W_1 equal to a product of Householder transformations ($mn^2 - n^3/3$ operations).

(ii) Further reduction to bidiagonal form

$$R = W_2 B Y^T$$

where W_2 and Y are products of approximately $n^2/2$ plane rotations.

This step requires $4n^3/3$ operations, and a typical stage of the reduction is shown below

$$\begin{array}{cc} \begin{array}{ccccc} & \overset{5}{\frown} & \overset{3}{\frown} & \overset{1}{\frown} & \end{array} & \\ \begin{pmatrix} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \\ & & & & x \end{pmatrix} & \begin{array}{l} \\ \}6 \\ \}4 \\ \}2 \\ \\ \end{array} \end{array},$$

where the figures indicate the order in which the rotations are applied. In the column transformations elements in the first row are put equal to zero and new non-zero subdiagonal elements are created. The latter are annihilated in the row transformations. Note that only the matrix Y needs to be saved (if there is only one right hand side to be processed). This algorithm has been suggested to the author by Å. Björck (1974) and it appears to be new.

If we make the transformation of variables

$$(3.3) \qquad f_1 = Y^T f, \quad \hat{g} = W^T g = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \begin{matrix} \}n \\ \}m-n \end{matrix},$$

we easily see that (3.1) is equivalent to

$$(3.4) \qquad \min_{f_1} \left\| \begin{pmatrix} B \\ \sqrt{\mu} I \end{pmatrix} f_1 - \begin{pmatrix} g_1 \\ 0 \end{pmatrix} \right\| .$$

This least squares problem is now solved by first determining an orthogonal matrix Q_μ, such that

$$(3.5) \qquad Q_\mu{}^T \left(\begin{array}{c|c} B & g_1 \\ \sqrt{\mu} I & 0 \end{array} \right) = \left(\begin{array}{c|c} B_\mu & z_1 \\ 0 & z_2 \end{array} \right),$$

where B_μ is upper bidiagonal. Then f_1 is computed from

$$(3.6) \qquad B_\mu f_1 = z_1 .$$

The matrix Q_μ in (3.5) is constructed as a product of plane rotations. We demonstrate the algorithm by considering a small example ($n=3$). First a rotation in the $(1,n+1)$-plane annihilates the element $(n+1,1)$, and a non-zero element is created in $(n+1,2)$. This is canceled by a rotation in the $(n+1,n+2)$-plane.

$$(3.7)\qquad \begin{pmatrix} a_1 & b_1 & \\ & a_2 & b_2 \\ & & a_3 \\ \sqrt{\mu} & & \\ & \sqrt{\mu} & \\ & & \sqrt{\mu} \end{pmatrix} \curvearrowright \begin{pmatrix} a_1' & b_1' & \\ & a_2 & b_2 \\ & & a_3 \\ 0 & \lambda & \\ & \sqrt{\mu} & \\ & & \sqrt{\mu} \end{pmatrix} \curvearrowright \begin{pmatrix} a_1' & b_1' & \\ & a_2 & b_2 \\ & & a_3 \\ 0 & 0 & \\ & \sqrt{\mu+\lambda^2} & \\ & & \sqrt{\mu} \end{pmatrix}$$

where

$$a_1' = (a_1^2+\mu)^{\frac{1}{2}}, \qquad b_1' = b_1 a_1/a_1', \qquad \lambda = b_1\sqrt{\mu}/a_1' .$$

Now the dimension of the problem has been reduced by one. In the next step the element $(n+2,2)$ disappears after a rotation in the $(2,n+2)$-plane, and so on. A description of the algorithm in pseudo-Algol can be found in [2]. The whole transformation (3.5) together with the solution of (3.6) requires about $11n$ multiplications, $5n$ divisions, $5n$ additions and $2n$ square roots.

The main advantage of our algorithm lies in using the bidiagonalization (3.2) instead of the *SVD*. If $m \approx n$ then the computation of (3.2) (using the first, finite part of the *SVD*-algorithm [24]) usually requires less than one third of the work for the full *SVD*. The number of operations for the repeated solution of (3.1) using our algorithms is of the same order of magnitude as for the corresponding *SVD*-based algorithms (see section 3.3 for operation counts).

3.2 *Algorithms for R*1 *and R*2

Before giving formulas for the computation of function values and derivatives of $h_1(\mu)$, we make some remarks about the difficulties which arise in the numerical solution of the equation $h_1(\mu)=0$ (the remarks also apply to the solution of $h_2(\mu)=0$).

It is easily seen that $h_1(\mu)$ is a rational function in μ, and that all the poles lie on the negative real axis (due to (1.5)). In general the equation $h_1(\mu)=0$ has exactly one positive solution (see [3]), which can be found by e.g. Newton's method. Often this solution is small and since the rightmost pole lies close to the origin, the convergence of Newton's method will be slow. One possible method to accelerate the convergence is to solve the equation $1/\|f\|_2^2 - 1/\omega^2 = 0$, instead of $\|f\|_2^2 - \omega^2 = 0$ ([16],

[8]). Numerical tests indicate that iterative methods based on rational approximation converge faster than methods of the same order based on polynomial approximation. As will be seen below, derivatives of arbitrary order can be computed, each derivative requiring less than one fifth of the work for computing one function value. Therefore high order iteration methods can be used. We have tried Halley's method, which in some cases converges about twice as fast as Newton's method. Further research is needed in this area.

In $R1$ we shall compute (remember $L=I$) $h_1(\mu)=f^Tf-\omega^2$. From (3.3) we get $h_1(\mu)=f_1^Tf_1-\omega^2$. Thus, given f_1, the computation of each function value requires n operations. In [2] it has been shown that also derivatives of h_1 can be computed cheaply. We have

$$h_1'(\mu) = -2v_1^Tv_1, \quad B_\mu^Tv_1 = f_1 ,$$

where B_μ is defined by (3.5).
Similarly higher derivatives can be computed recursively:

$$h_1^{(p)}(\mu) = (-1)^p(p+1)!\,v_p^Tv_p, \quad p \geqq 0 , \tag{3.8}$$

where

$$v_0 = f_1, \quad v_p = \begin{cases} B_\mu^{-1}v_{p-1}, & p \text{ even}, \\ (B_\mu^T)^{-1}v_{p-1}, & p \text{ odd}. \end{cases} \tag{3.9}$$

Each derivative thus requires about $2n$ multiplications and additions and n divisions.

In $R2$ we shall compute $h_2(\mu)=\|Kf-g\|_2^2-\varepsilon^2$. Using the decomposition (3.2) and (3.3) we get

$$h_2(\mu) = \|Bf_1-g_1\|_2^2+\|g_2\|_2^2-\varepsilon^2 .$$

The computation of h_2 with given f_1 requires about $3n$ operations. Again derivatives can be computed cheaply. We find (see [2])

$$h_2^{(p)}(\mu) = (-1)^p p!\,f_1^T[(p-1)C_\mu^{-p+1}-\mu(p+1)C_\mu^{-p}]f_1 ,$$

where $C_\mu=B_\mu^TB_\mu$. It is easily seen that the recursive scheme (3.8), (3.9) can be used to compute each derivative with the same amount of work as in $R1$.

3.3 *Summary of operation counts and comparison to SVD-based algorithms.*

In table 1 we summarize the operation counts for the algorithms in section 3.1-3.2 and for the corresponding *SVD*-based algorithms. We assume that $m\approx n$ and give the leading term in the count of floating point multiplications and divisions. Note that the operation count is made for each algorithm separately, so that e.g. the count n operations for computing $h_1(\mu)=f_1{}^Tf_1-\omega^2$ presupposes that f_1 is available.

Table 1. Operation count for the algorithms in section 3.

	Number of operations for	
	our algorithm	corresponding SVD-based algorithm
Decomposition of K (3.2)	$4n^3/3\cdot$	$3n^3$ to $5n^3$
Solution of (3.4)	$16n$ $+2n$ square roots	n
h_1: function values	n	n
derivatives	$3n$	$1.5n$
h_2: function values	$3n$	$2n$
derivatives	$3n$	$1.5n$
Backtransformation (3.3)	n^2	n^2

Remark. For the transformation of variables (3.3) the matrix Y is needed, and it can be stored in factorized form in the empty locations of K both when Y is taken as a product of Householder transformations and when Y is taken as a product of plane rotations. In the latter case the backtransformation (3.3) requires $2n^2$ operations. When the SVD is computed, the orthogonal matrix corresponding to Y must be computed explicitly. The count $3n^3$ to $5n^3$ operations presupposes that the SVD subroutine has the option to compute only one of the orthogonal factors.

From Table 1 it is seen that the major part of the work in both approaches lies in the decomposition of K. The total amount of work for solving a specific problem depends of course on the number of different values of μ, for which (3.4) is solved, and on the number of derivatives computed for each μ. In Table 2 we compare the total amount of work using our algorithms and SVD-based algorithms in a typical example. We define

$$C_0(n,k) = 4n^3/3+23nk ,$$

which is the total amount of work in our algorithms, if (3.4) is solved for k different μ and if for each μ, $h_1(\mu)$, $h_1'(\mu)$ and $h_1''(\mu)$ are computed. The corresponding quantity for the SVD-based algorithms is taken to be

$$C_s(n,k) = 4n^3+5nk .$$

Note, that the work for the backtransformation, which is the same for both approaches, has not been included. In Table 2 we give the ratio C_0/C_s for different n and k.

Table 2. The ratio C_0/C_s.

$k \backslash n$	10	15	20	30	40	60
10	0.81	0.56	0.46	0.39	0.37	0.35
20	1.2	0.76	0.58	0.45	0.40	0.36
30	1.5	0.94	0.70	0.50	0.43	0.38

Thus we conclude that for practical values of n the overall efficiency or our algorithms is better.

4. Conclusion.

The algorithms described in this paper permit efficient experiments with the choice of regularization method and parameter. This is an advantage, since sometimes it may be difficult to obtain the a priori information required for $R1$ and $R2$. An interactive program for regularization is presently being developed, based on the algorithms given in this paper. It uses a graphic display for presentation of the results and allows the user to study interesting features of the solution as the regularization is varied. A pilot version is running (Feb. 1977) on a DEC-10 system. Other regularization methods than the two described in this paper will also be implemented, e.g. the one based on generalized cross validation by Wahba [23]. The program will be described in two forthcoming reports [4], [5].

5. Acknowledgements.

The author wishes to thank Professor Åke Björck for his support during this work. He has provided constructive criticism and suggested a considerable improvement of the algorithm (3.7). Helpful advice has also been given by the referees.

This work was supported in part by the Swedish Institute for Applied Mathematics.

REFERENCES

1. B. C. Cook, *Least Structure Solution of Photonuclear Yield Functions*, Nuclear Instruments and Methods 24 (1963), pp. 256–268.
2. L. Eldén, *Numerical Methods for the Regularization of Fredholm Integral Equations of the First Kind*, Report LiH-MAT-R-1974-7, Dept. of Mathematics, Linköping University, 1974.
3. L. Eldén, *A Note on Weighted Pseudoinverses with Application to the Regularization of Fredholm Integral Equations of the First Kind*, Report LiTH-MAT-R-75-11, 1975, Dept. of Mathematics, Linköping University.

4. L. Eldén, *A Program for Interactive Regularization*, Part I: Implementation of Numerical Algorithms, to appear as a Linköping University, Dept. of Mathematics report.

5. L. Eldén, *A Program for Interactive Regularization*, Part II: Interaction and File Handling, to appear as a Linköping University, Dept. of Mathematics report.

6. G. H. Golub and V. Pereyra, *The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems Whose Variables Separate*, SIAM J. Numer. Anal. 10 (1973) pp. 413–432.

7. G. H. Golub, *Some Modified Eigenvalue Problems*, SIAM Review 15 (1973), pp. 318–334.

8. V. I. Gordonova and V. A. Morozov, *Numerical Algorithms for Parameter Choice in the Regularization Method*, Zh. Vychisl. Mat. Mat. Fiz. 13 (1973), pp. 539–545.

9. R. J. Hanson and J. L. Phillips, *An Adaptive Numerical Method for Solving Linear Fredholm Integral Equations of the First Kind*, Numer. Math. 24 (1975), pp. 291–307.

10. V. K. Ivanov, *On Linear Problems which are not Well-Posed*, Dokl. Akad. Nauk SSSR 145 (1962), pp. 270–272.

11. L. S. Jennings, *Orthogonal Transformations and Improperly Posed Problems*, Thesis, Australian National University, 1973.

12. N. Köckler, *Parameterwahl und Fehlerabschätzung bei der regularisierten Lösung von inkorrekt gestellten Problemen*, Dissertation, Johannes Gutenberg-Universität, Mainz, 1974.

13. C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, 1974.

14. M. Z. Nashed, *On Moment Discretization and Least Squares Solutions of Linear Integral Equations of the First Kind*, J. Math. Anal. Appl. 53 (1976), pp. 359–366.

15. D. L. Phillips, *A Technique for the Numerical Solution of Certain Integral Equations of the First Kind*, J. ACM 9 (1962), pp. 84–97.

16. C. H. Reinsch, *Smoothing by Spline Functions*, Numer. Math. 16 (1971), pp. 451–454.

17. A. N. Tihonov, *Solution of Incorrectly Formulated Problems and the Regularization Method*, Dokl. Akad, Nauk SSSR 151 (1963), pp. 501–504 = Soviet Math. Dokl. 4 (1963), pp. 1035–1038.

18. A. N. Tihonov, *Regularization of Incorrectly Posed Problems*, Dokl. Akad. Nauk SSSR 153 (1963), pp. 49–52 = Soviet Math. Dokl. 4 (1963), pp. 1624–1627.

19. C. F. Van Loan, *Generalizing the Singular Value Decomposition*, SIAM J. Numer. Anal. 13 (1976), pp. 76–83.

20. J. M. Varah, *A Practical Examination of Some Numerical Methods for Linear Discrete Ill-Posed Problems*, Technical Report 76–08, Dept. of Computer Science, University of British Columbia, Vancouver, 1976.

21. V. V. Voevodin, *The Method of Regularization*, Zh. Vychisl. Mat. Mat. Fiz. 9 (1969), pp. 673–675 = USSR Comp. Math. Math. Phys. 9 (1969), pp. 228–232.

22. G. Wahba, *Convergence Rates of Certain Approximate Solutions to Fredholm Integral Equations of the First Kind*, J. Appr. Theory 8 (1973), pp. 167–185.

23. G. Wahba, *Practical Approximate Solutions to Linear Operator Equations when the Data are Noisy*, SIAM J. Numer. Anal., to appear.

24. J. H. Wilkinson and C. H. Reinsch, *Handbook for Automatic Computation*, Vol. II, Linear Algebra, Springer-Verlag, Berlin, 1971.

LINKÖPING UNIVERSITY
DEPARTMENT OF MATHEMATICS
S-581 83 LINKÖPING
SWEDEN