27
3-23-76
25 NTIS

MASTER

IS-3841

AN ITERATIVE METHOD FOR VOLTERRA INTEGRAL EQUATIONS OF THE FIRST KIND

William C. Peterson
and Robert J. Lambert

AMES LABORATORY, USERDA
IOWA STATE UNIVERSITY
AMES, IOWA

Date Transmitted: February 1976

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
UNDER CONTRACT W-7405 eng-82

DISCLAIMER

DISCLAIMER

IS-3841

AN ITERATIVE METHOD FOR VOLTERRA INTEGRAL EQUATIONS OF THE FIRST KIND

William C. Peterson and Robert J. Lambert

Ames Laboratory, ERDA
Iowa State University
Ames, Iowa 50010

Date Transmitted: February 1976

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION UNDER CONTRACT NO. W-7405-eng-82

Available from: National Technical Information Service
U. S. Department of Commerce
P.O. Box 1553
Springfield, VA 22161

Price: Microfiche $2.25
Paper Copy $4.00

TABLE OF CONTENTS

ABSTRACT

An iterative method is developed to find an approximation to the least squares solution of minimum $L_2(0,1)$ norm, f_0, of a Volterra integral equation of the first kind

$$Lf(g) = \int_0^y k(y,x)f(x)dx = g(y) \quad 0 \le y \le 1$$

where $g \in L_2(0,1)$ and

$$\int_0^1 \int_0^y k(y,x)^2 dxdy < \infty.$$

A sequence $\{f_{\lambda_j}\}_{j=1}^{\infty} \subset N(L)$ converging to f_0 is constructed. f_{λ_j} minimizes the functional

$$Q_{\lambda_j}(f) = ||Lf - g||^2 + \lambda_j ||f||^2, \ \lambda_j > 0.$$

Distribution List

Chicago Patent Office	1
USERDA-TIC	27
Ames Laboratory Library	5
Robert J. Lambert	16
William C. Peterson	5
Total	54

Introduction

We consider the Volterra integral equation of the first kind,

$$Kf(y) = \int_0^y k(y,x)f(x)dx = g(y) \qquad (1)$$

where k(y,x), g(y) are given functions and f(x) is to be determined. Here we assume that we have Riemann integration.

Hochstadt [5] notes that the equation

$$\int_0^y k(y,x)f(x)dx = g(y), \; g(0) = 0 \qquad (2)$$

reduces to a Volterra integral equation of the second kind if the kernel k(y,x) and g(y) are continuously differentiable and $k(y,y) \neq 0$. The Volterra integral equation (2) is then transformed into the Volterra integral equation of the second kind

$$f(y)k(y,y) + \int_0^y k_1(y,x)f(x)dx = g'(y)$$

or

$$f(y) + \int_0^y \frac{k_1(y,x)}{k(y,y)} f(x)dx = \frac{g'(y)}{k(y,y)}, \; g(0) = 0. \qquad (3)$$

Hochstadt gives a proof of the existance and uniqueness of the solution to (3).

Mikhlen and Smoletskey [10] suggest that a numerical approximation to the solution of (2) with the above hypotheses can be obtained without reduction to an integral equation of the second kind by the application of quadrature formulas. They obtain the system of equations

$$\sum_{i=1}^{n} A_i^{(n)} k(x_n, x_i) f(x_i) = g(x_n) \qquad (4)$$

where $A_i^{(n)} \neq 0$ for $i = 1, \ldots, n$ are the quadrature weights. From (4) an approximation to $f(x_n)$ is determined recursively at all mesh points of the quadrature formula, except for $x_0 = 0$ where they take $f(0) = \frac{g'(0)}{k(0,0)}$.

Riesz and Nagy [15] consider the Volterra integral equation

$$\int_0^y k(y,x) f(x) dx = g(y), \quad 0 \le y \le 1 \qquad (5)$$

as a special case of the Fredholm integral equation of the first kind

$$Lf(y) = \int_0^1 \ell(y,x) f(x) dx = g(y) \qquad (6)$$

where

$$\ell(y,x) = \begin{cases} k(y,x) & \text{if } y \ge x \\ 0 & \text{if } y < x. \end{cases}$$

We will assume that (1) is written in the form (6). In addition we will assume that $\int_0^1 \int_0^1 | \ell(y,x) |^2 dxdy < \infty$ so that L is a completely continuous linear operator from $H_1 = L_2[0,1]$ to $H_2 = L_2[0,1]$ and $g \in L_2[0,1] = H_2$.

We will use the notation: $R(L)$ = range of L, $N(L) = \{f \in H_1 : Lf = 0\}$ and $M^{\perp} = \{f \in H_1 : \langle f,h \rangle = 0 \text{ for all } h \in M \subset H_1\}$ where $\langle \ , \ \rangle$ denotes inner product in the Hilbert space $L_2[0,1]$.

From Strand [17] we have that for a completely continuous operator L $H_2 = N(L^*) + \overline{R(L)}$ where $\overline{R(L)}$ is the closure of R(L), $H_1 = N(L) + (N(L))^{\perp}$, and $\overline{R(L^*)} = (N(L))^{\perp}$.

We are interested in finding an approximation to the least squares solution of minimum norm, f_0; of the operator equation $Lf = g$ where $g \in H_2$, when such a solution exists. Kammerer and Nashed [8] give the following definitions.

Definition 1. For $f \in H_1$, $g \in H_2$, an element $\overline{f} \in H_1$ is a least squares solution of $Lf = g$ if and only if

$|| L\overline{f} - g || = \inf \{|| Lf - g || \mid f \in H_1\}$.

Definition 2. For $f \in H_1$, $g \in H_2$, an element $f_0 \in H_1$ is a least squares solution of $Lf = g$ and $|| f_0 || \leq || \overline{f} ||$ for all least squares solutions $\overline{f}$ of $Lf = g$.

Strand [17] notes that the least squares solution of minimum norm exists (is unique and in $N(L)^{\perp}$) if and only if $g \in H_2$ can be written in the form

$$g = g_1 + g_2$$

where $g_1 \in R(L)$ and $g_2 \in N(L^*)$. For this paper we will assume g is always in the form $g = g_1 + g_2$, with $g_1 \in R(L)$. From Strand [17] and Diaz and Metcalf [3], f is a solution of

$$|| Lf - g || = \inf \{|| Lh - g || \mid h \in H_1\} \quad (7)$$

if and only if f is a solution to $Kf = g_1$. They also note that the least squares solutions to $L^*Lf = L^*g$ and $Lf = g_1$ are the same.

Twomey [20], Phillips [14], Tikhonov [18] and Strand [17] considers the functional

$$Q(f) = \| Lf - g \|^2 + \lambda \| f - p \|^2, \ \lambda > 0, \qquad (8)$$

where $p \in H_1$ is an estimate for the least squares solution of minimum norm of $Lf = g$. Strand [17] proves that (8) is minimized by

$$f_\lambda = (L^*L + \lambda I)^{-1} (L^*g + \lambda p).$$

Strand [17] gives a proof of the following theorem.

<u>Theorem 1</u>. $f_p = \lim_{\lambda \to 0^+} f_\lambda$ exists and f_p is the unique solution of $L^*Lf = L^*g$ for which $\| f - p \|^2$ is a minimum.

For the case where $p \equiv 0$ we have an approximation to the least squares solution of minimum norm of $Lf = g$. In general a method for determing a good choice for λ in actual applications and construction of the inverse to represent $(L^*L + \lambda I)^{-1}$ is difficult since $(L^*L + \lambda I)^{-1}$ is generally ill-conditioned for a useful choice for λ.

Basic Theorem

The following theorem will be implemented as a numerical algorithm, related to a steepest discent method for Tikhonov-Twomey regularization, which converges to the least squares solution of minimum norm, when such a solution exists. The method developed allows one to find an approximation to the solution of $(L^*L + \lambda I)f = L^*g$ for a $\lambda > 0$. When the convergence fails or becomes slow then the advantage of this method is that a new starting vector can be obtained. A proof and discussion of the theorem as well as a choice for the starting vector $f_{1,0}$ can be found in [13].

<u>Theorem 2</u>. If

1. f_0 is the least squares solution of minimum norm of $Lf = g$,
2. $f_{1,0} \in H_1$ be given as starting vector,
3. $\{\{f_{j,n}\}_{n=0}^{\infty}\}_{j=1}^{\infty}$ be a sequence such that

$$f_{j,n} = f_{j,n-1} + \alpha_{j,n} W_{j,n}$$

where

$$W_{j,n} = L^*Lf_{j,n} + \lambda_j f_{j,n} - L^*g$$

$$\alpha_{j,n} = \begin{cases} -\dfrac{< W_{j,n}, W_{j,n} >}{< LW_{j,n}, LW_{j,n} > + \lambda_j < W_{j,n}, W_{j,n} >} & \text{for } W_{j,n} \neq 0 \\ 0 \text{ for all } n > n_j \text{ if } n_j \text{ is the first } n \\ \text{such that } W_{j,n} = 0, \end{cases}$$

4. $\{\lambda_j\}_{j=1}^{\infty}$ is a sequence of positive real numbers converging to zero,

then for any $\epsilon > 0$ there exists positive integers $\bar{n}$ and $\bar{j}$ such that $||f_{\bar{j},\bar{n}} - f_0|| < \epsilon$.

For j fixed in the above theorem Peterson [13] notes that $\{f_{j,n}\}_{n=0}^{\infty}$ converges to $f_{\lambda_j} = f_j$, the function in the R(L*) that minimizes

$$Q_{\lambda_j}(f) = || Lf - g ||^2 + \lambda_j || f ||^2 .$$

Let $\sigma_0(L^*L)$ be the set of non-zero eigenvalues of L*L. We have the following theorem, proven in [13], using the above notation, related to convergence.

<u>Theorem 3</u>. If $\gamma = \inf\limits_{\gamma_n \in \sigma_0(L^*L)} \gamma_n$

then

$$|| f_{j,n} - f_0 || \le \frac{|| W_{j,n} ||}{\gamma + \lambda_j} + \frac{\lambda_j || f_j ||}{\gamma} \text{ when } \gamma \neq 0$$

and

$$|| f_{j,n} - f_0 || \le \frac{|| W_{j,n} ||}{\gamma + \lambda_j} + \frac{\lambda_j || f_0 ||}{\lambda_j + \gamma} .$$

Discretization of the Integral Equation

We will consider the real n-dimensional space R^n with scalar field the real numbers. Define a mapping from $R^n x R^n$ to R by

$$(U,V) \to \langle U,V \rangle = \sum_{j=1}^{n} T_j U_j V_j$$

where $U = (U_1, U_2, \ldots, U_n)^T$, $V = (V_1, V_2, \ldots, V_n)^T$ and $T_j > 0$, $T_j \in R$ for all $j = 1, \ldots, n$. Thus we have a Hilbert space, which will be denoted by H_T^n. Let $\{e_1^n, e_2^n, \ldots, e_n^n\}$ be the orthogonal (not necessarily normalized) basis for H_T^n where $e_j^n = (0, 0, \ldots, 1_j, \ldots, 0)^T$, $j = 1, n$.

If $A : H_T^m \to H_S^n$ then $A^* : H_S^n \to H_T^m$ will denote the adjoint. The following theorem is given in [13].

<u>Theorem 4</u>. If $A : H_T^m \to H_S^n$,

$$A = (T_j a_{i,j})\begin{matrix} i = 1,n \\ j = 1,m \end{matrix} \qquad \text{then} \qquad A^* = (S_i a_{i,j})\begin{matrix} j = 1,m. \\ i = 1,n \end{matrix}$$

In this paper we will be restricted to the case where $A : H_S^n \to H_S^n$ and $a_{i,j} = 0$ for $i > j$, with the additional requirement that the S_i for $i = 1, \ldots, n$ appearing in the lower triangular matrix A depends on the row as well as the column.

<u>Theorem 5</u>. If $A : H_S^n \to H_S^n$,

$$A = (S_j^{(i)} a_{i,j})\begin{matrix} i = 1,n \\ j = 1,n \end{matrix} \qquad \text{where} \qquad a_{i,j} = 0$$

for $i > j$ then $A^* = (S_i^{(j)} a_{i,j})\begin{matrix} j = 1,n \\ i = 1,n \end{matrix}$

where $S_i^{(j)} = S_j$ for $i \neq j$.

Proof: We will show that the above A* is the adjoint of A.

$$< Ae_k^n, e_\ell^n > = < \sum_{i=1}^{n} S_k^{(i)} a_{i,k} e_i^n, e_\ell^n >$$

$$= S_k^{(\ell)} a_{\ell,k} < e_\ell^n, e_\ell^n >$$

$$= S_k^{(\ell)} a_{\ell,k} S_\ell$$

and

$$< e_k^n, A^* e_\ell^n > = < e_k^n, \sum_{j=1}^{n} S_\ell^{(j)} a_{\ell,j} e_j >$$

$$= S_\ell^{(k)} a_{\ell,k} < e_\ell^n, e_k^n >$$

$$= S_\ell^{(k)} a_{\ell,k} S_k$$

$$= < Ae_k^n, e_\ell^n >$$

for all k and ℓ, therefore A* is the adjoint of A.

Now we develop a method of discretization of the integral equation

$$\int_0^1 \ell(y,s) \int_0^1 \ell(y,x) f(x)dxdy + \lambda f(s) = \int_0^1 \ell(y,s) g(y)dy \qquad (6)$$

where from Bachman and Narici [2] $L^*f(s) = \int_0^1 \ell(x,s) f(x)dx$ is the adjoint to L.

(6) is equivalent to

$$\int_s^1 k(y,s) \int_0^y k(y,x) f(x)dxdy + \lambda f(s) = \int_s^1 k(y,s) g(y)dy. \qquad (7)$$

Partition the interval [0,1] into n-1 equal subintervals, $t_i, y_i, x_i \in [0,1]$ such that

$$0 = y_1 < y_2 < \ldots < y_n = 1,$$

$$0 = x_1 < x_2 < \ldots < x_n = 1,$$

and

$$0 = t_1 < t_2 < \ldots < t_n = 1.$$

Now from (7) we can write

$$\int_{t_i}^{1} k(y,t_i)g(y)dy = \sum_{j=i}^{n} S_j^{(i)} k(y_j,t_i)g(y_j) + R_n(t_i)$$

where $R_n(t_i)$ is the discretization error. Here $S_j^{(i)} > 0 \quad j = i,n$ and $i = 1,n$. For $\int_{t_i}^{1} k(y,t_i)\int_0^y k(y,x)f(x)dxdy$ we have

$$\int_{t_i}^{1} k(y,t_i)\int_0^y k(y,x)f(x)dxdy = \sum_{j=i}^{n} S_j^{(i)} k(y_j,t_i) \sum_{k=1}^{j} S_k^{(j)} k(y_j,x_k)f(x_k) + \overline{R}(t_i)$$

where $\overline{R}(t_i)$ is the discretization error. Thus (7) can be written in the form, for $s = t_i$,

$$\sum_{j=i}^{n} S_j^{(i)} k(y_j,t_i) \sum_{k=1}^{j} S_k^{(j)} k(y_j,x_k)f(x_k) + \lambda f(t_i) + \overline{R}\,(t_i)$$
$$= \sum_{j=i}^{n} S_j^{(i)} k(y_j,t_i)g(y_j) + R_n(t_i). \tag{8}$$

(8) can be written in the equivalent form

$$(S_j^{(i)}k(y_j,t_i))^T_{j=1,n}(S_k^{(j)}k(y_j,x_k))_{\substack{j=1,n\\k=1,n}}(f(x_k))_{k=1,n}$$

$$+ \lambda f(t_i) + \overline{R}(t_i) = \sum_{J=i}^{n} S_j^{(i)}k(y_j,t_i)g(y_j) + R_n(t_i)$$

where $k(y_j,x_k) = 0$ for $k > j$,

$k(y_j,t_i) = 0$ for $j > i$

and

$(v_i)_{i=1,\ell}$ denotes an ℓ-dimensional column vector.

Now taking $i = 1, \ldots, n$ we obtain the linear system

$$(S_j^{(i)}k(y_j,t_i))_{\substack{i=1,n\\j=1,n}}\cdot(S_i^{(k)}k(y_j,x_k))_{\substack{j=1,n\\k=1,n}}(f(x_k))_{k=1,n}$$

$$+ \lambda(f(t_i))_{i=1,n} = (S_j^{(i)}k(y_j,t_i))_{\substack{i=1,n\\j=1,n}}(g(y_j))_{j=1,n}$$

$$+ (R_n(t_i) - \overline{R}(t_i))_{i=1,n}.$$

For notational convenience we can take $x_i = t_i$, $i = 1,n$ and replace the index k with i.

For convenience, using the same notation as for the original integral equation, let

$$L = (S_i^{(j)}k(y_j,x_i))_{\substack{j=1,n\\i=1,n}}, \qquad L : H_S^k \to H_S^k;$$

$$L^* = (S_j^{(i)} k(y_j, x_i))_{\substack{i=1,n \\ j=1,n}}, \qquad L^* : H_S^n \to H_S^n;$$

$$\vec{f} = (f(x_i))_{i=1,n} \qquad , \qquad \vec{f} \in H_S^n;$$

$$\vec{g} = (g(x_j))_{j=1,n} \qquad , \qquad \vec{g} \in H_S^n$$

and

$$\vec{R}_1 = (R_n(x_i) - R(x_i))_{i=1,n}, \qquad \vec{R} \in H_S^n \quad .$$

Now with theorem 5, the same error bounds hold, as developed in [13], where $L^*L\vec{f}_0 = L^*\vec{g} + \vec{R}_0$, $\vec{R}_0$ the discretization error vector in the matrix representation for

$$\int_t^1 k(y,t) \int_0^y k(y,x) f_0(x) dx dy = \int_t^1 k(y,t) g(y) dy \ .$$

The theorem relating error bounds is given for the case where g is not known exactly, we have

$$\int_0^y k(y,x) f(x) dx = g(y) + \epsilon(y),$$

where $\epsilon(y) \in L_2[0,1]$. (9) may or may not have a solution, see Strand [17].

Similar to the previous discretization we obtain the matrix representation

$$L^*L\vec{f} + \lambda\vec{f} = L^*\vec{g} + L^*\vec{\epsilon} + \vec{R}_2$$

where $\vec{R}_2$ is the discretization error vector.

The system of equations actually solved is

$$L^*L\vec{f} + \lambda\vec{f} = L^*\vec{g} + L^*\vec{\epsilon}. \tag{10}$$

Let $\vec{\bar{f}}_{j,n,\epsilon}$ be a term in the sequence defined in theorem 2 for (10). If a bound for $\| L^*\vec{\epsilon} \|$ or $\| \vec{\epsilon} \|$ can be found, we can obtain bounds on $\| \vec{f}_0 - \vec{\bar{f}}_{j,n,\epsilon} \|$ where $\vec{\bar{f}}_{\lambda,\epsilon}$ is the solution to (10) as stated in the theorem below.

<u>Theorem 6</u>.

$$\|L^*L(\vec{f}_0 - \vec{f}_{j,n,\epsilon})\| \le \|\vec{R}_0\| + \|L^*L\vec{\bar{f}}_{j,n,\epsilon} - L^*(\vec{g} + \vec{\epsilon})\| \tag{11}$$

$$\|\vec{f}_0 - \vec{\bar{f}}_{j,n,\epsilon}\| \le \frac{\|\vec{R}_0\| + \|L^*L\vec{\bar{f}}_{j,n,\epsilon} - L^*(\vec{g} + \vec{\epsilon})\|}{\gamma}, \text{ if } \gamma \ne 0 \tag{12}$$

$$\|\vec{f}_0 - \vec{\bar{f}}_{j,n,\epsilon}\| \le \frac{\|W_{j,n}\|}{\lambda_j + \gamma} + \frac{\|\vec{R}_0\| + \|L^*\vec{\epsilon}\| + \lambda_j\|\vec{\bar{f}}_j\|}{\gamma}, \text{ if } \gamma \ne 0 \tag{13}$$

$$\|\vec{f}_0 - \vec{\bar{f}}_{j,n,\epsilon}\| \le \frac{\|W_{j,n}\|}{\lambda_j + \gamma} + \frac{\|\vec{R}_0\| + \|L^*\vec{\epsilon}\| + \lambda_j\|\vec{f}_0\|}{\lambda_j + \gamma} \tag{14}$$

$$\|\vec{f}_0 - \vec{\bar{f}}_{j,n,\epsilon}\| \le \frac{\|W_{j,n}\|}{\lambda_j + \gamma} + \frac{(\lambda_j + \gamma)\|\vec{R}_0\| + \lambda_j\|L^*(\vec{g} + \vec{\epsilon})\|}{\gamma(\lambda_j + \gamma)}, \text{ if } \gamma \ne 0. \tag{15}$$

If g is known exactly, take $\epsilon = 0$.

The proof of theorem 6 follows that given in [13] and will be omitted here.

Examples

Example 1

We will consider the Volterra integral equation of the first kind

$$\int_0^y (1. + x + y)f(x)dx = \frac{y^2}{2} + \frac{5}{6}y^3 \quad 0 \le y \le 1 \qquad (16)$$

with unique solution $f(y) = y$.

The quadrature method used for (15) was the trapezoidal method with 11 equally spaced points, so $L^*L : H_T^{11} \rightarrow H_T^{11}$. The matrix operator L^*L constructed has smallest non-zero eigenvalue 5.08×10^{-5}. For such a small minimum eigenvalue the error bounds given in theorem 6 are not readily applicable. Numerical experimentation indicates that, in general, a good choice for terminal λ_j in the algorithm given in the appendix is to choose terminal λ_j to be such that $10\lambda_j \cdot ||\vec{f}_0||$ or $10\lambda_j \cdot ||\vec{f}_j||$ is of the same order of magnitude as $||\vec{R}_0||$.

For example 1, $||\vec{R}_0||$ was computed to be 1.2×10^{-3} and $||\vec{f}_0|| = \sqrt{3}/3$. The following initial information was used in the algorithm in the appendix:

Terminal Lambda = $1. \times 10^{-4}$,

Control in $||W_{n,j}||^2 = 1. \times 10^{-12}$,

Lambda multiplier = .02,

and

Starting vector = $(0.,0.,0.,0.,0.,0.,0.,0.,1.,1.,1.)^T$.

The results of using the algorithm are given in table 1 under S1.

The algorithm reduced the following norms to the results indicated:

$$||ATAF + LAMDA*F - ATG|| = .2333x10^{-5},$$

$$||AF - G|| = .1696x10^{-3},$$

and

$$||AT(Af - G)|| = .5975x10^{-4}$$

after 554 iterations.

Example 1 was also solved using the method given in (4) with results given in table 1 under S2.

Table 1

X	S1	D1	S2	D2
0.0	0.0030	-0.0030	0.	0
0.1	0.0962	0.0038	0.0972	0.0028
0.2	0.2002	-0.0002	0.2004	-0.0004
0.3	0.2985	0.0015	0.2979	0.0021
0.4	0.3994	0.0006	0.4005	-0.0005
0.5	0.4998	0.0002	0.4982	0.0018
0.6	0.5981	0.0019	0.6006	-0.0006
0.7	0.7021	-0.0021	0.6985	0.0015
0.8	0.7960	0.0040	0.8006	-0.0006
0.9	0.9040	-0.0040	0.8987	0.0013
1.0	0.9904	0.0096	1.0540	-0.0540
Norm of	0.5779	0.0033	0.5831	0.0126

Entries in table 1 rounded correct to four decimal places.

S1 = Computed solution from algorithm in the appendix;

S2 = Computed solution using (4);

D1 = Actual solution - S1, pointwise;

D2 = Actual solution - S2, pointwise.

Example 2

We will consider the Volterra integral equation of the first kind

$$\int_0^y (x + y)f(x)dx = \frac{3}{2}y^2 + \frac{5}{6}y^3 \qquad 0 \le y \le 1, \qquad (17)$$

with least squares solution of minimum norm $f(y) = 1 + y$.

The quadrature method used for this example was the trapezoidal rule with 11 equally spaced points, so $L^*L : H_T'' \to H_T''$. The matrix operator L^*L has smallest non-zero eigenvalue 1.2×10^{-5}.

For example 2, $||\vec{R}_0||$ was computed to be 6.5×10^{-4} and $||\vec{f}_0|| = 1.5275$.

The following information was used in the algorithm in the appendix:

Terminal Lambda = 4.3×10^{-5},

Control on $||W_{n,j}||^2 = 1.\times10^{-12}$

Lambda multiplier = .02

and

Starting vector = $(0.,0.,0.,0.,0.,0.,0.,0.,1.,1.,1.)^T$.

The algorithm reduced the following norms to the results indicated:

$$||ATAF + Lambda*F - ATG|| = .5642\times10^{-5},$$

$$||AF - G|| = .2135\times10^{-3}$$

and

$$||AT(AF - G)|| = .6839\times10^{-4}$$

after 151 iterations.

The method suggested in (4) is not applicable since $k(0,0) = 0$. The solution is not unique, however, the least squares solution of minimum norm is unique.

Table 2

X	A	S3	D3
0.0	1.0	0.9949	0.0051
0.1	1.1	1.0934	0.0066
0.2	1.2	1.1965	0.0035
0.3	1.3	1.2995	0.0005
0.4	1.4	1.4014	-0.0014
0.5	1.5	1.4967	0.0033
0.6	1.6	1.6026	-0.0026
0.7	1.7	1.6988	0.0012
0.8	1.8	1.7982	0.0018
0.9	1.9	1.9026	-0.0030
1.0	2.0	1.9873	0.0127
Norm of	1.5275	1.5264	0.0043

All entries are rounded correct to four decimal places.

A = Actual solution, pointwise;

S3 = Computed solution, pointwise;

D3 = Actual solution - Computed solution, pointwise.

Example 3

We will consider the Volterra integral equation of the first kind

$$\int_0^y \sin(\pi(x - y))f(x)dx = \frac{1}{2}y\cos(\pi y) - \frac{1}{2\pi}\sin(\pi y) \qquad (18)$$

$0 \le y \le 1$ with least squares solution of minimum norm

$$f(y) = \sin(\pi y).$$

The quadrature method used for this example was the trapezoidal rule with 61 equally spaced points and secondly a simple partition with 61 equally spaced points. For example 3 the following information was used for both the case of the simple partition and the trapezoidal rule in the algorithm in the appendix:

Terminal Lambda = $5.6\text{x}10^{-6}$,

Control on $||W_{n,j}||^2 = 1.\text{x}10^{-14}$,

Lambda multiplier = .05,

and

Starting vector = $(a_1, a_2, \ldots, a_{61})$

where $a_i = 1.$ for $i = 1, \ldots, 61$.

The algorithm reduced the norms given in table 3 to the results indicated.

The final results are given for selected points in table 4.

$$\|\vec{R}_0\| = \begin{cases} 0.4083\text{x}10^{-4} & \text{for the simple partition,} \\ 0.4024\text{x}10^{-4} & \text{for the trapezoidal rule.} \end{cases}$$

Table 3

	Trapezoidal Rule	Simple Partition
N1	$0.5382\text{x}10^{-6}$	$0.5238\text{x}10^{-6}$
N2	$0.5637\text{x}10^{-4}$	$0.5080\text{x}10^{-4}$
N3	$0.3728\text{x}10^{-5}$	$0.3975\text{x}10^{-5}$
IT	729	689

N1 = ||ATAF + LAMBDA*F - ATG||,

N2 = ||AF - G||,

N3 = ||AT(AF - G)||,

IT = Total iterations.

Table 4

X	A	S4	D4	S5	D5
0.0	0.0	0.0035	-0.0035	0.0046	-0.0046
0.1	0.3090	0.3091	-0.0001	0.3090	0.0000
0.2	0.5878	0.5877	0.0001	0.5878	0.0000
0.3	0.8090	0.8084	0.0006	0.8085	0.0005
0.4	0.9511	0.9511	0.0000	0.9408	0.0003
0.5	1.0000	1.0016	-0.0016	1.0016	-0.0016
0.6	0.9511	0.9498	0.0012	0.9502	0.0008
0.7	0.8090	0.8059	0.0031	0.8056	0.0034
0.8	0.5878	0.5965	-0.0087	0.5973	-0.0095
0.9	0.3090	0.2941	0.0149	0.2901	0.0190
1.0	0.0	0.0	0.0		
Norm of	0.7071	0.7068	0.0107	0.7066	0.0134

All entries rounded correct to four decimal places.

A = Actual solution, pointwise;

S4 = Computed solution, pointwise, using a simple partition;

D4 = Actual solution - Computed solution;

S5 = Computed solution, pointwise, using trapezoidal rule;

D5 = Actual solution - Pointwise solution.

Conclusion

The dropping of the requirement that $k(y,y) \neq 0$ for $0 \le y \le 1$ gives the algorithm an advantage over the methods found in the literature.

An upper bound for $||\vec{R}_0||$ can be difficult to obtain unless some information about the solution is known.

A better, more useful, method for obtaining theoretical error bounds should be obtained.

Bibliography

1. Anselone, P. M., Collectively Compact Operator Approximation Theory and Applications to Integral Equations, Englewood Cliffs, New Jersey, Prentice Hall, Inc., 1971.

2. Bachman, G. and Narici, L., Functional Analysis, New York, Academic Press Inc., 1966.

3. Diaz, J. B. and Metcalf, F. T., "On Iteration Procedures for Equations of the First Kind, Ax = y, and Picard's Criterion for the Existance of a Solution", Math. Comp. 24 (1970), pp. 923-935.

4. Hanson, R. J., "A Numerical Method for Solving Fredholm Integral Equations of the First Kind Using Singular Values", SIAM J. Numer. Anal. 8 (1971), pp. 616-622.

5. Hochstadt, H., Integral Equations, New York, John Wiley and Sons, Inc., 1973.

6. Isaacson, E. and Keller, H., Analysis of Numerical Methods, New York, John Wiley and Sons, Inc., 1966.

7. Kammer, W. J. and Nashed, M. Z., "Steepest Decsent for Singular Linear Operators with Nonclosed Range", Appl. Anal. 1 (1971), pp. 143-159.

8. Kammer, W. J. and Nashed, M. Z., "Iterative Methods for Best Approximate Solutions of Linear Integral Equations of the First and Second Kinds", J. Math. Anal. Appl. 40 (1972), pp. 437-573.

9. Kammer, W. J. and Hashed, M. Z., "On the Convergence of the Conjugate Gradient Method for Singular Linear Operator Equations", SIAM J. Numer. Anal. 9 (1972), pp. 165-181.

10. Mikhlen, S. G. and Smoletskey, K. L., Approximate Methods for Solution of Differential and Integral Equations, Edited by R. E. Kalaba, New York, Academic Press, Inc., 1971.

11. Nashed, M. Z., "Steepest Descent for Singular Linear Operator Equations", SIAM J. Numer. Anal. 7 (1970), pp. 358-362.

12. Nashed, N. Z., "Generalized Inverses, Normal Solvability, and Iteration for Singular Operator Equations", In Nonlinear Functional Analysis and Applications, pp. 311-359, Edited by L. B. Rall, New York, Academic Press, Inc., 1971.

13. Peterson, W. C., An Iterative Method for Fredholm Equations of the First Kind, Ph. D. Thesis, Iowa State University, 1975.

14. Phillips, D. L., "A Technique for the Numerical Solution of Certain Integral Equations of the First Kind", J. Assoc. Comput. Mach. 9 (1962), pp. 84-97.

15. Riesz, F. and Nagy, B., Functional Analysis, New York, Frederick Ungai Publishing Co., 1955.

16. Smithies, F., "Integral Equations", Cambridge Tracts in Mathematics and Mathematical Physics 49 (1958), pp. 1-169.

17. Strand, O. N., Theory and Methods for Operator Equations of the First Kind, Ph. D. Thesis, Colorado State University, 1972, Ann Arbor, Michigan, University Microfilms (Diss. Akstr. 73:2820).

18. Tikhonov, A. N., "Solution of Incorrectly Formulated Problems and the Regularization Method", Soviet Math. Dokl. 4 (1963), pp. 1035-1038.

19. Tricomi, F. G., Integral Equations, New York, Interscience Publishers, Inc., 1957.

20. Twomey, S., "On the Numerical Solution of Fredholm Integral Equations of the First Kind by Inversion of the Linear System Produced by Quadrature", J. Assoc. Comput. Mach. 10 (1963), pp. 97-101.

Appendix

Subroutine BIT is a program written to find a λ-approximate least squares solution of minimum norm of an Volterra integral equation of the first kind. The subroutine BIT assumes that the integral equation is of the form

$$Lf(y) = \int_0^y k(y,x)f(x)dx = g(y), \quad y \in [0,1].$$

A driver program must be written which obtains the following information:

1. Partitions the interval integrated over, [0,1], into N-1 subintervals. The N-points constructed are stored in XV and YV.

2. Construct the vector GV(I) = g(YV(I)) I = 1,N.

3. Construct the NxN matrix A representing the operator L.

4. Construct the NxN matrix AT representing the operator L*.

5. Construct the vector WTK of weights for integration.

6. The driver program must be dimensioned as the dimension statement for the subroutine BIT.

7. The driver program must supply the controls on convergence given by

   XLMDA = Lambda multiplier,
   BB = Terminal Lambda,
   M = Maximum number of iteration per change in Lambda,

   and

   DD = Control on $||ATAF + LAMBDA*F - ATG||^2$.

```
      SUBROUTINE BIT(N,A,GV,WTK,XV,YV,F,XLMDA,M,BB,AT,
     %ATA,R,T,V,W,WW,S,DD)
      IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION A(N,N),AT(N,N),WTK(N),W(N),XV(N),YV(N)
      DIMENSION F(N),ATA(N,N),GV(N),R(N),V(N),WW(N),S(N)
      DIMENSION T(N)
      LL=0
C
C        TEST FOR A ZERO SOLUTION.
C
      DO 10 I=1,N
      W(I)=0.
      DO 10 J=1,N
   10 W(I)=W(I)+AT(I,J)*GV(J)
      ZZ=0.
      DO 20 I=1,N
   20 ZZ=ZZ+WTK(I)*(W(I)*W(I))
      ZZ=SQRT(ZZ)
      WRITE(6,11) ZZ
   11 FORMAT('0','|| ATG ||=',E16.8)
      IF(ZZ.GE.1.E-20) GO TO 30
      WRITE (6,31)
   31 FORMAT('0',6X,'SOLUTION IS THE ZERO FUNCTION',/)
      GO TO 600
   30 DO 40 I=1,N
      DO 40 J=1,N
      ATA(I,J)=0.
      DO 40 L=1,N
   40 ATA(I,J)=ATA(I,J)+AT(I,L)*A(L,J)
C
C        OBTAIN AN ACTUAL STARTING VECTOR AND INITIAL LAMBDA.
C
      DO 50 I=1,N
      T(I)=0.
      DO 50 J=1,N
   50 T(I)=T(I)+ATA(I,J)*F(J)
      ZZ=0.
      DO 60 I=1,N
   60 ZZ=ZZ+WTK(I)*(T(I)*T(I))
      ZZ=SQRT(ZZ)
      DO 70 I=1,N
   70 T(I)=T(I)/ZZ
      DO 80 I=1,N
      R(I)=0.
      DO 80 J=1,N
   80 R(I)=R(I)+A(I,J)*T(J)
      CR=1.
      U=2.
      J=1
```

```
   92 X=0.0
      DO 90 I=1,N
   90 X=X+(GV(I)-R(I))*(R(I)*WTK(I))
      IF(X.GE.0.0) GO TO 1000
      IF(J.EQ.2) GO TO 1001
      J=J+1
      DO 100 I=1,N
      T(I)=-T(I)
  100 R(I)=-R(I)
      GO TO 92
 1001 DO 110 I=1,N
  110 GV(I)=U*GV(I)
      J=1
      CR=CR*U
      GO TO 92
 1000 DO 120 I=1,N
  120 R(I)=R(I)-GV(I)
      RR=0.0
      DO 130 I=1,N
  130 RR=RR+WTK(I)*(R(I)*R(I))
      DO 140 I=1,N
      W(I)=0.0
      DO 140 J=1,N
  140 W(I)=W(I)+AT(I,J)*R(J)
      DD=DD*CR*CR
      ZZ=0.0
      DO 150 I=1,N
  150 ZZ=ZZ+WTK(I)*(W(I)*W(I))
      YY=ZZ-X*X
      AB=RR+X
      DO 160 I=1,N
  160 W(I)=W(I)+X*T(I)
C
C        OUTPUT SECTION FOR THE STARTING INFORMATION.
C
      WRITE(6,2) X
    2 FORMAT('0','INITIAL LAMBDA=',E16.8)
      WRITE(6,3)
    3 FORMAT('0',12X,'X=',18X,'STARTING=',12X,'ATAF+LAMBDA*F-ATG=',/)
      DO 170 I=1,N
  170 WRITE(6,4) XV(I),T(I),W(I)
    4 FORMAT(' ',3E24.8)
      WRITE(6,5)
    5 FORMAT('0',12X,'Y=',18X,'AF-G=',/)
      DO 180 I=1,N
```

```
  180 WRITE(6,6) YV(I),R(I)
    6 FORMAT(' ',2E24.8)
      WRITE(6,7) CR
    7 FORMAT('0','SCALING FACTOR=',E16.8)
      WRITE(6,8) YY
    8 FORMAT('0','MINIMUM VALUE=',E16.8)
      XX=X
C
C        THE MAIN ITERATIONS OF THE ALGORITHM.
C
  171 X=XX
      K=1
   41 AA=AB
      YY=ZZ
      DO 190 I=1,N
      WW(I)=W(I)
      W(I)=0.0
      DO 190 J=1,N
  190 W(I)=W(I)+AT(I,J)*R(J)
      DO 200 I=1,N
  200 W(I)=W(I)+X*T(I)
      ZZ=0.0
      DO 210 I=1,N
  210 ZZ=ZZ+WTK(I)*(W(I)*W(I))
      IF(ZZ.LT.DD) GO TO 403
      DO 220 I=1,N
      V(I)=R(I)
  220 F(I)=T(I)
      VV=RR
      DO 230 I=1,N
      S(I)=0.0
      DO 230 J=1,N
  230 S(I)=S(I)+A(I,J)*W(J)
      Y=0.0
      DO 240 I=1,N
  240 Y=Y+WTK(I)*(S(I)*S(I))
      Y=Y+X*ZZ
      B=ZZ/Y
      DO 250 I=1,N
  250 T(I)=T(I)-B*W(I)
      DO 260 I=1,N
      R(I)=0.0
      DO 260 J=1,N
  260 R(I)=R(I)+A(I,J)*(J)
      DO 270 I=1,N
  270 R(I)=R(I)-GV(I)
      RR=0.0
      DO 280 I=1,N
```

```
  280 RR=RR+WTK(I)*(R(I)*R(I))
      AB=0.0
      DO 290 I=1,N
  290 AB=AB+WTK(I)*(T(I)*T(I))
      AB=RR+X*AB
      IF(K.EQ.1) GO TO 201
      IF(AB.LT.AA) GO TO 201
      DO 300 I=1,N
      T(I)=F(I)
      W(I)=WW(I)
  300 R(I)=V(I)
      RR=VV
      ZZ=YY
      K=K-1
      GO TO 401
  201 IF(M-K) 600,401,301
  301 K=K+1
      GO TO 41
  403 K=K-1
  401 LL=LL+K
      IF(K.LT.M) GO TO 440
      WRITE(6,1) X
    1 FORMAT('0',3X,'AT LAMBDA=',E16.8,3X,'THE MAXIMUM VALUE OF M HAS
     %BEEN ATTAINED')
  440 IF(X.EQ.BB) GO TO 601
      XX=XLMDA*X
      IF(XX.LT.BB) GO TO 610
      GO TO 171
  610 XX=BB
      GO TO 171
C
C        OUTPUT SECTION FOR THE FINAL RESULTS.
C
  601 WRITE(6,23)
   23 FORMAT('0', 12X,'X=',14X,'FINAL SOLUTION F=',10X,
     %'ATAF+LAMBDA*F-ATG=',/)
      DO 310 I=1,N
      T(I)=T(I)/CR
      W(I)=W(I)/CR
  310 WRITE(6,4) XV(I),T(I),W(I)
      WRITE(6,5)
      DO 320 I=1,N
      R(I)=R(I)/CR
  320 WRITE(6,6) YV(I),R(I)
      ZZ=ZZ/(CR*CR)
      RR=RR/(CR*CR)
      ZZ=SQRT(ZZ)
      RR=SQRT(RR)
      WRITE(6,26) ZZ,RR
```

```
 26 FORMAT('0','|| ATAF+LAMBDA*F-ATG ||=',E16.8,5X,'||AF-G||=',
   %E16.8)
    WRITE(6,28) LL
 28 FORMAT('0','TOTAL ITERATIONS=',I6)
    CC=0.0
    DO 330 I=1,N
330 CC=CC+WTK(I)*(T(I)*T(I))
    CC=SQRT(CC)
    WRITE(6,9) CC
  9 FORMAT('0','NORM OF THE SOLUTION=',E16.8)
    DO 340 I=1,N
    V(I)=0.0
    DO 340 J=1,N
340 V(I)=V(I)+AT(I,J)*R(J)
    VVV=0.0
    DO 350 I=1,N
350 VVV=VVV+WTK(I)*(V(I)*V(I))
    VVV=SQRT(VVV)
    WRITE(6,29) VVV
 29 FORMAT('0',8X,'|| AT(AF-G) ||=',E16.8)
600 RETURN
    END
```