An Algorithm for Computing Piecewise Constant Solutions to Fredholm Integral Equations *

V. V. Tchernyi
Department of numerical analysis, SRCC of the Moscow State University. Vorobyevy Gory, Moscow 119899, Russia.
E-mail: morozov@apvk.srcc.msu.su

A. V. Smirnova
Department of apllied mathemathics
Kuban State University
149 Stavropolskaya st., Krasnodar 350040, Russia

Abstract

We describe an algorithm for computing the discontinuous solutions of Fredholm integral equations of the first kind, stable with respect to perturbation of the right-hand side of equation. The algorithm is based on regularizing CG iterations and consists of two steps: approximating the breakpoints of the exact solution and solving the original equation using the obtained breakpoints. If the right-hand side is perturbed, smoothing via iterative spline-approximation method is applied. The stability with respect to right-hand side perturbations is obtained via applying the residual criterion both for smoothing and to stop the CG iterations.

1 Introduction

The problem of finding discontinuous solutions to Fredholm equations of the first kind arises in many applied fields, e.g. in geophysics [6]. There are many practically important problems with unknown boundaries of some layers and unknown characteristics of these layers. The problems of that type define operator equations with a discontinuous solution. The breakpoints of the solution correspond to the bounds of the layers, and

*Part of this work was carried out during a visit to Institute of Mathematical Modelling at the Danish Technical University, supported by the Danish Natural Science Research Council, Grant No. 11-0436-1

the value of the solution defines the quantitative characteristics of the media.

The present paper describes an algorithm for computing the discontinuous solutions, stable with respect to perturbations of the right-hand.

2 Statement of the problem

Consider the linear Fredholm equation of the first kind

$$Au = \int_a^b K(x,s)u(s)ds = f(x), \quad x \in [a,b] \tag{1}$$

$$A : U \to U, \quad U = L_2[a,b],$$

$$f(x) \in F, \quad u(s) \in U, \quad K(x,s) \in L_2[a,b] \times L_2[a,b],$$

$$s \in [a,b], \quad x \in [a,b].$$

Here $L_2[a,b]$ is the space of all functions measurable in Lebesgue's sence and possessing finite norm $\|f(s)\|_{L_2} = \left(\int_a^b (f(s))^2 ds\right)^{1/2}$.

Let the kernel of the integral operator be symmetric $K(x,s) = K(s,x)$; this guarantees that the operator A is self-conjugate in $L_2[a,b]$. Suppose the exact solution $\hat{u}$ of equation (1) exists, is unique and possesses a finite number of breakpoints. Also assume that both left and right limits of the solution in the breakpoints are finite, and that the derivative of the exact solution exists everywhere on $[a,b]$ except the breakpoints.

The problem of finding the solution of (1) in $L_2[a,b]$ is ill-posed because the solution is unstable with respect to small perturbations of the right-hand side $f(x)$.

3 A CG-based algorithm for computing noncontinuous solutions

Let the operator be seft-conjugate and positive, i.e. $(Au,u) > 0$ for all $u \in U : \|u\| \neq 0$. Then the problem (1) is equivalent to minimizing the quadratic functional [7]

$$J(u) = (Au,u) - 2\,(u,f)\,. \tag{2}$$

We apply the CG method for minimizing $J(u)$ (see, for example, [2]):

$$u_{k+1} = u_k - \tau_k p_k, \quad p_{k+1} = g_{k+1} + \beta_k p_k,$$

$$g_{k+1} = Au_{k+1} - f, \quad p_0 = g_0 = Au_0 - f, \;\; k = 0, 1, \dots \tag{3}$$

$$\tau_k = \frac{(g_k, p_k)}{(Ap_k, p_k)}, \quad \beta_k = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}.$$

Usually the CG method gives smooth approximations to the exact solution, “seizing” its structure. We describe a modification of the computation scheme that allows to obtain approximations from the same class of the discontinuous functions as the exact solution belongs to.

Suppose that near any breakpoint the smooth approximation is more steep than elsewhere. Then the derivative of that approximation must have a local extremum near the breakpoint, and the absolute value of extremum must surpass the absolutie value of local extremums elsewhere.

Under that assumption we have to pick out the local extremums, that correspond to the breakpoints and thus define approximations to the points of discontinuity. The proposed method for picking out the breakpoints is close to method, described in [3], [4].

Let us define the closed convex set M as the set of positive functions, in any point of the segment being greater then or equal to given threshold T:

$$M = \{u(s) \in L_2[a, b] : u(s) \geq T \;\forall s \in [a, b]\}, \quad T \geq 0. \tag{4}$$

The operator, projecting arbitrary function $y(s) \in L_2[a, b]$ onto the set M will have the form

$$P_M(y) = \begin{cases} y(s), & y(s) \geq T, \\ T, & y(s) < T. \end{cases} \tag{5}$$

Here the threshold T is some empirically chosen constant (see (16)).

The proposed method for solving the problem (1) consists of the following steps.

1. If the right-hand side of equation is perturbed, apply smoothing. Terminate the smoothing process via the residual criterion.

2. Compute the smooth approximation to the exact solution via the CG method. The CG process is terminated via the residual criterion.

3. Compute the first derivative of the obtained approximation.
4. Compute the projection of the derivative's absolute value onto the set M. Compute the points of local maximums of the projection. Treat the computed values as the approximations to the breakpoints of desired solution.
5. Solve the origin problem, using information about the breakpoints.

If we know the breakpoint positions, we can solve the problem using the operator of smooth representation. The proposed approach was described in paper [8] for the case when solution is continuous on $[a, b]$.

Suppose we found L breakpoints $z_1, z_2, \dots, z_L$, that define $L+1$ semi-intervals of continuity: $[a, z_1)$, $[z_1, z_2), \dots, [z_L, b]$. We shall represent the solution of the problem (1) in the form

$$\begin{aligned} u(s) &= B\left(u^{(0)}, u^{(1)}, \dots, u^{(L)}, v(\xi)\right) \\ &= \begin{cases} u^{(0)} + \int\limits_a^s v(\xi), & s \in [a, z_1), \\ u^{(0)} + u^{(1)} + \int\limits_a^s v(\xi), & s \in [z_1, z_2), \\ \vdots & \\ \sum\limits_{j=0}^{L} u^{(j)} + \int\limits_a^s v(\xi), & s \in [z_L, b]. \end{cases} \end{aligned} \tag{6}$$

Here $u^{(0)}$ is the value of the solution in the point a; $u^{(1)}, \dots, u^{(L)}$ are the values of solution jumps in the points $z_1, \dots, z_L$, respectively:

$$u^{(i)} = \lim_{x \to z_i + 0} u(s) - \lim_{x \to z_i - 0} u(s), \; i = 1, \dots, L.$$

Under representation (6) problem (1) takes the form

$$AB\left(u^{(0)}, u^{(1)}, \dots, u^{(L)}, v(\xi)\right) = f(x)$$

or in the integral form

$$\begin{aligned} & u^{(0)} \int\limits_a^{z_1} K(x,s)ds \\ + & \left(u^{(0)} + u^{(1)}\right) \int\limits_{z_1}^{z_2} K(x,s)ds + \ldots + \left(\sum_{j=0}^{L} u^{(j)}\right) \int\limits_{z_L}^{b} K(x,s)ds \\ + & \int\limits_a^b K(x,s) \int\limits_a^s v(\xi) d\xi ds = f(x), \quad x \in [a, b]. \end{aligned} \tag{7}$$

Here the unknowns are the function $v(\xi) \in L_2[a,b]$ and the $L+1$ values $u^{(0)}, u^{(1)}, \ldots, u^{(L)}$. The operator $B\left(u^{(0)}, u^1, \ldots, u^L, v(\xi)\right)$ is linear with respect to its arguments, and if we calculate the gradient of the linear functional

$$J_{bp}\left(u^{(0)}, u^1, \ldots, u^L, v(\xi)\right) = \left\| AB\left(u^{(0)}, u^1, \ldots, u^L, v(\xi)\right) - f(x)\right\|^2 \quad (8)$$

with respect to unknown variables, we can minimize (8), thus solving the problem (1) using information about breakpoints. We stress that minimizatoin of J_{bp} in (8) is a linear problem.

4 Finite-dimensional approximation of problems with piecewise-constant solutions

For discretization of equation (1) we introduce the uniform grid on variables x and s: $x_i = a+(i-1)h$, $s_j = a+(j-1)h$, where $h = (b-a)/(N-1)$ is the step, and N is the number of nodes. Using trapezoidal quadrature rule, we approximate the integral equation by a system of linear equations

$$\tilde{A}u = \tilde{f}, \quad (9)$$

where $\tilde{A}$ is a square matrix with elements $a_{ij} = \gamma_j K(x_i, s_j)$, $i, j = 1 \ldots N$, and

$$\gamma_j = \begin{cases} \frac{h}{2}, & j = 1, \\ h, & j = 2 \ldots N-1, \\ \frac{h}{2}, & j = N, \end{cases} \quad (10)$$

are the quadrature coefficients.

4.1 Smoothing considerations

It follows from the definition of the CG method (3) that in step k the approximation u_k is defined as a linear combination of gradients at every step from 1 to k, $u_k = \sum_{j=1}^{k} c_j g_j$. Since the gradient of the functional (2) coincides with the residual of equation (1), the obtained CG approximation will contain the error component of the right-hand side, multipled by some coefficient. Hence we need to compute the derivative of the approximation in some stable way; one of the possibilities is to apply the iterative spline smoothing process [1] before actually computing the derivative. Potentially, there are two ways for organizing the process:

- Apply smoothing to the right-hand side $\tilde{f}$ of (9).
- Apply smoothing to the obtained approximation u_k.

When implementing both ways it was noticed, that applying smoothing process to the obtained approximation u_k is difficult since this approximation contains the error component from the Krylov space $\mathcal{K}_k(\tilde{A}, e) = span\{e, \tilde{A}e, \tilde{A}^2e, \ldots, \tilde{A}^ke\}$. Here e is the initial error component of the right-hand side, and $\tilde{A}$ is the matrix of the system (9). Difficulties with estimating the norm of the vector from the space $\mathcal{K}_k(\tilde{A}, e)$ force us to apply smoothing to the right-hand side.

Nevertheless, the question arises: how much does the vector, obtained after terminating the smoothing process, differ from the exact right-hand side vector, if the meansquare estimate of the error is δ? Let $\tilde{f} = \hat{f} + \delta\theta\hat{f}$, where $\tilde{f}$ is pertubed and $\hat{f}$ is the exact right-hand side and $\delta \in [0, 1]$ is an estimate of the error. We have $\|\tilde{f} - \hat{f}\|_2 \leq \delta\|\hat{f}\|_2 \leq \delta\|\tilde{f}\|_2$. Let f_s be the vector, obtained after terminating the smoothing process. If the stopping rule for the smoothing process is the residual criterion [1], the following estimate is guaranteed: $\|\tilde{f} - f_s\|_2 \leq \delta\|\tilde{f}\|_2$. Hence $\|\hat{f} - f_s\|_2 = \|\hat{f} - \tilde{f} + (\tilde{f} - f_s)\|_2 \leq \|\hat{f} - \tilde{f}\|_2 + \|\tilde{f} - f_s\|_2 \leq 2\delta$. The obtained estimate of $\|\hat{f} - f_s\|_2$ can be used in the residual criterion for stopping the CG iterations.

For smoothing of the right-hand side of (9) we shall use the iterative smoothing method for linear operator equations of the first kind [1]. The right-hand side vector in equation (9) is formed according to the rule

$$\tilde{f}_i = (1 + \delta\theta)\hat{f}_i, \tag{11}$$

where θ is the uniformly distributed random value in $[-1, 1]$, and $\delta \in [0, 1]$ is the relative error level.

We shall use explicit spline-approximation operator for cubic spline on the uniform grid, mapping any vector $\tilde{f} \in \mathbf{R}^N$ onto the cubic spline $S(x)$ according to the rule

$$S(x) = \sum_{i=1}^{N} \tilde{f}_i s_{i-2}(x), \tag{12}$$

where $s_{i-2}(x)$ is the basis local cubic spline on the interval $[x_{i-2}, x_{i+2}]$.

Having discretized the spline $S(x)$ in the points $x_i,\ i = 1, \dots, N$, we get

$$S(x_i) = \frac{1}{6}\tilde{f}(x_{i-1}) + \frac{2}{3}\tilde{f}(x_i) + \frac{1}{6}\tilde{f}(x_{i+1}).$$

Now we need to define the value of $\tilde{f}(x)$ in the points x_0 and x_{N+1}. In case of a non-periodical function we can put these values $\tilde{f}(x_0) = \tilde{f}(x_1)$, $\tilde{f}(x_{N+1}) = \tilde{f}(x_N)$. We can write the result of applying the spline-approximation operator to the vector $\tilde{f}$ as $\mathcal{S}\tilde{f}$, where the matrix $\mathcal{S}$ is tridiagonal

$$\mathcal{S} = \begin{bmatrix} 5/6 & 1/6 & & & & \\ 1/6 & 2/3 & 1/6 & & 0 & \\ & 1/6 & 2/3 & 1/6 & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & 1/6 & 2/3 & 1/6 \\ & & & & 1/6 & 5/6 \end{bmatrix}.$$

The iterative smoothing method consists of repeatedly applying the spline-approximation operator to the perturbed right-hand side. The process is terminated via the residual criterion; the final iteration number is

$$k_{stop} = \min\left\{i : \|\tilde{f} - \mathcal{S}^i\tilde{f}\|_2 \geq \delta\|\tilde{f}\|_2\right\},$$

where

$$\|\tilde{f}\|_2 = \left(\sum_{i=1}^{N} f_i^2\right)^{1/2}.$$

Thus, the vector f_s is formed as $f_s = \mathcal{S}^{k_{stop}}\tilde{f}$.

4.2 Representation of the piecewise-constant solution

Let the exact solution be a piecewise-constant function.

When we choose a piecewise constant function as the exact solution, we thus introduce additional information about desired solution. Now we can reformulate the last part of the algorithm described above.

Suppose we obtained L approximate breakpoints $z_1, \dots z_L$. Suppose each point z_i coincides with some node of the grid. These points divide the interval $[a, b]$ into $L+1$ subintervals in which the solution is constant. Define also points $z_0 = a$ and $z_{L+1} = b$. Let the value of the function be the value of its right limit, i.e. the desired function on the interval

$[z_{l-1}, z_l)$ is equal to y_l. Using the introduced grids on variables x and s, we can rewrite the original integral equation (1) in the form

$$\sum_{l=1}^{L+1} y_l \int_{z_{l-1}}^{z_l} K(x_i, s)ds = f(x_i), \quad i = 1, \ldots, N. \tag{13}$$

We can approximate the values of the integrals in the summation in (13) using the values $K(x_i, s_j)$, obtained via discretization of problem (1).

Problem (13) corresponds to an overdetermined system of linear equations

$$A_1 y = \tilde{f}, \tag{14}$$

where the matrix A_1 is $N \times L+1$ with elements

$$(A_1)_{il} = \int_{z_{l-1}}^{z_l} K(x_i, s)ds, \; i = 1, \ldots, N, \; l = 1, \ldots, L+1.$$

Having computed the least squares solution of (14), we thus define the desired function on $[a, b]$.

Taking stock of abovementioned, we can formulate the step 5 of the general algorithm for piecewise-constant solution in the following form.

5 Compute the values of the solution y_k in $L+1$ subintervals as the LS solution of system (14).

5 Numerical results

Having reduced the problem of solving the ill-conditioned linear system (9) to solving the system (14), it is natural to expect the direct least squares solution to be unacceptable. Nevertheless, if the number of breakpoints of the exact solution of the problem (1) is not very high, this is not the case, i.e. the matrix A_1 is well-conditioned. If the solution has many breakpoints, then A_1 may be ill-conditioned, and regularization (such as CG iterations) may be nessesary.

As model problem we use linear approximation to the inverse gravimetry problem [5]:

$$\int_{-2}^{2} \frac{1}{1+(x-s)^2} u(s)ds = f(x). \tag{15}$$

The matrix and the right hand side of the linear system (9) was formed as follows. Having discretized the integral operator in (15), we then compute the right-hand side as the value of the operator on the exact solution. After that the noise was added according to (11).

The described problem was considered with the exact solution of three types, shown in the figures by the solid line. In dependence on the solution type the problem is referred to as "problem A", "problem B" and "problem C". The numerical experiments show, that for each of the problems A, B and C the coordinates of the vector y, obtained via applying CG method for solving the system (14), practically coinside with those obtained via solving the system $A_1^* A_1 y = A_1^* \tilde{f}$ by means of QR factorization.

The threshold T, nessesary to construct the projector P_M, was chosen empirically. Papers [3] and [4] suggest using the value $T = 1.25\|\tilde{u}_{k_{stop}}\|_2$ with respect to signal processing problems, where $u_{k_{stop}}$ is the approximation of CG method (3), applied to the perturbed system (9) and terminated via the residual criterion. One can obtain acceptable results solving the problem described here when we use

$$T = 0.5 \left\| \Lambda \tilde{u}_{k_{stop}} \right\|_{L_2[a,b]}, \tag{16}$$

where Λ is the matrix, approximation the first derivative operator and $\|\cdot\|_{L_2[a,b]}$ is the energy norm, approximating the integral norm $L_2[a,b]$:

$$\|y\|_{L_2[a,b]} = \left(\sum_{i=1}^{N} \gamma_i y_i^2 \right)^{1/2}.$$

Here γ_i are the quadrature coefficients (10).

Fig. 1 shows the results of solving the model problems with no perturbation of the right-hand side $\delta = 0$, and the dotted lines depict the obtained approximations. The case $\delta = 0$ is included since the model problem requires regularization even if the right-hand side contains approximation errors only. If the breakpoints are located far from the end of the interval, the algorithm gives good approximations to both the positions of the breakpoints and the values of the solution. The quality of the approximation decreases if the breakpoint is located near the end of the interval (problem C). This seems to be a feature of the algorithm and should be corrected by changing the statement of the problem, if possible.

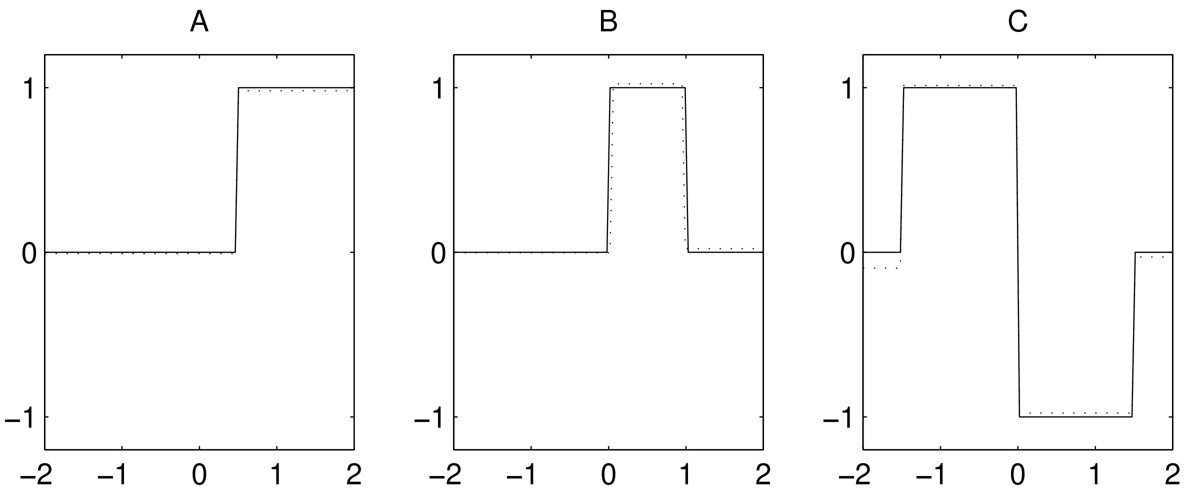

Figure 1. Results of the algorithm at $\delta = 0$.

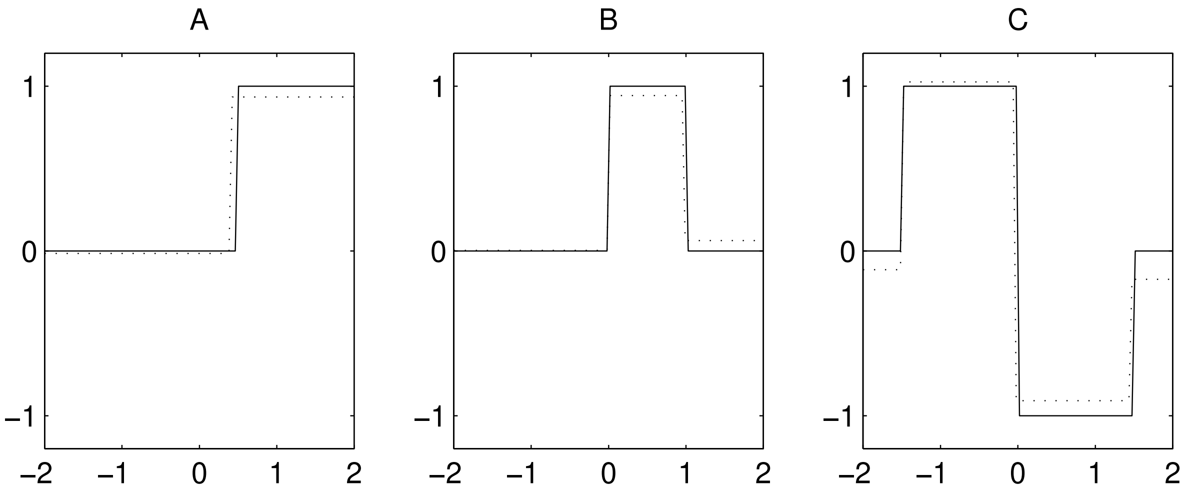

Figure 2. Results of the algorithm at $\delta = 0.05$, smoothing applied.

Fig. 2 shows the results of solving the model problems at $\delta = 0.05$ and smoothing applied. The approximations here show more deviation from exact solution, but still the values of the solution and the breakpoint positions are approximated good enough. This makes the algorithm potentially usable as a preprocessor for other regularization methods.

In general, the described algorithm seems to have some limitations and should only be applied to find piecewise-constant or slowly ascending (descending) solutions with a small number of breakpoints. If the solution grows fast by itself, one can obtain false breakpoints, i.e. some points could be identified as breakpoints only because the derivative of exact solution in that point has comparatively big value. Besides, the relation between the jumps of the solution in the breakpoints is important. All the jumps of the solution in the breakpoints must be comparable. Similar problems are described in the literature (cf., for example, [4], [3]) and require a further development of the algorithm.

The authors are grateful to V. A. Morozov for the statement of the problem and to S. F. Gilyasov and A. I. Grebennikov for valuable comments and discussion of the results. All computations were made in *MATLAB for Windows v.4.2.*

References

[1] **Morozov V. A., Grebennikov A. I.** Methods for solving incorrect posed problems. Algorithmic aspect. Moscow University publishing, 1992. In Russian.

[2] **Gilyasov S. F.** Approximate solving of ill-posed problems. Theory and algorithms. Moscow University publishing, 1995. In Russian.

[3] **Figueiras-Vidal A. R., Docampo-Amoedo D., Casar-Corredera J. R.** and **Artés-Rodrigues A.**, Adaptive Iterative Algorithms for Spiky Deconvolution, IEEE *Trans. on Acoustic, Speech and Signal Processing* ASSP–38 (8) August 1990, 1462–1466.

[4] **Sánchez-Avila C.**, An Adaptive Regularized Method for Deconvolution of Signals with Edges by Convex Projections, IEEE *Trans. on Signal Processing* SP–42 (7) July 1994, 1849–1851.

[5] **Tikhonov A. N., Arsenin V. Ya.** Methods for solving ill-posed problems. Nauka, Moscow, 1986. In Russian.

[6] **Starostenko V. I.** Stable numerical methods in gravimetry problems. Kiev, Naukova Dumka, 1978. In Russian.

[7] **Mikhlin S. T.** Variational methods in mathematical physics. Second edition. Nauka, Moscow, 1970. In Russian.

[8] **Morozov V. A., Gilyasov S. F., Tchernyi V. V.** On solving integral Fredholm equation via CG method. To appear in V. A. Morozov (Ed.), Numerical methods in analysis. Moscow State University, 1997. In Russian.