Computer Physics Communications 36 (1985) 423–432
North-Holland, Amsterdam

A PROGRAM FOR SOLVING FIRST KIND FREDHOLM INTEGRAL EQUATIONS BY MEANS OF REGULARIZATION

H.J.J. te RIELE

Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Received 5 February 1985

PROGRAM SUMMARY

Title of program: F1REGU

Catalogue number: AABX

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Computer: CDC CYBER 175-750; *Installation:* SARA (Academic Computer Centre Amsterdam)

Operating system: NOS/BE

Programming language: FORTRAN 77

Program Library used: NAG FORTRAN LIBRARY, MARK 10 (SUBROUTINE F04ASF; this, in turn, uses other NAG routines)

Programming language: FORTRAN 77

High speed storage requested: 60 K

No. of bits in a word: 60

Peripheral used: line printer

No. of lines in combined program and test deck: 321

Keywords: Fredholm integral equation of the first kind, regularization, elastic electron–atom scattering, dispersion relation

Nature of the physical problem
Fredholm integral equations of the first kind arise in the mathematical analysis of many physical problems (cf. Nedelkov [1]). An important characteristic of such problems is that the information which we seek about a physical quantitiy A can only be obtained *indirectly*, by measuring some other quantity B which has some connection with A. Often, this connection can be expressed mathematically in terms of a Fredholm first kind integral equation.

Method of solution
The first kind Fredholm integral equation is solved by means of the regularization method of Tihonov [2,3] and Phillips [4].

Running time
Approximately proportional to the third power of the number of data points.

References

[1] I.P. Nedelkov, Comput. Phys. Commun. 4 (1972) 157.
[2] A.N. Tihonov, Soviet Math. Dokl. 4 (1963) 1035, 1624.
[3] A.N. Tihonov and V.Y. Arsenin, Solution of Ill-posed Problems (Winston, Washingon, DC, 1977).
[4] D.L. Phillips, J. ACM 9 (1962) 84.

LONG WRITE-UP

1. Introduction

The linear first kind Fredholm integral equation

$$\int_a^b K(x, y) f(y) \mathrm{d}y = g(x), \quad c \leqslant x \leqslant d, \tag{1.1}$$

where f is the unknown function, and g and K are given functions, arises in the mathematical analysis of problems from many branches of physics, chemistry and biology [3]. Also several classical mathematical problems, like the problem of harmonic continuation, numerical inversion of the Laplace transform, the backwards heat equation, and numerical differentiation, can be formulated as equations of the form (1.1).

We assume that f and g are elements of certain linear spaces F and G, respectively. Defining the linear operator $\mathscr{K}: F \to G$ by $(\mathscr{K}f)(x) := \int_a^b K(x, y) f(y) \mathrm{d}y$, we write (1.1) in operator notation as:

$$\mathscr{K}f = g,\ g \in G \text{ given},\ f \in F \text{ sought}. \tag{1.2}$$

In general, numerical solution of (1.1) is difficult, because (1.1) belongs to the class of so-called *ill-posed*, or *improperly posed* problems. The problem (1.2) is ill-posed (in the sense of Hadamard, cf. ref. [2]) if at least one of the following three assertions is *false* (F and G are assumed to be complete metric spaces):

(i) for every $g \in G$ there exists a solution $f \in F$;
(ii) the solution of (1.2) is unique;
(iii) the solution of (1.2) depends continuously on the data g.

Note that this definition depends on the spaces F and G. A problem may be ill-posed with respect to given F and G, but well-posed in other metrics. In general, (1.2) is ill-posed because the solution f of (1.2) does not depend continuously on the data function g. This may be explained, at least heuristically, as follows. If K is a smooth function, then $\mathscr{K}$ is a smoothing operator and small perturbations in g may be caused by large perturbations in f, which were smoothed down by $\mathscr{K}$.

In practical situations, the data function g is often the output of some measuring process, so that it is only approximately known in some discrete set of points $x_i \in [c, d]$. Consequently, rather than (1.2) it is more realistic to consider the problem.

$$\mathscr{K}f = \tilde{g}, \tag{1.3}$$

where only $\tilde{g}$ and ϵ are known such that $\| \tilde{g} - g \| \leqslant \epsilon$ for some norm $\| \cdot \|$. This may cause $\tilde{g}$ to lie outside the range of the operator $\mathscr{K}$, so that there may not exist a solution of (1.3).

2. The regularization method

A survey of numerical methods for solving (1.1)–(1.3) may be found in refs. [5,11]. Here, we describe a simple implementation of the so-called *regularization* method of Phillips [4] and Tihonov [9–11]. This method essentially consists of the replacement of the ill-posed problem (1.3) by the well-posed problem (i.e. for which the three assertions (i), (ii) and (iii) above are *true*):

Minimize the quadratic functional $\phi_a(f)$, defined by

$$\phi_a(f) := \| \mathscr{K}f - \tilde{g} \|^2 + \alpha \| Lf \|^2, \tag{2.1}$$

over all functions f in the compact set:

$$\{ f: \| \mathscr{K}f - \tilde{g} \| \leqslant \epsilon \}.$$

Here, α is a fixed positive number, the so-called *regularization parameter* and L is some linear operator, e.g. $Lf = f$, f' or f'', or $Lf = f - \hat{f}$ if an a priori approximation $\hat{f}$ of f can be provided. If Lf is the ith derivative of f, then it is customary to speak about ith *order regularization*.

Under certain, mild conditions, (2.1) has a unique solution, which will be denoted by f_α. Moreover, f_α will converge as $\epsilon \to 0$, uniformly on $[a, b]$, to the solution of the equation $\mathscr{K}f = g$ (if it exists), provided that α satisfies

$$C_1 \epsilon^2 < \alpha < C_2 \epsilon^2 \tag{2.2}$$

for positive numbers C_1 and C_2. Unfortunately, g is not known exactly and the ill-posedness of (1.3) will, generally, cause the solution f_α of (2.1) to oscillate very wildly around the solution of the equation $\mathscr{K}f = g$, when α is chosen to be close to zero. An increase of α will result in an increase of the residual $\| \mathscr{K} f_\alpha - \tilde{g} \|$, and a decrease of the "penalty term" $\| Lf_\alpha \|$; and vice versa. For suitably chosen L the term $\| Lf_\alpha \|$ will have an increasing damping effect on unwanted oscillations of f_α, with increasing α.

The question then arises: how do we have to choose α? Up till now, this has not been resolved in a satisfactory way. The choice (2.2) may be of some use in practice. In any case, α should be chosen in such a way that *both* $\| \mathscr{K} f_\alpha - \tilde{g} \|$ *and* $\| Lf_\alpha \|$ (which, e.g., measures the smoothness of f_α in case $Lf = f''$) are acceptable to the user. Consequently, the proper choice of α depends considerably on the particular problem at hand.

3. The numerical solution of (2.1)

In order to solve (2.1) numerically, we introduce the following discretizations: we assume that $\tilde{g}(x)$ is given in N not necessarily equidistant points $x = x_i$, $i = 1,2,\dots,N$ ($c \leqslant x_1 < x_2 < \dots < x_n \leqslant d$) with $\tilde{g}(x_i) =: g_i$, and we split up the integration interval $[a, b]$ into N subintervals $[y_{j-1}, y_j]$, $j = 1, 2,\dots, N$ ($a = y_0 < y_1 < \dots < y_N = b$). The integrals $(\mathscr{K} f)(x)$ occurring in (2.1) are approximated, for any given $x = x_i$, by using the repeated mid-point rule:

$$
\begin{aligned}
(\mathscr{K} f)(x_i) &= \int_a^b K(x_i, y)\, f(y) \mathrm{d}y \\
&= \sum_{j=1}^{N} \int_{y_{j-1}}^{y_j} K(x_i, y)\, f(y) \mathrm{d}y \\
&\approx \sum_{j=1}^{N} K_{ij} f_j,
\end{aligned}
$$

where $K_{ij} := (y_i - y_{j-1})\, K(x_i, \bar{y}_j)$, $\bar{y}_j := \frac{1}{2}(y_{j-1} + y_j)$ and $f_j := f(\bar{y}_j)$ is an (unknown) approximation of f in the point $\bar{y}_j$. After defining $\hat{f}_j = \hat{f}(\bar{y}_j)$ as an a priori known estimate of f_j, $\epsilon_i := \sum_{j=1}^{N} K_{ij} f_j - g_i$, $i = 1, 2,\dots, N$, and writing

$$Lf := a_o(f - \hat{f}) + a_1 f' + a_2 f'',$$

where $a_i = 0$ or 1, we replace the *continuous* problem (2.1) by the discrete problem:

Minimize the quadratic functional $\bar{\phi}_\alpha(f)$, defined by

$$
\begin{aligned}
\bar{\phi}_\alpha(f) := \sum_{i=1}^{N} \epsilon_i^2 + \alpha \Bigg\{ & a_0 \sum_{j=1}^{N} \left(f_j - \hat{f}_j\right)^2 \\
& + a_1 \sum_{j=1}^{N-1} \left(f_{j+1} - f_j\right)^2 \\
& + a_2 \sum_{j=2}^{N-1} \left(f_{j+1} - 2f_j + f_{j-1}\right)^2 \Bigg\}
\end{aligned}
\tag{3.1}
$$

over all vectors $f = [f_1, f_2,\dots, f_N]^{\mathrm{T}} \in \mathscr{R}^N$ for which $\sum_{i=1}^{N} \epsilon_i^2 \leqslant \epsilon^2$.

From the necessary condition $\partial\bar{\phi}_\alpha / \partial f_j = 0$, $j = 1, 2,\dots, N$, we find, after some simple calculations, the linear matrix–vector equation:

$$
\begin{aligned}
&\left\{ K^{\mathrm{T}} K + \alpha\left(a_0 H_0 + a_1 H_1 + a_2 H_2\right) \right\} f \\
&\quad = K^{\mathrm{T}} g + \alpha a_0 \hat{f},
\end{aligned}
\tag{3.2}
$$

where $g = [g_1,\dots, g_n]^{\mathrm{T}}$, $\hat{f} = [\hat{f}_1,\dots, \hat{f}_N]^{\mathrm{T}}$, $K = (K_{ij})$, $K^{\mathrm{T}} = (K_{ji})$, $H_0 = I_N$ (the $N \times N$ identity matrix),

$$
H_1 = \begin{bmatrix}
1 & -1 & & & & 0 \\
-1 & 2 & -1 & & & \\
 & -1 & 2 & -1 & & \\
 & & \ddots & \ddots & \ddots & \\
 & 0 & & -1 & 2 & -1 \\
 & & & & -1 & 1
\end{bmatrix}_{N \times N},
$$

$$
H_2 = \begin{bmatrix}
1 & -2 & 1 & & & & & \\
-2 & 5 & -4 & 1 & & & 0 & \\
1 & -4 & 6 & -4 & 1 & & & \\
 & 1 & -4 & 6 & -4 & 1 & & \\
 & & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & 0 & & 1 & -4 & 6 & -4 & 1 \\
 & & & & 1 & -4 & 5 & -2 \\
 & & & & & 1 & -2 & 1
\end{bmatrix}_{N \times N}
$$

The linear symmetric system (3.2) is solved by using the standard NAG-Library routine F04ASF.

4. Description of the program

The main program calls a subroutine F1REGU which solves the minimization problem (3.1). The heading of this subroutine reads as follows:

```
SUBROUTINE F1REGU (KERNEL, N, X, G,
  Y, ALFA, LINFUN, F, RES)
EXTERNAL KERNEL
REAL KERNEL, X(N), G(N), Y(0:N), RES(6)
```

The parameters of F1REGU are:

KERNEL: a user-supplied external function which delivers the value of the kernel function K in the point (x, y) for any x in the interval $[c, d]$ and any y in the interval $[a, b]$;

N: the number of data points for which g is given and for which approximations to f are to be found; the maximum number allowed is 64;

X(N), G(N): arrays containing, on entry, the abscissae $x_1, \ldots, x_N$ and the corresponding data values $g_1, \ldots, g_N$;

Y(0:N): array of length N + 1 containing, on entry, a subdivision of $[a, b]$;

ALFA the regularization parameter, to be supplied by the user;

LINFUN: with this parameter, the user monitors the choice of the linear functional L:
LINFUN $= 1$: $Lf = f - \hat{f}$,
$= 2$: $Lf = f'$,
$= 3$: $Lf = f''$;

F(N) array of length N which, on exit, contains approximations to the solution f in the midpoints $\bar{y}_j$; if LINFUN = 1 then, on entry, the user most provide in F an a priori estimate of the solution in these midpoints;

RES(6) array containing, on exit, the following information: ($\| \cdot \|$ is the discrete L_2-norm)
RES (1) = $\| f - \hat{f} \|$,
RES (2) = $\| f' \|$,
RES (3) = $\| f'' \|$,
RES (4) = $\| Kf - g \|$,
RES (5) = minimum absolute value of the components of $Kf - g$,
RES (6) = maximum absolute value of the components of $Kf - g$.

5. Workspace

F1REGU uses 8448 words blank common workspace to be declared in the main program as follows:

```
COMMON K (64, 64), MAT (64, 64), RHS (64),
  WK1 (64), WK2 (64), FH (64)
REAL K, MAT, RHS, WK1, WK2, FH
```

6. Test-examples

The subroutine F1REGU has been tested on the following problem with known solution:

$$K(x, y) = (x + y)^{-1}, \quad g(x) = x^{-1} \ln\left[\frac{1 + x/a}{1 + x/b}\right],$$
$$f(y) = y^{-1},$$
$$[a, b] = [c, d] = [1, 5], \quad N = 16, 32,$$
$$x_i = 1 + (i-1) * h_1,$$
$$i = 1, 2, \ldots, N, \ h_1 = 4/(N-1), \tag{6.1}$$
$$y_i = 1 + i * h_2, \quad i = 0, 1, \ldots, N, \ h_2 = 4/N,$$
$$\alpha = 10^{-r}, \ r = 0, 1, \ldots, 14.$$

Table 1
Minimum number of correct digits obtained when solving problem (6.1)–(1.1) with subroutine F1REGU

α	Exact data		Perturbed data	
	$N = 16$	$N = 32$	$N = 16$	$N = 32$
1	0.1	0.1	0.1	0.1
10^{-1}	0.4	0.4	0.4	0.4
10^{-2}	0.5	0.5	0.4	0.4
10^{-3}	0.9	0.8	0.7	0.7
10^{-4}	1.1	1.1	0.7	0.7
10^{-5}	1.1	1.1	0.1 *	−0.2 *
10^{-6}	1.6	1.7		
10^{-7}	1.2	1.5		
10^{-8}	1.3	1.7		
10^{-9}	1.1	1.7		
10^{-10}	1.0	1.6		
10^{-11}	1.0	1.5		
10^{-12}	0.7 *	1.3		
10^{-13}	0.7 *	1.0 *		
10^{-14}	0.4 *	0.1 *		

* Numerical solution *not* monotonically decreasing.

Table 2
Perturbed values $\hat{g}(x_i)$ of $g(x_i)$, used in the test examples

$N=16$				$N=32$			
x_i	$g(x_i)$	$\hat{g}(x_i)$	error (%)	x_i	$g(x_i)$	$\hat{g}(x_i)$	error (%)
1.0000	0.510826	0.500345	−2.1	1.0000	0.510826	0.500345	−2.1
1.2667	0.467766	0.476891	2.0	1.1290	0.488975	0.498514	2.0
1.5333	0.431776	0.427548	−1.0	1.2581	0.469034	0.464441	−1.0
1.8000	0.401186	0.409976	2.2	1.3871	0.450752	0.460627	2.2
2.0667	0.374826	0.365242	−2.6	1.5161	0.433920	0.422825	−2.6
2.3333	0.351849	0.352851	0.3	1.6452	0.418367	0.419558	0.3
2.6000	0.331624	0.333124	0.5	1.7742	0.403946	0.405773	0.5
2.8667	0.313673	0.316543	0.9	1.9032	0.390533	0.394106	0.9
3.1333	0.297623	0.301847	1.4	2.0323	0.378022	0.383387	1.4
3.4000	0.283180	0.283277	0.0	2.1613	0.366322	0.366448	0.0
3.6667	0.270109	0.278030	2.9	2.2903	0.355354	0.365775	2.9
3.9333	0.258219	0.263798	2.2	2.4194	0.345050	0.352504	2.2
4.2000	0.247355	0.241919	−2.2	2.5484	0.335348	0.327978	−2.2
4.4667	0.237387	0.239189	0.8	2.6774	0.326197	0.328672	0.8
4.7333	0.228207	0.232738	2.0	2.8065	0.317549	0.323854	2.0
5.0000	0.219722	0.217096	−1.2	2.9355	0.309362	0.305665	−1.2
				3.0645	0.301600	0.305834	1.4
				3.1935	0.294230	0.293497	−0.2
				3.3226	0.287222	0.284772	−0.9
				3.4516	0.280549	0.283210	0.9
				3.5806	0.274188	0.273713	−0.2
				3.7097	0.268116	0.272964	1.8
				3.8387	0.262313	0.269326	2.7
				3.9677	0.256763	0.258344	0.6
				4.0968	0.251448	0.252259	0.3
				4.2258	0.246354	0.240569	−2.3
				4.3548	0.241466	0.237938	−1.5
				4.4839	0.236772	0.236306	−0.2
				4.6129	0.232261	0.228952	−1.4
				4.7419	0.227923	0.228027	0.0
				4.8710	0.223746	0.225888	1.0
				5.0000	0.219722	0.217514	−1.0

For the linear functional L we chose $Lf=f$ (zero-order regularization). The initial vector $\hat{f}$ was taken to be $\boldsymbol{0}$. In table 1 we give for each test combination of α and N the minimum number of correct digits obtained for f in the mid-points $\bar{y}_i = \frac{1}{2}(y_{i-1}+y_i)$, $i=1, 2,\dots, N$. Since the exact solution is monotonic decreasing, we have marked those cases by an asterisk (*) for which the numerical solution was *not* monotonically decreasing.

As a second test we have run the same problem with *perturbed* data g_i, obtained by multiplying $g(x_i)$, $i=1,2,\dots,N$, by the factor $1+0.03(2\rho-1)$ where ρ is a random number in the interval (0,1] generated by the FORTRAN random number generator. Consequently, the maximum perturbation in g is 3%. In order to facilitate reproduction of our tests, we give in table 2 the perturbed values $\hat{g}(x_i)$ of $g(x_i)$ used in our computations, together with the percentages of the perturbation. The results of the second test are given in the part of table 1 with heading "Perturbed data". For $\alpha < 10^{-5}$ the numerical values of f obtained were wildly oscillating and completely worthless.

In the case of exact data, the best results were obtained for values of α which lie in the range

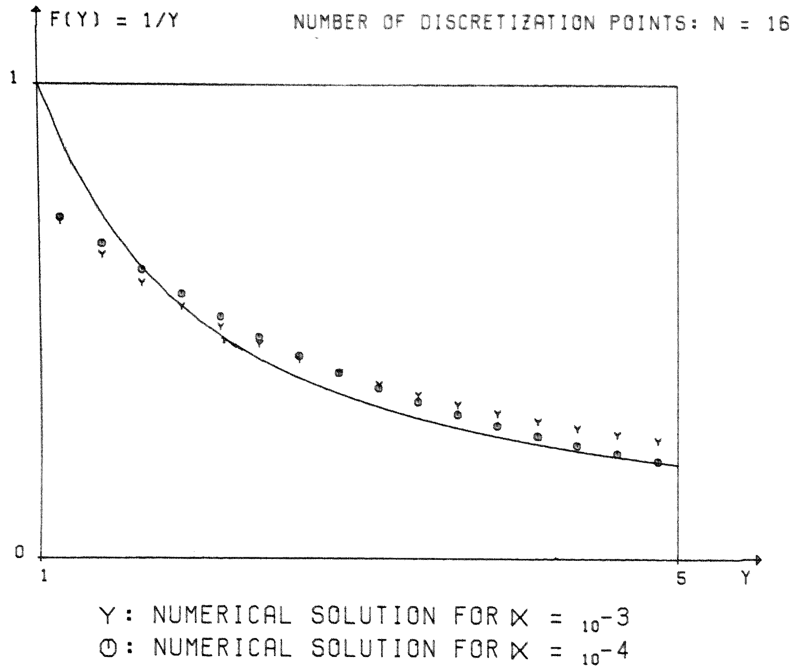

Fig. 1. Numerical solutions f_α for the cases $\alpha = 10^{-3}$ and 10^{-4}, $N = 16$.

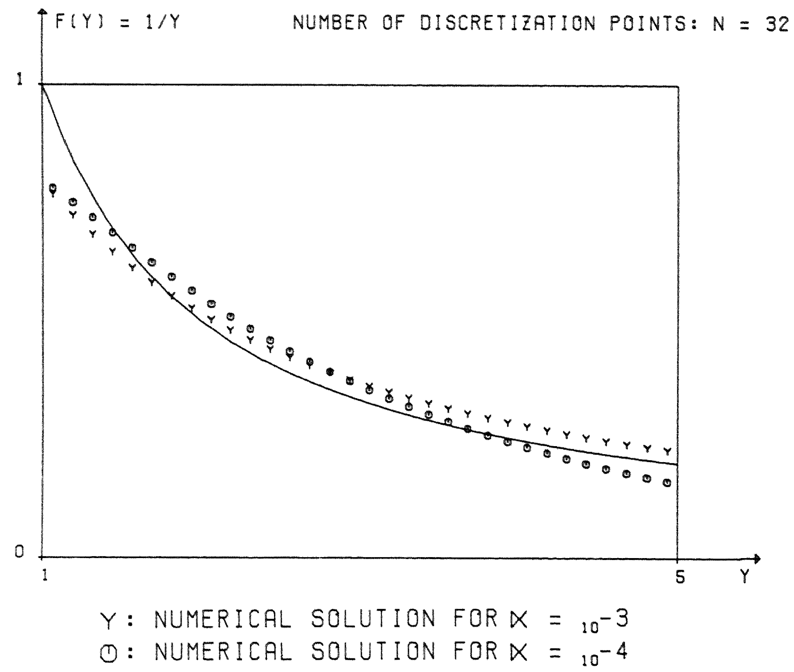

Fig. 2. Numerical solutions f_α for the cases $\alpha = 10^{-3}$ and 10^{-4}, $N = 32$.

$10^{-11} < \alpha < 10^{-4}$. Doubling the number N of discretization points has some effect only for very small values of $\alpha(< 10^{-8}$, say).

In the case of inexact data, the best results were obtained for $\alpha = 10^{-3}$ and $\alpha = 10^{-4}$. A maximal error of 3% corresponds, roughly, to $\epsilon = 0.03$ in (2.1). The values of α for which the best results were obtained agree reasonably well with the theoretical choice $\alpha = \mathcal{O}(\epsilon^2)$ expressed in (2.2).

In figs. 1 and 2 we present graphs of the numerical solutions f_α obtained in the cases $\alpha = 10^{-3}$, 10^{-4}, $N = 16$, 32, with inexact data. The drawn line is the exact solution.

The line printer output of the tests shown in figs. 1 and 2 is given below.

The subroutine F1REGU has also been used recently to solve a problem arising in elastic electron–atom scattering [8,12]. Some experiments with a (ALGOL 60) predecessor of F1REGU have been reported in ref. [5].

References

[1] J.J.M. Cuppen, ref. [6], p. 67.

[2] M.M. Lavrentiev, Springer Tracts in Natural Philosophy, vol. 11 (Springer, Berlin, 1967).

[3] I.P. Nedelkov, Comput. Phys. Commun. 4 (1972) 157.

[4] D.L. Phillips, J. ACM 9 (1962) 84.

[5] H.J.J. te Riele, in: Colloquium Numerieke Programmatuur, Deel 2, ed. H.J.J. te Riele (MC Syllabus 29.2, Amsterdam, 1977) p. 147 (in Dutch).

[6] H.J.J. te Riele, ed., MC Syllabus 41 (Math. Centre, Amsterdam, 1979).

[7] H.J.J. te Riele, ref. [6], p. 47.

[8] H.J.J. te Riele and R. Wagenaar, Proc. Conf. on Constructive Methods for the Practical Treatment of Integral Equations, eds. G. Hämmerlin and K.-H. Hoffmann, Oberwolfach, Fed. Rep. Germany (24–30 June 1984) (Birkhaüser Verlag, Basel, 1985) p. 224.

[9] A.N. Tihonov, Soviet Math. Dokl. 4 (1963) 1035.

[10] A.N. Tihonov, Soviet Math. Dokl. 4 (1963) 1624.

[11] A.N. Tihonov and V.Y. Arsenin, Solution of Ill-posed Problems (Winston, Washington, DC, 1977).

[12] R. Wagenaar, Doctor's Thesis, Amsterdam (1984).

TEST RUN OUTPUT

```
TEST OF F1REGU
ORDER OF REGULARIZATION= 0
NUMBER OF POINTS N=16  ALFA= .00010000
RESIDUES RES(1),...,RES(6)=
.1763E+01 .1416E+00 .1100E-01 .2256E-01 .1611E-04 .1103E-01
```

Y	F(Y) EXACT	F(Y) COMPUTED	NUMBER OF CORRECT DIGITS	ERROR PERCENTAGE
1.125000	.888889	.719862	.7	19.0
1.375000	.727273	.666015	1.1	8.4
1.625000	.615385	.611811	2.2	.6
1.875000	.533333	.560268	1.3	-5.1
2.125000	.470588	.512573	1.0	-8.9
2.375000	.421053	.469044	.9	-11.4
2.625000	.380952	.429594	.9	-12.8
2.875000	.347826	.393959	.9	-13.3
3.125000	.320000	.361803	.9	-13.1
3.375000	.296296	.332782	.9	-12.3
3.625000	.275862	.306564	1.0	-11.1
3.875000	.258065	.282844	1.0	-9.6
4.125000	.242424	.261347	1.1	-7.8
4.375000	.228571	.241827	1.2	-5.8
4.625000	.216216	.224069	1.4	-3.6
4.875000	.205128	.207880	1.9	-1.3

```
MINIMUM NUMBER OF CORRECT DIGITS:   .7

TEST OF F1REGU
ORDER OF REGULARIZATION= 0
NUMBER OF POINTS N=16  ALFA= .00100000
RESIDUES RES(1),...,RES(6)=
.1751E+01 .1367E+00 .1952E-01 .2297E-01 .3586E-03 .1283E-01
```

Y	F(Y) EXACT	F(Y) COMPUTED	NUMBER OF CORRECT DIGITS	ERROR PERCENTAGE
1.125000	.888889	.712591	.7	19.8
1.375000	.727273	.641722	.9	11.8
1.625000	.615385	.582561	1.3	5.3
1.875000	.533333	.532537	2.8	.1
2.125000	.470588	.489759	1.4	-4.1
2.375000	.421053	.452814	1.1	-7.5
2.625000	.380952	.420627	1.0	-10.4
2.875000	.347826	.392367	.9	-12.8
3.125000	.320000	.367380	.8	-14.8
3.375000	.296296	.345148	.8	-16.5
3.625000	.275862	.325257	.7	-17.9
3.875000	.258065	.307366	.7	-19.1
4.125000	.242424	.291201	.7	-20.1
4.375000	.228571	.276530	.7	-21.0
4.625000	.216216	.263164	.7	-21.7
4.875000	.205128	.250941	.7	-22.3

```
MINIMUM NUMBER OF CORRECT DIGITS:   .7
```

TEST OF F1REGU
ORDER OF REGULARIZATION= 0
NUMBER OF POINTS N=32 ALFA= .00010000
RESIDUES RES(1),...,RES(6)=
.2520E+01 .1196E+00 .4719E-02 .3017E-01 .3131E-03 .1149E-01

Y	F(Y) EXACT	F(Y) COMPUTED	NUMBER OF CORRECT DIGITS	ERROR PERCENTAGE
1.062500	.941176	.781930	.8	16.9
1.187500	.842105	.751169	1.0	10.8
1.312500	.761905	.719359	1.3	5.6
1.437500	.695652	.687305	1.9	1.2
1.562500	.640000	.655555	1.6	-2.4
1.687500	.592593	.624474	1.3	-5.4
1.812500	.551724	.594304	1.1	-7.7
1.937500	.516129	.565195	1.0	-9.5
2.062500	.484848	.537235	1.0	-10.8
2.187500	.457143	.510467	.9	-11.7
2.312500	.432432	.484903	.9	-12.1
2.437500	.410256	.460534	.9	-12.3
2.562500	.390244	.437333	.9	-12.1
2.687500	.372093	.415267	.9	-11.6
2.812500	.355556	.394294	1.0	-10.9
2.937500	.340426	.374369	1.0	-10.0
3.062500	.326531	.355443	1.1	-8.9
3.187500	.313725	.337470	1.1	-7.6
3.312500	.301887	.320402	1.2	-6.1
3.437500	.290909	.304191	1.3	-4.6
3.562500	.280702	.288792	1.5	-2.9
3.687500	.271186	.274162	2.0	-1.1
3.812500	.262295	.260259	2.1	.8
3.937500	.253968	.247042	1.6	2.7
4.062500	.246154	.234474	1.3	4.7
4.187500	.238806	.222519	1.2	6.8
4.312500	.231884	.211143	1.0	8.9
4.437500	.225352	.200314	1.0	11.1
4.562500	.219178	.190001	.9	13.3
4.687500	.213333	.180177	.8	15.5
4.812500	.207792	.170815	.7	17.8
4.937500	.202532	.161889	.7	20.1

MINIMUM NUMBER OF CORRECT DIGITS: .7

TEST OF F1REGU
ORDER OF REGULARIZATION= 0
NUMBER OF POINTS N=32 ALFA= .00100000
RESIDUES RES(1),...,RES(6)=
.2486E+01 .1138E+00 .9113E-02 .3137E-01 .2840E-04 .1277E-01

Y	F(Y) EXACT	F(Y) COMPUTED	NUMBER OF CORRECT DIGITS	ERROR PERCENTAGE
1.062500	.941176	.768431	.7	18.4
1.187500	.842105	.723721	.9	14.1
1.312500	.761905	.683226	1.0	10.3
1.437500	.695652	.646404	1.2	7.1
1.562500	.640000	.612803	1.4	4.2
1.687500	.592593	.582039	1.7	1.8
1.812500	.551724	.553787	2.4	-.4
1.937500	.516129	.527766	1.6	-2.3
2.062500	.484848	.503736	1.4	-3.9
2.187500	.457143	.481492	1.3	-5.3
2.312500	.432432	.460850	1.2	-6.6
2.437500	.410256	.441655	1.1	-7.7
2.562500	.390244	.423769	1.1	-8.6
2.687500	.372093	.407068	1.0	-9.4
2.812500	.355556	.391447	1.0	-10.1
2.937500	.340426	.376811	1.0	-10.7
3.062500	.326531	.363074	1.0	-11.2
3.187500	.313725	.350161	.9	-11.6
3.312500	.301887	.338005	.9	-12.0
3.437500	.290909	.326545	.9	-12.2
3.562500	.280702	.315727	.9	-12.5
3.687500	.271186	.305501	.9	-12.7
3.812500	.262295	.295824	.9	-12.8
3.937500	.253968	.286655	.9	-12.9
4.062500	.246154	.277957	.9	-12.9
4.187500	.238806	.269698	.9	-12.9
4.312500	.231884	.261847	.9	-12.9
4.437500	.225352	.254377	.9	-12.9
4.562500	.219178	.247262	.9	-12.8
4.687500	.213333	.240480	.9	-12.7
4.812500	.207792	.234008	.9	-12.6
4.937500	.202532	.227828	.9	-12.5

MINIMUM NUMBER OF CORRECT DIGITS: .7