

# Solving Fredholm Integral Equations of the Second Kind in MATLAB

K. E. Atkinson\*  
Dept of Mathematics  
University of Iowa

L. F. Shampine†  
Dept of Mathematics  
Southern Methodist University

May 5, 2007

## Abstract

We present here the algorithms and user interface of a MATLAB program, **Fie**, that solves numerically Fredholm integral equations of the second kind on an interval  $[a, b]$  to a specified, modest accuracy. The kernel function  $K(s, t)$  is to be moderately smooth on  $[a, b] \times [a, b]$  except possibly across the diagonal  $s = t$ . If the interval is finite, **Fie** provides for kernel functions that behave in a variety of ways across the diagonal, viz.  $K(s, t)$  may be smooth, have a discontinuity in a low-order derivative, have a logarithmic singularity, or have an algebraic singularity. **Fie** also solves a large class of integral equations with moderately smooth kernel function on  $[0, \infty)$ .

**ACM Categories and Subject Descriptors:** G.1.9 [Numerical Analysis]: Integral Equations

**General Terms:** Algorithms

**Additional Keywords and Phrases:** numerical solution, MATLAB

## 1 Introduction

We present here the algorithms and user interface of a MATLAB program, **Fie**, for the numerical solution of integral equations of the form

$$\lambda x(s) - \int_a^b K(s, t)x(t) dt = f(s), \quad a \leq s \leq b \quad (1)$$

which we write symbolically as  $(\lambda - \mathcal{K})x = f$ . **Fie** is a successor of the FORTRAN program IESIMP in [3] that extends greatly the range of problems that

---

\*Dept. of Mathematics, University of Iowa, Iowa City, IA 52242.

Email: atkinson@math.uiowa.edu

†Dept. of Mathematics, Southern Methodist University, Dallas, TX 75275.

Email: lshampin@mail.smu.edu

can be treated. Despite the additional capabilities in **Fie**, we have been able to exploit the MATLAB [12] problem solving environment (PSE) to make it notably easier to use than IESIMP.

The **Fie** program solves integral equations with kernel functions that are moderately smooth on  $R = [a, b] \times [a, b]$ , except possibly across the diagonal  $s = t$ . When the interval is finite, we assume that the equation has a unique solution for any continuous  $f(s)$ . **Fie** provides for kernels with the following kinds of behavior across the diagonal:

- **Smooth**  $K(s, t)$  is moderately smooth on all of  $R$ . **Fie** uses Simpson's rule to discretize the equation and the natural interpolant of Nyström to get an approximate solution on all of  $[a, b]$ . This case is discussed in §3. We remark that this is the class of kernels and the approach taken in IESIMP.
- **Discontinuous**  $K(s, t)$  has a discontinuity in a low-order derivative across the diagonal. **Fie** uses a variant of Simpson's rule that accounts for the discontinuity and so achieves the same rate of convergence as Simpson's rule applied to a smooth kernel. This case is discussed in §4.
- **Logarithmic Singularity**  $K(s, t) = L(s, t) \log |s - t|$  for a function  $L(s, t)$  that is moderately smooth on all of  $R$ . **Fie** uses product integration to account for the singular behavior across the diagonal. In addition it uses a graded mesh to deal with a lack of differentiability of the solution at the ends of the interval. This case is discussed in §5.
- **Algebraic Singularity**  $K(s, t) = L(s, t)/|s - t|^\alpha$  for a function  $L(s, t)$  that is moderately smooth on all of  $R$  and  $0 < \alpha < 1$ . As in the case of a logarithmic singularity, **Fie** uses product integration to account for the singular behavior across the diagonal and a graded mesh to deal with a lack of differentiability of the solution at the ends of the interval. This case is discussed in §6.

**Fie** also solves problems set on infinite intervals, but only the interval  $[0, \infty)$ . In this case we assume that there is a unique solution for all  $f(s)$  that are bounded as well as continuous. **Fie** requires that the kernel is moderately smooth on all of  $R$  and the integral operator is compact on a suitable function space. It solves a problem of this kind by first transforming the equation to one posed on  $[0, 1]$ . A two-point Gaussian formula is used instead of Simpson's rule to avoid evaluating the kernel at infinity. This case is discussed in §??.

The Guide to Available Mathematical Software [9] lists only four programs for the solution of Fredholm integral equations. Two are the FORTRAN programs IESIMP and IEGAUS of [3] that solve equations with smooth kernels. The NAG FORTRAN library [14] has a program D05AAF for the solution of problems with kernels that are discontinuous like those of §4 and a modification of this program called D05ABF that solves equations with smooth kernels. All four of these programs are for problems on finite intervals.

We discuss our implementation of `Fie` in §2, including a discussion of the important capabilities of MATLAB that we use. Along with the solver itself, we have written a collection of test programs that show how to use the solver and illustrate its performance. In each of the sections where we discuss a problem class, we take up two examples from the corresponding test program. Section 2 includes a URL for all these programs and a note about the applicable versions of MATLAB.

## 2 Implementation

MATLAB has capabilities that we exploit to simplify the user interface of the solver `Fie`. In this section we discuss some aspects of the design of the code. Problem solving environments make it easy to formulate problems, solve them numerically, and interpret results graphically. The price of this convenience is speed. For this reason we give some attention to efficient computation in the PSE. The environment makes modest accuracy natural, which led us to implement numerical methods of relatively low order.

A call to `Fie` has the form

```
[sol,errest,cond] = Fie(lambda,a,b,behavior,kernel,RHS,...
                        AbsTol,RelTol)
```

With the exception of `behavior`, the first six input arguments provide a straightforward description of the integral equation (1). The kernel  $K(s, t)$  must be moderately smooth on all of  $R = [a, b] \times [a, b]$  except possibly for the diagonal  $s = t$ . When the interval is finite, the input variable `behavior` tells the solver how the kernel behaves across the diagonal. The possibilities are

- 1  $K(s, t)$  is smooth.
- 2  $K(s, t)$  is discontinuous in a low-order derivative.
- 3  $K(s, t) = L(s, t) \log |s - t|$ .
- `alpha`  $K(s, t) = L(s, t)/|s - t|^\alpha$  for  $0 < \alpha < 1$ .

The interval  $[0, \infty)$  is specified in a natural way, namely by `a = 0` and `b = Inf`, which is the way that MATLAB represents infinity. The PSE has a built-in function that `Fie` uses to test whether `b` is infinite. When the interval is infinite, the kernel must be smooth across the diagonal, so the input value of `behavior` is ignored.

Vectorization is very important to efficiency in MATLAB, so we require that the functions defining the equation be coded to accept array arguments and return arrays. The argument `RHS` is the handle of a function that is to accept a column vector `s` and return a column vector of values  $f(s)$ . The argument `kernel` is the handle of a function that is to evaluate  $L(s, t)$  if `behavior` is `3` or `alpha` and otherwise,  $K(s, t)$ . In either case the function is to accept matrices `s` and `t` defining a grid and return a matrix of corresponding function values.

The optional input arguments **AbsTol** and **RelTol** are absolute and relative error tolerances, respectively. **Fie** tries to find a solution  $z$  that approximates the true solution  $x$  on a set of nodes in  $[a, b]$  so that

$$\|x - z\|_\infty \leq atol + rtol \|z\|_\infty$$

Here  $atol = \max(AbsTol, 0)$  and  $rtol = \max(RelTol, 100 \text{ eps})$  are used to deal with improper or unreasonable values of **AbsTol** and **RelTol**. (The quantity **eps** is the unit roundoff in **MATLAB**.) It is intended that **Fie** solve integral equations to modest accuracy, so the default values for the tolerances are **AbsTol** = **1e-6** and **RelTol** = **1e-3**. The approximation to  $\|x - z\|_\infty$  that is used in this test is available as the optional output argument **errest**.

The PSE provides for a complex data structure that **Fie** uses for the output argument **sol**. The program computes the approximate solution on a set of nodes in  $[a, b]$ , but these values determine an approximate solution that is accurate throughout the interval  $[a, b]$ , namely the natural interpolant of Nyström. It is evaluated conveniently at an array of points **sint** by a call to **ntrpFie** of the form

```
xint = ntrpFie(sol,sint)
```

This interpolant requires information about the problem such as **lambda** and handles for the functions **kernel** and **RHS**. All this information is placed in the solution structure **sol** by **Fie**, making it very easy for a user to evaluate the interpolant. The nodes and the solution at the nodes are available as the fields **sol.s** and **sol.x**, but it is so easy to use **ntrpFie** that there is little reason to access these values directly.

Some of the capabilities of **MATLAB** make **Fie** easier to use and the coding much simpler than **IESIMP**. We have already discussed the use of complex data structures to facilitate evaluation of the numerical solution. Another capability is automatic and dynamic allocation of storage. Array operations not only simplify and clarify the coding, but some actually reduce the run time because they are executed as compiled code rather than the usual interpreted code of the PSE. It was necessary to supply **IESIMP** with auxiliary functions such as a linear equation solver that are already present in the PSE. Like many of the built-in functions, the linear equation solver of **MATLAB** is executed as compiled code with a corresponding reduction in run time.

**Fie** is not intended for eigenvalue problems and no special provision is made for values of  $\lambda$  that are close to eigenvalues. The program approximates the integral equation by a system of linear equations  $Az = c$  where the components of  $c$  are values of  $f(s)$  and the components of  $z$  are approximations to  $x(s)$  at certain nodes in  $[a, b]$ . The optional output argument **cond** is an inexpensive estimate of the condition of  $A$  computed with a built-in function. It provides a way of recognizing the ill-conditioning of the integral equation that results from a  $\lambda$  that is close to an eigenvalue.

Along with the programs for solving Fredholm integral equations of the second kind, we also provide a collection of test programs, one for each kind of

behavior allowed. They illustrate the use of `Fie` and `ntrpFie` and show how well they perform. Each will be discussed below in context, but we note here that each begins with comments explaining the use of the program and an example call. The program `TestAll` collects these example calls and so runs through all the possibilities. Each test program allows the user to specify not just a problem, but also values of  $\lambda$  and  $[a, b]$ . In some cases, the problem depends on a parameter  $c$  that the user can choose. The test program displays the computed solution and reports `cond`, an estimate of the condition of the problem, as well as several measures of error. One is `errest`, an estimate of the maximum absolute error at nodes. The true error at the nodes is determined and reported for comparison with the estimate. To assess the error of the interpolant, the solution is evaluated at 150 points in the interval and the maximum true error at these points is reported.

`Fie` and `ntrpFie` make use of nested functions, a feature that was added to `Matlab` at version 7.0 (May 2004). All the programs are available at

[www.math.uiowa.edu/ftp/atkinson/Fie.package/](http://www.math.uiowa.edu/ftp/atkinson/Fie.package/)

### 3 Smooth Kernel

We begin with the approximation of the integral operator

$$\mathcal{K}x(s) = \int_a^b K(s, t)x(t) dt, \quad a \leq s \leq b \quad (2)$$

when the kernel  $K(s, t)$  is smooth. The numerical method that we describe and implement converges when  $K$  is just continuous, but it works best when  $K(s, t)$  is at least three times continuously differentiable. A standard approach is to discretize the integral with a quadrature scheme

$$\int_a^b g(t) dt \approx \sum_{j=0}^n w_{j,n} g(t_{j,n})$$

for a sequence of values of  $n \rightarrow \infty$ . The quadrature points  $\{t_{j,n} \mid j = 0, \dots, n\}$  are contained in the interval  $[a, b]$ . For smooth kernels we use the composite Simpson's rule for an even integer  $n$ . Using these nodes and weights, we approximate  $\mathcal{K}x(s)$  by

$$\mathcal{K}_n x(s) \equiv \sum_{j=0}^n w_{j,n} K(s, t_{j,n}) x(t_{j,n}), \quad a \leq s \leq b \quad (3)$$

and then approximate  $(\lambda - \mathcal{K})x = f$  by

$$\lambda x_n(s) - \sum_{j=0}^n w_{j,n} K(s, t_{j,n}) x_n(t_{j,n}) = f(s), \quad a \leq s \leq b \quad (4)$$

or more abstractly,  $(\lambda - \mathcal{K}_n)x_n = f$ . This is solved by first finding the values of  $x_n$  at the nodes  $\{t_{j,n}\}$ . Collocating (4) at the nodes leads to the linear system

$$\lambda x_n(t_{i,n}) - \sum_{j=0}^n w_{j,n} K(t_{i,n}, t_{j,n}) x_n(t_{j,n}) = f(t_{i,n}), \quad i = 0, 1, \dots, n \quad (5)$$

After solving this system, the general solution to (4) can be obtained at all other points  $s \in [a, b]$  by solving (4) to get the Nyström interpolation formula,

$$x_n(s) = \frac{1}{\lambda} \left[ f(s) + \sum_{j=0}^n w_{j,n} K(s, t_{j,n}) x_n(t_{j,n}) \right], \quad a \leq s \leq b \quad (6)$$

A complete presentation of the Nyström method, including an analysis of its error is given in [4, §4.1]. It is found that for all sufficiently large  $n$ , the approximating equation (4) has a unique solution and there is a constant  $c > 0$  such that

$$\|x - x_n\|_\infty \leq c \|\mathcal{K}x - \mathcal{K}_n x\|_\infty \quad (7)$$

The quantity on the right here is the error of the approximation (3).

The scheme outlined above and using Simpson's rule is implemented in the FORTRAN program IESIMP [2, §5.1] and [3], and it is used in the MATLAB program **File**. If  $K(s, t)$  is continuous on  $[a, b] \times [a, b]$ , then for all sufficiently large values of  $n$  the equation (4) has a unique solution  $x_n$ . Moreover, we find with Simpson's rule that if  $K(s, \cdot)x \in C^4[a, b]$  for  $a \leq s \leq b$ , the bound (7) implies

$$\|x - x_n\|_\infty = O(n^{-4}) \quad (8)$$

The new program **File** begins with  $n = 8$  and doubles  $n$  until a solution  $x_n$  is found with sufficient accuracy or an upper limit of 512 is reached. It predicts the error assuming a geometric behavior

$$\|x - x_n\|_\infty = O(n^{-p}) \quad (9)$$

for some  $p \geq 1$ . As seen in (8), for moderately smooth kernel functions this will be true with  $p = 4$ . Using three successive values of  $n$ , say

$$n_1, \quad n_2 = 2n_1, \quad n_3 = 2n_2$$

we can estimate  $p$  using

$$2^{-p} \approx \frac{\|x_{n_3} - x_{n_2}\|_\infty}{\|x_{n_2} - x_{n_1}\|_\infty} \equiv \text{ratio} \quad (10)$$

With this we have the standard estimate

$$\|x - x_{n_3}\|_\infty \approx \frac{\text{ratio}}{1 - \text{ratio}} \|x_{n_3} - x_{n_2}\|_\infty \quad (11)$$

The reliability of the estimate is enhanced by restricting the value of *ratio* so that

$$\frac{1}{16} \leq \text{ratio} \leq \frac{1}{2} \quad (12)$$

The lower value here is consistent with (8).

The `TestSmooth` program has eight test problems with smooth kernels. There are more possibilities than is apparent because as explained in §2, the finite interval and the parameter  $\lambda$  are to be specified. Furthermore, some problems are families depending on a parameter  $c$  that is also to be specified. Depending on the problem, there may be restrictions on the values allowed for these variables. Two of the test problems will illustrate the possibilities and the performance of the solver for problems with smooth kernels. Note that the numbering of problems, both here and in the following sections, is taken from the problem index of the corresponding test program.

**Problem 4** This is a standard test problem with the Runge kernel. It arises in electrostatics [11] where it is called Love's equation. The kernel function has a peak along  $s = t$  when the positive parameter  $c$  is small. The equation is

$$\lambda x(s) - \int_0^1 \frac{c x(t)}{c^2 + (s-t)^2} dt = f(s), \quad 0 \leq s \leq 1 \quad (13)$$

with  $f(s)$  defined so that  $x(s) = 0.06 - 0.8s + s^2$ . For the numerical results reported here,  $c = 0.1$  and  $\lambda = -0.5$ . It is seen in Table 1 that `errest` is a close bound on the actual error at the nodes and the accuracy of the solution at the nodes is preserved throughout the interval by the Nyström interpolant. As a side note, this kernel function approaches a singular function as  $c$  decreases to 0, and thus a larger value of  $n$  is needed in order to approximate the integral accurately. The norm of the integral operator is, however, always bounded by  $\pi$  when considered as a linear operator on  $C[0, 1]$  into  $C[0, 1]$ .

**Problem 8** This example in [13, Chap. 4] is a boundary integral equation for a plane interior Dirichlet problem for an ellipse. For  $f(s) = 25 - 16 \sin^2(s)$  the equation

$$x(s) + \frac{1}{\pi} \int_{-\pi}^{\pi} \frac{0.3 x(t)}{1 - 0.64 \cos^2(\frac{1}{2}(s+t))} dt = f(s), \quad -\pi \leq s \leq \pi \quad (14)$$

has the solution  $x(s) = \frac{17}{2} + \frac{128}{17} \cos(2s)$ . Because of periodicity, Simpson's rule converges faster than the assumed geometric rate (9). A consequence seen in Table 1 is that `errest` is a bound on the error at nodes, but it is quite pessimistic. Even with exceptionally accurate solution values at the nodes, the Nyström interpolant provides an equally accurate solution throughout the interval.

## 4 Discontinuous Kernel

In this section we consider kernels that are moderately smooth on all of  $[a, b] \times [a, b]$  except for a discontinuity in a low order derivative across the diagonal

Table 1: TestSmooth with default tolerances.

Problem	Final n	cond	Interpolation Error	Error at Nodes	
				errest	true
#4	64	16	6.6e-6	1.1e-5	6.6e-6
#8	32	4	1.1e-8	4.9e-5	1.1e-8

$s = t$ , a behavior typical of Green's functions. The method developed in §3 is considerably less satisfactory for such kernels because Simpson's rule is applied to subintervals where the integrand is not smooth. The integrand is piecewise smooth and we know where the lack of smoothness occurs, so we can get accurate approximations by developing an appropriate quadrature formula, an approach first given in Cubillos [6]. Another approach given by El-Gendi [7] is implemented in the FORTRAN program D05AAF of the NAG library [14]. A modification of this program called D05ABF solves equations with smooth kernels.

As with Simpson's rule in §3, we assume that  $n$  is an even integer, define  $h = (b - a) / n$ , and work with the nodes  $t_j = a + jh$  for  $j = 0, 1, \dots, n$ . In stating the formulas it will be convenient to denote the usual Simpson's rule applied to a function  $g(t)$  on the interval  $[\alpha, \beta]$  by  $\mathcal{S}\{g, \alpha, \beta\}$ . We need to approximate

$$\mathcal{K}x(s) \equiv \int_a^b K(s, t)x(t) dt = \sum_{\substack{j=1 \\ j \text{ odd}}}^{n-1} \int_{t_{j-1}}^{t_{j+1}} K(s, t)x(t) dt \quad (15)$$

When  $s$  is a node  $t_k$  with  $k$  even, we approximate the integrals on the right hand side of this expression with Simpson's rule. For any other  $s$ , suppose that  $i$  is the odd integer for which  $t_{i-1} \leq s \leq t_{i+1}$  and let  $\tilde{x}(t)$  be the quadratic polynomial that interpolates  $x(t)$  at both ends and the middle of this subinterval. For this  $s$ , we approximate  $\mathcal{K}x(s)$  by

$$\begin{aligned} \mathcal{K}_n x(s) \equiv & \sum_{\substack{j=1 \\ j \text{ odd}, j \neq i}}^{n-1} \mathcal{S}\{K(s, \cdot)x, t_{j-1}, t_{j+1}\} \\ & + \mathcal{S}\{K(s, \cdot)\tilde{x}, t_{i-1}, s\} + \mathcal{S}\{K(s, \cdot)\tilde{x}, s, t_{i+1}\} \end{aligned} \quad (16)$$

It is straightforward to show that in this approximation, when  $s$  approaches a node  $t_k$  with  $k$  even,

$$\lim_{s \rightarrow t_k} \mathcal{K}_n x(s) = \sum_{\substack{j=1 \\ j \text{ odd}}}^{n-1} \mathcal{S}\{K(t_k, \cdot)x, t_{j-1}, t_{j+1}\} \quad (17)$$

This tells us that the  $\mathcal{K}_n x(s)$  of (15) is continuous on all of  $[a, b]$ .



The approximation (16) results in a sum of the form

$$\mathcal{K}_n x(s) = \sum_{j=0}^n w_j(s) x(t_j), \quad a \leq s \leq b \quad (18)$$

with weight functions  $\{w_j(s)\}$  that are weighted combinations of values of  $K(s, \cdot)$ . This leads directly to

$$\lambda x_n(s) - \sum_{j=0}^n w_j(s) x_n(t_j) = f(s), \quad a \leq s \leq b \quad (19)$$

A standard convergence analysis provides the same kind of convergence result that we obtained in §3 for smooth kernels, namely that if  $x$  is sufficiently smooth and the kernel is sufficiently smooth except across  $s = t$ , then (19) has a solution for all sufficiently large  $n$  and

$$\|x - x_n\|_\infty = O(n^{-4})$$

We calculate  $x_n$  by solving the linear system

$$\lambda x_n(t_i) - \sum_{j=0}^n w_j(t_i) x_n(t_j) = f(t_i), \quad i = 0, \dots, n \quad (20)$$

For even index  $i$ , the coefficients  $\{w_0(t_i), \dots, w_n(t_i)\}$  are the usual weights of Simpson's rule, as seen in (17). For odd  $i$ , we have

$$\begin{aligned} \mathcal{K}_n x(t_i) \equiv & \sum_{\substack{j=1 \\ j \text{ odd}, j \neq i}}^{n-1} \mathcal{S}\{K(t_i, \cdot) x, t_{j-1}, t_{j+1}\} \\ & + \mathcal{S}\{K(t_i, \cdot) \tilde{x}, t_{i-1}, t_i\} + \mathcal{S}\{K(t_i, \cdot) \tilde{x}, t_i, t_{i+1}\} \end{aligned} \quad (21)$$

We see from this expression that for  $j < i - 1$  or  $j > i + 1$ , the weights  $w_j(t_i)$  are the usual weights of Simpson's rule, exactly as for even  $i$ . A straightforward calculation leads to the following expressions for the remaining weights:

$$\begin{aligned} w_{i-1}(t_i) &= \frac{h}{6} \left[ 3K(t_i, t_{i-1}) + \frac{3}{2}K(t_i, t_{i-1/2}) - \frac{1}{2}K(t_i, t_{i+1/2}) \right] \\ w_i(t_i) &= \frac{h}{6} \left[ 3K(t_i, t_{i-1/2}) + 2K(t_i, t_i) + 3K(t_i, t_{i+1/2}) \right] \\ w_{i+1}(t_i) &= \frac{h}{6} \left[ -\frac{1}{2}K(t_i, t_{i-1/2}) + \frac{3}{2}K(t_i, t_{i+1/2}) + 3K(t_i, t_{i+1}) \right] \end{aligned}$$

For  $t_{i-1} \leq s \leq t_{i+1}$ , the Nyström interpolant is

$$\begin{aligned} x_n(s) &= \frac{1}{\lambda} [f(s) + \sum_{\substack{j=1 \\ j \text{ odd}, j \neq i}}^{n-1} \mathcal{S}\{K(s, \cdot) x_n, t_{j-1}, t_{j+1}\} \\ & \quad + \mathcal{S}\{K(s, \cdot) \tilde{x}_n, t_{i-1}, s\} + \mathcal{S}\{K(s, \cdot) \tilde{x}_n, s, t_{i+1}\}] \end{aligned}$$

Here  $\tilde{x}_n$  is the quadratic polynomial interpolating  $x_n$  at the nodes  $t_{i-1}, t_i, t_{i+1}$ , namely

$$\tilde{x}_n(s) = x_n(t_{i-1}) + \mu \Delta x_n(t_{i-1}) + \frac{1}{2} \mu (\mu - 1) \Delta^2 x_n(t_{i-1})$$

In this expression  $\Delta$  is the forward difference operator and  $\mu = (s - t_{i-1})/h$ .

The `TestDiscontinuous` program has five test problems with kernels that are discontinuous in a derivative of low order across the diagonal. All the problems require the interval  $[0, 1]$  except for Problem #3, which allows an interval of the form  $[0, b]$ . Problem #4 requires  $\lambda = 1$ , but  $\lambda$  can be specified for the other problems. Problem #1 is a family that depends on a parameter  $c > 0$  that is also to be specified. Two of the test problems will illustrate the possibilities and the performance of the solver for problems with discontinuous kernels.

**Problem 4** This is an example in [8, Chap. 14]. The equation

$$x(s) - \int_0^1 |s - t| x(t) dt = -(2s^3 - 9s + 2)/6, \quad 0 \leq s \leq 1 \quad (22)$$

has the solution  $x(s) = s$ . With this solution and a quadrature scheme that accounts for the discontinuity in the first derivative across  $s = t$ , the numerical method of `File` is exact. Accordingly, the errors seen in Table 2 are all comparable to the unit roundoff of 2.2e-16.

**Problem 5** This is the example of the NAG documentation for D05AAF [14]. The solution of the equation

$$x(s) - \int_0^1 K(s, t) x(t) dt = \left(1 - \frac{1}{\pi^2}\right) \sin(\pi s), \quad 0 \leq s \leq 1 \quad (23)$$

with

$$K(s, t) = \begin{cases} s(1 - t), & s \leq t \\ t(1 - s), & t \leq s \end{cases} \quad (24)$$

is  $x(s) = \sin(\pi s)$ . It is seen in Table 2 that `errest` is a close bound on the actual error at the nodes and the accuracy of the solution at the nodes is preserved throughout the interval by the Nyström interpolant.

Table 2: `TestDiscontinuous` with default tolerances.

Problem	Final n	cond	Interpolation Error	Error at Nodes	
				<code>errest</code>	true
#4	32	3	2.2e-16	3.3e-16	6.7e-16
#5	32	1.4	1.7e-07	1.8e-07	1.7e-07

## 5 Logarithmic Singularity

In this section we consider integral equations of the form

$$\lambda x(s) - \int_a^b L(s,t) \log |s-t| x(t) dt = f(s), \quad a \leq s \leq b \quad (25)$$

The function  $L(s,t)$  is assumed to be moderately smooth on all of  $[a,b] \times [a,b]$ . The usual Simpson's rule will not provide a good approximation for integrands of this kind, so we use instead a product Simpson's rule [1], [4, §4.2] that incorporates the singularity. These integral equations can also be difficult numerically because the solution  $x(s)$  can behave badly near the end points of the interval. We use a graded mesh to deal with this possibility. We begin by developing a numerical procedure for  $L(s,t) \equiv 1$  and then extend it to general  $L(s,t)$ .

In the discretization we consider meshes  $\{t_0, \dots, t_n\}$  that are graded near both  $a$  and  $b$ . We discuss various possibilities for the grading, but in all cases  $n$  is divisible by 4,  $b - t_{n-j} = t_j - a$ , and

$$t_j = \frac{1}{2}(t_{j-1} + t_{j+1}), \quad j \text{ odd}, \quad j = 1, 3, \dots, n-1 \quad (26)$$

We approximate the integral  $\mathcal{K}x(s)$  using a product rule based on quadratic interpolation. As in the last section, let  $\tilde{x}_j(t)$  be the quadratic interpolating  $x(t)$  at the nodes  $\{t_{j-1}, t_j, t_{j+1}\}$ . We then have

$$\begin{aligned} \mathcal{K}x(s) &\equiv \int_a^b \log |s-t| x(t) dt = \sum_{\substack{j=1 \\ j \text{ odd}}}^{n-1} \int_{t_{j-1}}^{t_{j+1}} \log |s-t| x(t) dt \\ &\approx \sum_{\substack{j=1 \\ j \text{ odd}}}^{n-1} \int_{t_{j-1}}^{t_{j+1}} \log |s-t| \tilde{x}_j(t) dt \\ &= \sum_{k=0}^n w_k(s) x(t_k) \equiv \mathcal{K}_n x(s) \end{aligned} \quad (27)$$

As in the preceding section this approximation leads to

$$\lambda x_n(s) - \sum_{k=0}^n w_k(s) x_n(t_k) = f(s), \quad a \leq s \leq b \quad (28)$$

In familiar fashion we solve this by first solving the linear system

$$\lambda x_n(t_i) - \sum_{k=0}^n w_k(t_i) x_n(t_k) = f(t_i), \quad i = 0, 1, \dots, n \quad (29)$$

and then solving (28) for  $x_n(s)$  to get the Nyström interpolant. Below we discuss evaluation of the integrals in  $w_k(s)$ .

An error analysis for this scheme was first given in [1] and a more up-to-date discussion is given in [4, §4.2]. As with the methods discussed earlier, it is known that for all sufficiently large  $n$ , equation (28) has a solution and there is a constant  $c > 0$  for which (7) holds. However, in contrast to the methods discussed earlier, it is necessary here to use an appropriately graded mesh to get  $\|x - x_n\|_\infty = O(n^{-p})$  with  $p = 3$  or larger [4, Thm. 4.2.3]. Accordingly, for a grading parameter  $q \geq 1$ , we define the nodes of even index by

$$\begin{aligned} t_j &= a + \frac{b-a}{2} \left(\frac{2j}{n}\right)^q, \\ t_{n-j} &= b + a - t_j, \quad j = 0, 2, 4, \dots, n/2 \end{aligned} \quad (30)$$

and then define the nodes of odd index by (26). It is known that if  $q > 3$ , then  $\|\mathcal{K}x - \mathcal{K}_n x\|_\infty = O(n^{-3})$  and we get convergence at the same order. Often the convergence is even faster, so in **Fig**, we have chosen  $q = 3$  as being sufficient for our purposes.

Computing the weights of (29) involves three kinds of integrals. In stating the integrals for an odd integer  $j$ , we use the fact that  $h_j = t_{j+1} - t_j = t_j - t_{j-1}$ .

$$\begin{aligned} I_{1,j} &= \int_{t_{j-1}}^{t_{j+1}} \frac{(t-t_j)(t-t_{j+1})}{2h_j^2} \log|t-s| dt \\ I_{2,j} &= \int_{t_{j-1}}^{t_{j+1}} \frac{(t-t_{j-1})(t-t_{j+1})}{-h_j^2} \log|t-s| dt \\ I_{3,j} &= \int_{t_{j-1}}^{t_{j+1}} \frac{(t-t_{j-1})(t-t_j)}{2h_j^2} \log|t-s| dt \end{aligned}$$

In terms of the auxiliary integrals

$$I_k(\beta) = \int_{-1}^1 u^k \log|u-\beta| du, \quad k = 0, 1, 2 \quad (31)$$

and

$$\beta = \frac{s-t_j}{h_j} \quad (32)$$

the desired integrals are

$$\begin{aligned} I_{1,j} &= \frac{1}{3} h_j \log h_j + \frac{h_j}{2} [I_2(\beta) - I_1(\beta)] \\ I_{2,j} &= \frac{4}{3} h_j \log h_j + h_j [I_0(\beta) - I_2(\beta)] \\ I_{3,j} &= \frac{1}{3} h_j \log h_j + \frac{h_j}{2} [I_2(\beta) + I_1(\beta)] \end{aligned} \quad (33)$$

The auxiliary integrals (31) can be evaluated analytically as

$$I_0(\beta) = (1 - \beta) \log |1 - \beta| + (1 + \beta) \log |1 + \beta| - 2 \quad (34)$$

$$I_1(\beta) = -\beta + \frac{1}{2} (1 - \beta^2) \log \left| \frac{1 - \beta}{1 + \beta} \right| \quad (35)$$

$$I_2(\beta) = -\frac{2}{9} (1 + 3\beta^2) + \frac{1}{3} (1 - \beta^3) \log |1 - \beta| + \frac{1}{3} (1 + \beta^3) \log |1 + \beta| \quad (36)$$

These formulas are satisfactory for  $|\beta|$  of moderate size, but for graded meshes,  $|\beta|$  can be quite large and the formulas then suffer a loss of significance. To deal with large  $|\beta|$ , we rewrite (31) as

$$I_k(\beta) = \int_{-1}^1 u^k \left\{ \log |\beta| + \log \left| 1 - \frac{u}{\beta} \right| \right\} du \quad (37)$$

and expand  $\log |1 - u/\beta|$  in a Taylor series. This leads to the following approximations used by **File** for  $|\beta| > 10$ . Each is in error by less than  $10^{-16}$ .

$$I_0(\beta) \approx 2 \log |\beta| - 2 \left\{ \frac{1}{(2 \times 3) \beta^2} + \frac{1}{(4 \times 5) \beta^4} + \frac{1}{(6 \times 7) \beta^6} + \frac{1}{(8 \times 9) \beta^8} + \frac{1}{(10 \times 11) \beta^{10}} + \frac{1}{(12 \times 13) \beta^{12}} \right\} \quad (38)$$

$$I_1(\beta) \approx -2 \left\{ \frac{1}{(1 \times 3) \beta} + \frac{1}{(3 \times 5) \beta^3} + \frac{1}{(5 \times 7) \beta^5} + \frac{1}{(7 \times 9) \beta^7} + \frac{1}{(9 \times 11) \beta^9} + \frac{1}{(11 \times 13) \beta^{11}} \right\} \quad (39)$$

$$I_2(\beta) \approx \frac{2}{3} \log |\beta| - 2 \left\{ \frac{1}{(2 \times 5) \beta^2} + \frac{1}{(4 \times 7) \beta^4} + \frac{1}{(6 \times 9) \beta^6} + \frac{1}{(8 \times 11) \beta^8} + \frac{1}{(10 \times 13) \beta^{10}} + \frac{1}{(12 \times 15) \beta^{12}} \right\} \quad (40)$$

We now consider what changes are needed to deal with the general equation (25). The numerical integration operator  $\mathcal{K}_n x$  of (27) is extended in the obvious way of replacing the function  $x$  with  $L(s, \cdot)x$ , resulting in the approximation

$$\mathcal{K}_n x(s) = \sum_{k=0}^n w_k(s) L(s, t_k) x(t_k), \quad a \leq s \leq b \quad (41)$$

The weights  $\{w_k(s)\}$  and the process of computing  $x_n(s)$  are unchanged. It is not difficult to extend the theory outlined above for the special case to this more general situation.

The **TestLogSing** program has five test problems that have kernels of the form  $L(s, t) \log |s - t|$  with a function  $L(s, t)$  that is moderately smooth on all of  $[a, b] \times [a, b]$ . The interval can be specified and so can  $\lambda$ . Two of the test

problems will illustrate the possibilities and the performance of the solver for problems with logarithmic singularities.

**Problem 4** The function  $f(s)$  is defined so that the integral equation

$$\lambda x(s) - \int_0^b t \log |s - t| x(t) dt = f(s), \quad a \leq s \leq b \quad (42)$$

has the solution  $x(s) = s^2$ . For the numerical results reported here,  $b = 2$  and  $\lambda = 1$ . It is seen in Table 3 that **errest** is a close bound on the actual error at the nodes and the accuracy of the solution at the nodes is preserved throughout the interval by the Nyström interpolant.

**Problem 5** The function  $f(s)$  is defined so that the integral equation

$$\lambda x(s) - \int_0^b \log |s - t| x(t) dt = f(s), \quad a \leq s \leq b \quad (43)$$

has the solution  $x(s) = s \log(s)$ . For the numerical results reported here,  $b = 2$  and  $\lambda = 1$ . It is seen in Table 3 that **errest** is a close bound on the actual error at the nodes and the accuracy of the solution at the nodes is preserved throughout the interval by the Nyström interpolant.

Table 3: TestLogSing with default tolerances.

Problem	Final n	cond	Interpolation Error	Error at Nodes	
				errest	true
#4	32	12	3.7e-4	8.3e-4	1.6e-4
#5	32	9	8.3e-5	1.2e-4	3.8e-5

## 6 Algebraic Singularity

In this section we consider integral equations of the form

$$\lambda x(s) - \int_a^b \frac{L(s, t)}{|s - t|^\alpha} x(t) dt = f(s), \quad a \leq s \leq b \quad (44)$$

The function  $L(s, t)$  is assumed to be moderately smooth on all of  $[a, b] \times [a, b]$  and  $0 < \alpha < 1$ . The way that we deal with algebraic singularities is so much like the treatment of logarithmic singularities in §5 that we need only take up the differences.

We use a graded mesh that has the same form as the one used for logarithmic singularities, but  $q = 1/(1.5 - \alpha)$  instead of  $q = 3$ . This provides a mesh optimal for the  $L_2$  norm.

The equivalent of the auxiliary integrals (31) is

$$I_k(\beta) = \int_{-1}^1 \frac{u^k}{|u - \beta|^\alpha} du, \quad k = 0, 1, 2 \quad (45)$$

and the analogs of the formulas (33) are

$$\begin{aligned} I_{1,j} &= \frac{h_j^{1-\alpha}}{2} [I_2(\beta) - I_1(\beta)] \\ I_{2,j} &= h_j^{1-\alpha} [I_0(\beta) - I_2(\beta)] \\ I_{3,j} &= \frac{h_j^{1-\alpha}}{2} [I_2(\beta) + I_1(\beta)] \end{aligned} \quad (46)$$

Analytical evaluation of the auxiliary integrals (45) is somewhat more complicated than (35) because we must consider three cases:

$$\begin{aligned} I_0(\beta) &= \frac{1}{1-\alpha} \left[ (1-\beta)^{1-\alpha} - (-1-\beta)^{1-\alpha} \right], \quad \beta \leq -1 \\ &= \frac{1}{1-\alpha} \left[ (\beta+1)^{1-\alpha} - (\beta-1)^{1-\alpha} \right], \quad \beta \geq 1 \\ &= \frac{1}{1-\alpha} \left[ (1-\beta)^{1-\alpha} + (1+\beta)^{1-\alpha} \right], \quad -1 \leq \beta \leq 1 \end{aligned} \quad (47)$$

$$\begin{aligned} I_1(\beta) &= \frac{1}{2-\alpha} \left[ (1-\beta)^{2-\alpha} - (-1-\beta)^{2-\alpha} \right] + \beta I_0(\beta), \quad \beta \leq -1 \\ &= \frac{1}{2-\alpha} \left[ (\beta-1)^{2-\alpha} - (\beta+1)^{2-\alpha} \right] + \beta I_0(\beta), \quad \beta \geq 1 \\ &= \frac{1}{2-\alpha} \left[ (1-\beta)^{2-\alpha} - (1+\beta)^{2-\alpha} \right] + \beta I_0(\beta), \quad -1 \leq \beta \leq 1 \end{aligned} \quad (48)$$

$$\begin{aligned} I_2(\beta) &= \frac{1}{3-\alpha} \left[ (1-\beta)^{3-\alpha} - (-1-\beta)^{3-\alpha} \right] \\ &\quad + 2\beta I_1(\beta) - \beta^2 I_0(\beta), \quad \beta \leq -1 \\ &= \frac{1}{3-\alpha} \left[ (\beta+1)^{3-\alpha} - (\beta-1)^{3-\alpha} \right] \\ &\quad + 2\beta I_1(\beta) - \beta^2 I_0(\beta), \quad \beta \geq 1 \\ &= \frac{1}{3-\alpha} \left[ (1-\beta)^{3-\alpha} + (1+\beta)^{3-\alpha} \right] \\ &\quad + 2\beta I_1(\beta) - \beta^2 I_0(\beta), \quad -1 \leq \beta \leq 1 \end{aligned} \quad (49)$$

These formulas are satisfactory for  $|\beta|$  of moderate size, but for  $|\beta| > 10$ , we avoid loss of significance by using the truncated Taylor series

$$I_0(\beta) \approx |\beta|^{-\alpha} \sum_{\ell=0}^6 \binom{-\alpha}{2\ell} \frac{2}{2\ell+1} \beta^{-2\ell}$$

$$I_1(\beta) \approx -|\beta|^{-\alpha-1} \sum_{\ell=0}^6 \binom{-\alpha}{2\ell+1} \frac{2}{2\ell+3} \beta^{-2\ell}$$

$$I_2(\beta) \approx |\beta|^{-\alpha} \sum_{\ell=0}^6 \binom{-\alpha}{2\ell} \frac{2}{2\ell+3} \beta^{-2\ell}$$

which have relative errors comparable to the unit roundoff.

The `TestAlgSing` program has four test problems that have kernels of the form  $L(s, t)/|s - t|^\alpha$  with a function  $L(s, t)$  that is moderately smooth on all of  $[a, b] \times [a, b]$  and  $0 < \alpha < 1$ . All the test problems allow  $\lambda$  to be specified and the first two allow  $\alpha$  to be specified as well. Two of the test problems will illustrate the possibilities and the performance of the solver for problems with algebraic singularities.

**Problem 3** The function  $f(s)$  is defined so that the integral equation

$$\lambda x(s) - \int_0^1 \frac{x(t)}{\sqrt{|s-t|}} dt = f(s), \quad 0 \leq s \leq 1 \quad (50)$$

has the solution  $x(s) = s \log(s)$ . This equation with a general  $f(s)$  is sometimes called the Kirkwood-Riseman equation. It arises in the study of polymer hydrodynamics [10]. For the numerical results reported here,  $\lambda = 5$ . It is seen in Table 4 that `errest` is a close bound on the actual error at the nodes and the accuracy of the solution at the nodes is preserved throughout the interval by the Nyström interpolant.

**Problem 4** The function  $f(s)$  is defined so that the integral equation

$$\lambda x(s) - \int_0^{\pi/2} \frac{x(t)}{\sqrt{|\sin(s-t)|}} dt = f(s), \quad 0 \leq s \leq \pi/2 \quad (51)$$

has the solution  $x(s) = s \log(s)$ . To apply `Fie` we write the kernel as

$$K(s, t) = \frac{1}{\sqrt{|\sin(s-t)|}} = L(s, t) \frac{1}{\sqrt{|s-t|}}$$

defining  $L(s, t)$  implicitly as a function continuous across  $s = t$ . For the numerical results reported here,  $\lambda = 5$ . It is seen in Table 4 that `errest` is a close bound on the actual error at the nodes and the accuracy of the solution at the nodes is preserved throughout the interval by the Nyström interpolant.

## 7 Infinite Interval

In this section we consider integral equations of the form

$$\lambda x(s) - \int_0^\infty K(s, t) x(t) dt = f(s), \quad 0 \leq s < \infty \quad (52)$$



Table 4: `TestAlgSing` with default tolerances.

Problem	Final n	cond	Interpolation Error	Error at Nodes	
				errest	true
#3	32	9	2.4e-5	3.1e-5	1.1e-5
#4	32	19	5.1e-5	6.3e-5	2.6e-5

The approach we take to the numerical solution of an integral equation of this kind is to change variables to obtain an integral equation on  $[0, 1]$  and then apply a variant of the method developed for smooth kernels in §3. This approach might succeed in a variety of circumstances, but we discuss only one set of assumptions. We assume that  $f(s)$  is a bounded, continuous function, which is to say that  $f \in BC[0, \infty)$ . The solution  $x(s)$  is to be in the space of uniformly continuous functions on  $[0, \infty)$ ,

$$UC[0, \infty) = \left\{ x \in BC[0, \infty) \mid x(\infty) \equiv \lim_{t \rightarrow \infty} x(t) \text{ exists} \right\}$$

We use

$$\|x\|_\infty = \sup_{t \geq 0} |x(t)|$$

as a norm and assume that the integral operator

$$\mathcal{K}x(s) = \int_0^\infty K(s, t) x(t) dt \quad 0 \leq s < \infty \quad (53)$$

is compact on  $UC[0, \infty)$ . Changing variables to the interval  $[0, 1]$  results in a kernel function  $\widehat{K}(\sigma, \tau)$ . If it extends to a function that is continuous for  $0 \leq \sigma, \tau \leq 1$ , then  $\mathcal{K}$  is compact on  $UC[0, \infty)$ .

We use the changes of variable

$$t = \frac{1 - \tau}{\tau}, \quad s = \frac{1 - \sigma}{\sigma}, \quad 0 < \sigma, \tau \leq 1$$

which are inverted by

$$\tau = \frac{1}{1 + t}, \quad \sigma = \frac{1}{1 + s}, \quad 0 \leq s, t < \infty$$

If we introduce the functions

$$F(\sigma) = f(s), \quad X(\tau) = x(t)$$

$$\widehat{K}(\sigma, \tau) = \frac{1}{\tau^2} K(s, t) = \frac{1}{\tau^2} K\left(\frac{1 - \sigma}{\sigma}, \frac{1 - \tau}{\tau}\right), \quad 0 < \sigma, \tau \leq 1 \quad (54)$$

we find that equation (52) is equivalent to

$$\lambda X(\sigma) - \int_0^1 \widehat{K}(\sigma, \tau) X(\tau) d\tau = F(\sigma), \quad 0 < \sigma \leq 1 \quad (55)$$

With this change of variables,  $x \in UC[0, \infty)$  is equivalent to  $X \in C[0, 1]$ . If  $K(s, t) \rightarrow 0$  fast enough as  $s, t \rightarrow \infty$ , then  $\widehat{K}(\sigma, \tau)$  will be continuous for  $0 \leq \sigma, \tau \leq 1$ . To gain insight, suppose that for some  $p, q > 0$ ,

$$K(s, t) = O(s^{-p} + t^{-q}) \quad \text{as } s, t \rightarrow \infty \quad (56)$$

It then follows that

$$\widehat{K}(\sigma, \tau) = O(\sigma^p + \tau^{q-2}) \quad \text{as } \sigma, \tau \rightarrow 0$$

If  $q > 2$ , we have the continuity of  $\widehat{K}(\sigma, \tau)$  at  $(0, 0)$  that we want. We might also have the continuity with  $q = 2$ , but this will depend on details of the kernel. In general, the equation (55) on  $[0, 1]$  with kernel function (54) can be examined to determine if the corresponding equation on  $[0, \infty)$  is solvable.

As we noted at the beginning of this section, **Fie** may be able to solve equations on  $[0, \infty)$  with other assumptions about the problem. In particular, it may be able to solve problems with solutions that do not have a limit at infinity. There are reasonable prospects for solving such a problem if the solution  $x$  and the kernel  $K$  satisfy

$$K(s, t)x(t) = O(s^{-p} + t^{-q}) \quad \text{as } s, t \rightarrow \infty$$

with  $p \geq 4$  and  $q \geq 2$ , though it might succeed with just  $p > 2$  and  $q > 0$ . Though the solution may not have a limit at infinity, its behavior is known in these circumstances:

$$x(s) \approx \frac{1}{\lambda} f(s) \quad \text{as } s \rightarrow \infty$$

The numerical solution of (55) differs from the method of §3 in only a few details. The most important is that we use a different quadrature rule, specifically the two-point Gaussian quadrature rule. This has the same order as Simpson's rule, but does not evaluate the integrand at the ends of the interval. In this way we avoid evaluating the kernel function at zero, hence avoid evaluating the original kernel function at infinity. The coding is simplified by using the same quadrature rule for all subintervals. To increase the reliability of the convergence test, we use *ratio* = 0.5 in (11), hence estimate the error by

$$\|x - x_n\|_\infty \approx \|x_n - x_{n/2}\|_\infty$$

One consequence of this decision is that if all goes well, **errest** will be rather larger than the true error at the nodes.

The **TestInf** program has fourteen problems. Of course the interval is fixed, but  $\lambda$  can be specified. This test program has default tolerances **AbsTol** = **1e-4** and **RelTol** = **1e-2**. Although we think it better to ask for less accuracy when solving problems on an infinite interval, we have preferred that the default tolerances in **Fie** itself be uniform across the problem classes. Each of the test programs displays the computed solution, but when the interval is  $[0, \infty)$ ,

this may be unsatisfactory because it gives too much attention to the way  $x(s)$  approaches its limit at infinity. A more satisfactory display is obtained by means of an optional argument `section` that has default value 10. `TestInf` plots the computed solution on the interval  $[0, \text{section}]$ . It is to be appreciated that `section` has nothing to do with the computation of the solution, just the display. All the test programs for finite intervals assess the accuracy of the Nyström interpolant by measuring the error at 150 points equally spaced in  $[a, b]$ . Assuming that `section` identifies a region of interest when the interval is  $[0, \infty)$ , `TestInf` measures the error of the interpolant at 150 equally spaced points in  $[0, \text{section}]$ . Two of the test problems will illustrate the possibilities and the performance of the solver for problems on  $[0, \infty)$ .

**Problem 1** The function  $f(s)$  is defined so that the integral equation

$$\lambda x(s) - \int_0^\infty \frac{x(t)}{1 + s^2 + t^2} dt = f(s), \quad 0 \leq s < \infty \quad (57)$$

has the solution  $x(s) = 1/(1 + s^2)$ . I. Sloan [15] uses this problem with  $\lambda = 1$  to compare two ways of solving problems on the interval  $[0, \infty)$ , one of which is essentially the same as the method of `File`. For the numerical results reported here,  $\lambda = 1$  and the default tolerances of `TestInf` were used. This kernel function converges to zero at a relatively slow rate as the arguments tend to infinity, but the results seen in Table 5 are quite satisfactory. Note, however, that the estimated bound on the error is much more conservative than for the example problems on a finite interval.

**Problem 10** The function  $f(s)$  is defined so that the integral equation

$$\lambda x(s) - \int_0^\infty \frac{x(t)}{(1 + s^2 + t^2)^2} dt = f(s), \quad 0 \leq s < \infty \quad (58)$$

has the solution  $x(s) = \cos(s)/(1 + s^2)$ . For the numerical results reported here,  $\lambda = 2$  and the default tolerances of `TestInf` were used. Like the preceding example, the results seen in Table 5 are quite satisfactory and `errest` is rather conservative.

Table 5: `TestInf` with `AbsTol = 1e-4`, `RelTol = 1e-2`.

Problem	Final n	cond	Interpolation Error	Error at Nodes	
				<code>errest</code>	true
#1	32	23	5.3e-7	9.1e-6	5.3e-7
#10	32	2	5.9e-7	1.6e-5	5.9e-7

## 8 Cautionary Notes

The examples presented here are representative of a substantial set of test problems that show `File` to be an effective program for the solution of a large class of

Fredholm integral equations of the second kind. Working in a problem solving environment, it is appropriate that **Fie** solve integral equations to only modest accuracy. Correspondingly, all the methods implemented are of relatively low order and a moderate limit is placed on the size of the dense linear systems that arise. As a consequence, it is easy to make the solver fail by asking for too much accuracy.

**Fie** approximates an integral equation (1) by a system of linear equations  $Az = c$  where the components of  $c$  are values of  $f(s)$ . With modest assumptions we have convergence as the size of the system, and in particular the number of samples of  $f$ , tends to infinity, but we cannot expect an acceptable approximation to  $x(s)$  when a few hundred samples are not representative. This is an issue in solving any integral equation, but it is of special concern when solving a problem set on  $[0, \infty)$

**Acknowledgement.** We appreciate very much the exceptionally helpful reports provided by all the referees. Their comments have led to improvements in the program and the paper.

## References

- [1] Atkinson, K. 1967. The numerical solution of Fredholm integral equations of the second kind, *SIAM J. Num. Anal.* **4**, 337–348.
- [2] Atkinson, K. 1976. *A Survey of Numerical Methods for the Solution of Fredholm Integral Equations of the Second Kind*, SIAM, Philadelphia.
- [3] Atkinson, K. 1976. An automatic program for Fredholm linear integral equations of the second kind, *ACM Trans. Math. Softw.* **2**, 154–171.
- [4] Atkinson, K. 1997. *The Numerical Solution of Integral Equations of the Second Kind*, Cambridge University Press, Cambridge.
- [5] Atkinson, K. and Han, W. 2005. *Theoretical Numerical Analysis: A Functional Analysis Framework*, 2<sup>nd</sup> ed., Springer-Verlag, New York.
- [6] Cubillos, P. 1984. Integral operators with Green’s function type kernel, *J. of Comp. & Appl. Math.* **10**, 25–31.
- [7] El-Gendi, S.E. 1969. Chebyshev solution of differential, integral and integro-differential equations, *Comput. J.* **12**, 282–287.
- [8] Fox, L. 1962. *Numerical Solution of Ordinary and Partial Differential Equations*, Pergamon Press, London.
- [9] *GAMS. The Guide to Available Mathematical Software* is available at <http://gams.nist.gov/>.

- [10] Kirkwood, J. and Riseman, J. 1948. The intrinsic viscosities and diffusion constants of flexible macromolecules in solution, *J. Chem. Phys.* **16**, 565–573.
- [11] Love, E. 1949. The electrostatic field of two equal circular conducting disks, *Quart. J. Mech. Appl. Math.* **2**, 428–451.
- [12] MATLAB, The MathWorks, Inc., 3 Apple Hill Dr., Natick, MA 01760.
- [13] Mikhlin, S. G. and Smolitskiy, K. L. 1967. *Approximate Methods for Solution of Differential and Integral Equations*, Elsevier, London.
- [14] NAG FORTRAN Library, Numerical Algorithms Group Inc., Oxford, U.K.
- [15] Sloan, I. 1981. Quadrature methods for integral equations of the second kind over infinite intervals, *Math. Comp.* **36**, 511–523.