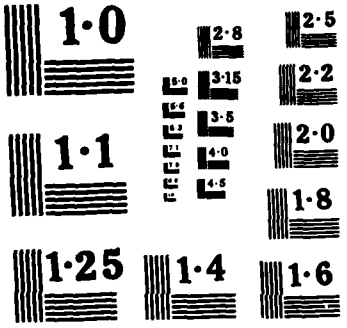
AD-A158 189 COMPUTABLE BOUNDS FOR SOLUTIONS OF INTEGRAL EQUATIONS (U) WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER L B RALL MAY 85 MRC-TSR-2816 DAAG29-80-C-0041 1/1

UNCLASSIFIED F/G 12/1 NL



(2)

MRC Technical Summary Report #2816

# COMPUTABLE BOUNDS FOR SOLUTIONS OF INTEGRAL EQUATIONS

L. B. Rall

AD-A158 189

**Mathematics Research Center**
**University of Wisconsin—Madison**
**610 Walnut Street**
**Madison, Wisconsin 53705**



May 1985

(Received May 2, 1985)



**Approved for public release**
**Distribution unlimited**

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

UNIVERSITY OF WISCONSIN-MADISON
MATHEMATICS RESEARCH CENTER

# COMPUTABLE BOUNDS FOR SOLUTIONS OF INTEGRAL EQUATIONS

L. B. Rall

Technical Summary Report #2816
May 1985

## ABSTRACT

Interval integration is used to obtain inclusions of integral operators of the form

$$g(u)(s) = \int g(s,t,u(s),u(t))dt \qquad (1)$$

which can be carried out on a computer. The resulting inclusions, combined with interval iteration, are used to compute guaranteed upper and lower bounds for solutions of integral equations of the form

$$u = g(u) \qquad (2)$$

for $s \in S$. It is also possible to establish existence or nonexistence of solutions of integral equations in given regions on the basis of results of the computation. Examples of applications of this technique to linear and nonlinear integral equations are eigenvalue problems for linear integral operators are given.

AMS (MOS) Subject Classifications: 65G10, 65J10, 65J20, 65R20

Key words: Integral equations, Eigenvalue problems, Error bounds, Interval integration, Interval iteration

Work Unit Number 3 - Numerical Analysis and Scientific Computing

Sponsored by the United States Army Under Contract No. DAAG29-80-C-0041.

## SIGNIFICANCE AND EXPLANATION

This report is the text of an invited address to be given at the 11th IMACS World Congress, to be held in Oslo, Norway, on August 5-9, 1985. The theories of interval integration, interval iteration and their applications to integral equations have been developed in previous papers. This report surveys these results, but concentrates on computer implementation by discussing various ways in which inclusions of integral transforms can actually be represented on a computer. The computations based on these inclusions are self-validating, in that existence of solutions and upper and lower bounds for the solutions of linear and nonlinear integral equations and eigenvalue problems can be guaranteed on the basis of computed results alone. It is assumed, of course, that the computing environment supports interval computation. Since support for interval arithmetic is now required by the IEEE standard for floating-point arithmetic, and provided by Pascal-SC for microcomputers and ACRITH for IBM 370 mainframe computers, the need to make special provision for interval arithmetic is only necessary on outdated systems.

Some simple examples illustrating the techniques presented in the paper are cited.



The responsibility for the wording and views expressed in this descriptive summary lies with MRC, and not with the author of this report.

# COMPUTABLE BOUNDS FOR SOLUTIONS OF INTEGRAL EQUATIONS

L. B. Rall


## Summary

Interval integration is used to obtain inclusions of integral operators of the form

$$g(u)(s) = \int_T g(s,t,u(s),u(t))dt \qquad (1)$$

which can be carried out on a computer. The resulting inclusions, combined with interval iteration, are used to compute guaranteed upper and lower bounds for solutions of integral equations of the form

$$u = g(u) \qquad (2)$$

for $s \in S$. It is also possible to establish existence or nonexistence of solutions of integral equations in given regions on the basis of results of the computation. Examples of applications of this technique to linear and nonlinear integral equations and eigenvalue problems for linear operators are given.


## Integral Equations

The integral equation expressed by (2) includes many of the varieties of integral equations of importance in applications. For example, for

---

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.

$$g(s,t,u(s),u(t)) = f(x) + \lambda K(s,t)u(t), \tag{3}$$

the result is the linear integral equation of Fredholm type and second kind,

$$u(s) = f(x) + \lambda \int_T K(s,t)u(t)dt, \tag{4}$$

while the choice

$$g(s,t,u(s),u(t)) = f(s,t,u(t)), \tag{5}$$

gives the nonlinear integral equation of Urysohn type

$$u(s) = \int_T f(s,t,u(t))dt, \tag{6}$$

and so on [12]. In general, S and T are subsets of some possibly high-dimensional real or complex spaces. For definiteness, we will take S = T = [0,1], and consider integral equations of the form

$$u(s) = \int_0^1 g(s,t,u(s),u(t))dt, \tag{7}$$

since the discussion given here will carry over immediately to the more general case. It is also worth noting that (7) includes Volterra integral equations as the special case $g(s,t,v,w) \equiv 0$ for $t > s$.

## Interval Analysis

Our purpose is to find functions $\underline{u}(t)$, $\bar{u}(t)$ such that

$$\underline{u}(s) \leq u(s) \leq \bar{u}(s) \tag{8}$$

$0 \leq s \leq 1$, in other words, an interval-valued function

$$U(s) = [\underline{u}(s), \bar{u}(s)] \tag{9}$$

such that $u(s) \in U(s)$ for $s \in [0,1]$ [1]. The process described in [12] for this purpose is a combination of interval integration [1] and interval iteration [9].

Interval Integration

The integral of an interval function U(t) is defined to be

$$\int_0^1 U(s)ds = [\underline{\int_0^1} \underline{u}(s)ds, \overline{\int_0^1} \bar{u}(s)ds], \tag{10}$$

where under and overbars on the integral signs within the brackets denote respectively lower and upper Darboux integrals [1]. Since these Darboux integrals always exist, it follows that all interval (and hence all real) functions are integrable in this sense.

Interval Iteration

Interval iteration for the fixed-point problem (2) works like this [9], [12]: Suppose that one has an interval inclusion G of the operator g, that is, an operator from interval functions to interval functions such that $g(u) \in G(U)$ for each $u \in U$. Then, starting with an intial interval function $U_0$, one computes

$$U_{n+1} = U_n \cap G(U_n). \tag{11}$$

Interval iteration has the following properties [9]:

1. If $u \in U_0$ is a solution of (2), then the interval iteration will converge to

$$U^* = \bigcap_{n=0}^{\infty} U_n, \tag{12}$$

and $u \in U^*$.

2. If for some integer n,

$$U_n \cap G(U_n) = \emptyset, \tag{13}$$

the empty set, then there are no solutions u of (2) in the initial interval $U_0$.

3. If

$$G(U_n) \subset U_n \tag{14}$$

for some integer n, then a solution u of (2) exists in the initial interval $U_0$.

Thus, interval iteration can be used to prove existence of solutions if (14) holds, nonexistence by (13), or to obtain improved bounds for solutions by (12). The problem in actual computation is then to construct an interval inclusion G of the integral operator g in (1). One way to do this is to construct an inclusion G(s,t,U(s),U(t)) of the integrand by using interval arithmetic [6], and then use interval integration to include the operator. Here, by "interval arithmetic", we mean interval inclusions of the ordinary library functions available on a computer in addition to the arithmetic operations themselves. We now present several ways to implement such a construction on an actual computer. It is assumed, of course, that interval computation is properly supported, as in Pascal-SC or ACRITH.

## Inclusion of Integral Operators

### Interval Arithmetic

A straightforward method to construct an inclusion of the integral operator (1) simply uses interval arithmetic. Here, the interval S = T = [0,1] is partitioned into n subintervals

$$S_i = T_i = [s_{i-1}, s_i]$$

with endpoints $s_{i-1} < s_i$, $s_0 = 0$, $s_n = 1$, and widths $w(S_i) = s_i - s_{i-1}$. Further, we define an interval step function U by $U(s) = U_i = [\underline{u}_i, \bar{u}_i]$ for $s \in S_i$. Then, if $u \in U$, it follows that

$$g(s,t,u(s),u(t)) \in G(S_i,T_j,U_i,U_j) \tag{15}$$

for $s \in S_i$, $t \in T_j$. Thus,

$$u(s) \in \sum_{j=1}^{n} G_{ij}(U_i,U_j) \cdot w(T_j), \tag{16}$$

where $G_{ij}(U_i,U_j) = G(S_i,T_j,U_i,U_j)$, because the interval Riemann sum on the right side of (16) contains the interval integral of G(s,t,U(s),U(t)) [1]. In this case, the interval iteration (11) can be carried out in the form

$$U_i^{n+1} = \Big[ \sum_{j=1}^{i-1} G_{ij}(U_i^n, U_j^{n+1}) \cdot w(T_j) + \sum_{j=i}^{n} G_{ij}(U_i^n, U_j^n) \cdot w(T_j) \Big] \cap U_i^n, \tag{17}$$

a "Gauss-Seidel" iteration which makes use of updated information as it is produced, and has the same convergence properties as (11) [13]. This simple approach to solving integral equations has the following advantages:

1. It is easy to implement on a computer.
2. No special properties of g(s,t,v,w) such as positivity or monotonicity are

required.

3. The computation is self-validating, with existence of a solution of the integral equation and lower and upper bounds for it guaranteed if (14) holds.

4. A form of "superconvergence" holds at the nodes $s_i$ for $i = 1,\ldots,n-1$. Since $u(s_i) \in U_i^n$ and $u(s_i) \in U_{i+1}^n$, it follows that

$$u(s_i) \in U_i^n \cap U_{i+1}^n, \tag{18}$$

which results in narrower bounds for $u(s)$ at these points [12].

The main disadvantage of this simple procedure is very slow convergence if iteration is carried out to the point at which $U^n = U^{n+1}$ on the computer [7]. Instead of wasting iterations which result in little improvement in the bounds for the solution, it is better to stop when the decrease in $w(U_i^n \cap U_{i+1}^n)$ falls below a given limit.

Monotonicity Methods

Interval analysis of integral equations is simplified considerably if the function $g(s,t,v,w)$ has known monotonicity properties with respect to $v,w$. The underlying partial ordering for real functions is, of course, componentwise. We say that $v \leq w$ if $v(s) \leq w(s)$ for all $s \in S$. If, for example, $g(s,t,v,w)$ is isotone (monotone increasing) with respect to $v$, and antitone (monotone decreasing) with respect to $w$, then

$$G(s,t,U(s),U(t)) =$$

$$= [g(s,t,\bar{u}(s),\underline{u}(t)),\ g(s,t,\bar{u}(s),\underline{u}(t))] \tag{19}$$

will provide an interval inclusion of $g(s,t,u(s),u(t))$ for $u \in U$. Since interval iteration is inclusion monotone [1], integration of (19) will yield an interval inclusion G of the integral operator g. Methods using monotonicity properties of operators to obtain lower and upper bounds for solutions of functional equations have been investigated extensively

by Collatz [2] and others. In theory, a function of bounded variation can be expressed as the sum of an isotone and an antitone function, so that g(s,t,u(s),u(t)) could be expressed in terms of a finite number of such functions; however, it is not usually possible to perform this construction explicitly [4].

The actual implementation of monotonicity methods is hampered by the fact that is difficult to carry out and represent the transform G(U) on the computer, even for simple interval functions U(s). For example, if $\underline{u}$ and $\overline{u}$ are step functions, then $G([\underline{u},\overline{u}])$ will not in general be a step function, and the representation of more and more complicated functions will be necessary as the iteration proceeds. A solution to this problem is provided by the concept of directed rounding in function space.

## Directed Rounding

For actual computation, we take a set $\Phi$ of functions which can be represented exactly on a computer. For example, $\Phi$ can consist of step functions which have nodes and values which are floating-point numbers. An upward rounding operator $\Delta$ from the set R of real functions into $\Phi$ has the following properties:

$$\text{(i)} \quad \Delta u = u \text{ if } u \in \Phi,$$

$$\text{(ii)} \quad u \leq \Delta u \text{ for } u \in R,$$

$$\text{(iii)} \quad \text{if } u \leq v, \text{ then } \Delta u \leq \Delta v.$$

These conditions imply, among other things, that there is no element v of $\Phi$ such that $u \leq v \leq \Delta v$ [3], hence, this rounding is "optimal" in a certain sense. The downward rounding operator $\nabla$ is defined in a similar way, with the inequality signs reversed. Thus, if one has the same monotonicity properties for the function g(s,t,u(s),u(t)) as in (19), an interval inclusion of g can be computed by

$$G(U) = [\nabla\int_0^1 G(s,t,\underline{u}(s),\bar{u}(t))dt,$$

$$\Delta\int_0^1 G(s,t,\bar{u}(s),\underline{u}(t))dt], \tag{20}$$

An an example of this type of computation, consider the integrand

$$g(s,t,u(s),u(t)) = 1 + \frac{\lambda u(s)u(t)}{s + t}, \tag{21}$$

which arises in radiative transfer theory, where $0 \leq \lambda \leq 0.5$ [12]. Here, $\Phi$ will be the set of step functions, and it follows from (21) that the integrand is isotone for positive functions. Let $f \in \Phi$ have value $f_i$ in the subinterval $T_i$ of $[0,1]$. Then, for $s \in S_i$,

$$g(f)(s) = 1 + \lambda s f_i \cdot \sum_{j=1}^{n} f_j \cdot \ln\{\frac{s + s_j}{s + s_{j-1}}\}, \tag{22}$$

a monotone function which can be rounded downward and upward to bound the integral operator [10]. Numerical computations using this method for bounding the integral operator show both faster convergence and applicability to an extended range of $\lambda$ as compared to the simple approximation to the integral operator using interval arithmetic [10]. For example, (14) holds when the operator is approximated by interval Riemann sums only for $0 \leq \lambda \leq$ 0.365, while the use of (22) extends this range to $0 \leq \lambda \leq 0.451$ [10]. The number of iterations is also drastically reduced over those reported in [7].

## Iterative Residual Correction

The method of interval iteration can be combined in a simple way with other approximate methods for the solution of integral equations. The result gives essentially a method for choosing the initial interval $U_0$ in which the solution u of (2) is sought. Suppose, for example, that a numerical solution of the integral equation yields the

approximation $u_0 = u_0(s)$ to a solution. If this solution is believed to be correct within $\varepsilon$, then it is natural to start the interval iteration with $U_0(s) = u_0(s) + \varepsilon \cdot [-1,1]$, for example. If it turns out that (14) is satisfied for $n = 0$, then the interval computation immediately validates the existence of the solution $u$ in the interval $U_0$, and also the error bound $\varepsilon$ [3]. The interval iteration can then be continued if it is desired to obtain a better error bound for the approximate solution than $\varepsilon$. The same applies to $U_n$ if (14) is satisfied for some value of $n > 0$.

The advantages to this approach are that it may be much quicker to obtain a good approximation to the solution of the integral equation using floating-point arithmetic instead of interval arithmetic, and the width of the initial interval $U_0$ is usually very small. This is important because interval inclusions of functions on intervals usually have excess width over the range of the function on that interval, that is, $w(F(X))$ can be much larger than $w(f(X))$ for a given real function $f$ and real interval $X$. However, it is known that $w(F(X)) \to w(f(X))$ as $w(X) \to 0$ if $f$ is continuous [5]. This convergence result implies that interval calculations conducted with intervals of small width will estimate the values of the included real function more accurately, and the iteration will hence also converge more rapidly to whatever tolerance limit is set by the solver of the equation or the computing equipment used.

In terms of the residual

$$r_0 = g(u_0) - u_0,$$

one can write

$$u - u_0 = r_0 + g(u) - g(u_0),$$

and if $G'$ denotes an interval extension of the Fréchet derivative $g'$ of $g$, then the interval mean value theorem [11] gives

$$u - u_0 \in r_0 + G'(U)(U - u_0),$$

or, for $U_0 = u_0 + \varepsilon \cdot [-1,1]$,

$$u - u_0 \in r_0 + \varepsilon \cdot G'(U_0) \cdot [-1,1].$$

## Eigenvalue Problems

A simple application of the interval method for solution of integral equations is to the eigenvalue problem

$$u(s) = \lambda \int_a^b K(s,t)u(t)dt \tag{31}$$

for Fredholm integral operators K with kernel $K(s,t)$. It follows from the above that if $U(s)$ is an interval function such that $0 \notin (U,U)$ and U contains an eigenfunction u of K, then the eigenvalue $\lambda$ belonging to the eigenfunction u will be contained in the interval Rayleigh quotient

$$\Lambda = (KU,U)/(U,U), \tag{32}$$

where the scalar products are formed by interval integration, and the division is done in interval arithmetic [8]. This process forms the basis of an interval iteration method. Starting with $U_0$, one can compute $\Lambda_0$ by (32). Having $U_n$ and $\Lambda_n$, the next step is to iterate

$$V_{n+1} = \Lambda_n K V_n \cap V_n \tag{29}$$

to convergence, starting with $V_0 = U_n$. If V* denotes the converged value of V, then one takes

$$U_{n+1} = V^* \tag{30}$$

and then

$$\Lambda_{n+1} = \Lambda_n \cap \frac{(KU_{n+1}, U_{n+1})}{(U_{n+1}, U_{n+1})} . \tag{31}$$

The interval Rayleigh quotient (32) has the advantage that it gives lower and upper bounds for all eigenvalues of K, including the smallest. Most methods for approximating the smallest eigenvalue give only upper bounds [8]. For example, let K(s,t) be the Green's function for the differential operator $-y''$ on $[0,\pi]$ with boundary conditions $y(0) = y(\pi) = 0$. One can start with $U_0(s)$ defined by

$$\bar{u}(s) = 1, \quad 0 \leq s \leq \pi,$$

$$\underline{u}(s) = (2/\pi)s, \quad 0 \leq s \leq \pi/2, \tag{32}$$

$$\underline{u}(s) = (2/\pi)(\pi - s), \quad \pi/2 \leq s \leq \pi,$$

and obtain $\Lambda_0 = [0.4052, 2.5033]$, while a second iteration gives $\Lambda_1 = [0.6792, 1.9454]$, which provides better guaranteed lower and upper bounds for the smallest eigenvalue $\lambda = 1$ of the problem.

## References


1. O. Caprani, K. Madsen, L. B. Rall. Integration of interval functions. SIAM J. Math. Anal. 12 (1981), 321-341.

2. L. Collatz. Funktionalanalysis und Numerische Mathematik. Springer, Berlin, 1964.

3. E. Kaucher and W. Miranker. Self-Validating Numerics for Function Space Problems. Academic Press, New York, 1984.

4. E. J. McShane. Integration. Princeton University Press, Princeton, N. J., 1944.

5. R. E. Moore. Interval Analysis. Prentice-Hall, Englewood Cliffs, N. J., 1966.

6. R. E. Moore. Methods and Applications of Interval Analysis. SIAM Studies in Applied Mathematics, Vol. 2, SIAM, Philadelphia, 1979.

7. L. B. Rall. Numerical integration and the solution of integral equations by the use of Riemann sums. SIAM Rev. 7 (1965), 55-64.

8. L. B. Rall. Interval bounds for stationary values of functionals. Nonlinear Anal. 6 (1982), 855-861.

9. L. B. Rall. A theory of interval iteration. Proc. Amer. Math. Soc. 86 (1982), 625-631.

10. L. B. Rall. Interval methods for fixed-point problems. MRC Tech. Summary Rept. No. 2583, Univ. of Wisconsin-Madison, 1983.

11. L. B. Rall. Mean value and Taylor forms in interval analysis. SIAM J. Math. Anal. 14 (1983), 223-238.

12. L. B. Rall. Application of interval integration to the solution of integral equations. J. Integral Equations 6 (1984), 127-141.

13. P. Wisskirchen. Vergleich intervallarithmetischer Iterationsverfahren. Computing 14 (1975), 45-49.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>2816 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>COMPUTABLE BOUNDS FOR SOLUTIONS OF INTEGRAL EQUATIONS | | 5. TYPE OF REPORT & PERIOD COVERED<br>Summary Report - no specific reporting period<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>L. B. Rall | | 8. CONTRACT OR GRANT NUMBER(s)<br>DAAG29-80-C-0041 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Mathematics Research Center, University of Wisconsin<br>610 Walnut Street<br>Madison, Wisconsin 53706 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>Work Unit Number 3 - Numerical Analysis and Scientific Computing |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>U. S. Army Research Office<br>P.O. Box 12211<br>Research Triangle Park, North Carolina 27709 | | 12. REPORT DATE<br>May 1985<br>13. NUMBER OF PAGES<br>12 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report)<br>UNCLASSIFIED<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Integral equations
Eigenvalue problems
Error bounds
Interval integration
Interval iteration

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Interval integration is used to obtain inclusions of integral operators of the form

$$g(u)(s) = \int_T g(s,t,u(s),u(t))dt \qquad (1)$$

which can be carried out on a computer. The resulting inclusions, combined with interval iteration, are used to compute guaranteed upper and lower bounds for

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT - cont'd.

solutions of integral equations of the form

$$u = g(u) \qquad (2)$$

for $s \in S$. It is also possible to establish existence or nonexistence of solutions of integral equations in given regions on the basis of results of the computation. Examples of applications of this technique to linear and nonlinear integral equations are eigenvalue problems for linear integral operators are given.

END

FILMED

9-85

DTIC