

## ALGORITHM 604 A FORTRAN Program for the Calculation of an Extremal Polynomial

FREDERICK W. SAUER MRC, University of Wisconsin

Categories and Subject Descriptors: G.1.2 [Numerical Analysis] Approximation—minimax approximation and algorithms; G.1.3 [Numerical Analysis] Numerical Linear Algebra—linear systems (direct and iterative methods), G.m [Mathematics of Computing]. Miscellaneous—FOR-TRAN

**General Terms: Algorithms** 

Additional Key Words and Phrases: Remes, extremal polynomial, Richardson iteration

## DESCRIPTION

Let S be the union of finitely many compact intervals in **R** such that  $S \cap (-\infty, 0] \neq \emptyset$ ,  $S \cap [0, +\infty) \neq \emptyset$ , and  $0 \notin S$ . Let  $\lambda$  be the linear functional on  $\pi_n$  (:= the polynomials of degree *n* or less) whose value at *p* is *p*(0). An *extremal* for  $\lambda$  is any polynomial of norm 1 at which  $\lambda$  takes on its norm, that is,  $p \in \pi_n$ ,  $||p||_S = 1$ , and  $\lambda p = ||\lambda||$ . Here

$$\|p\|_{S} := \max_{s \in S} |p(s)|$$

and

$$\|\lambda\| := \sup_{p \in \pi_n, \|p\|_{s}=1} |\lambda p| = 1/\min\{\|p\|_{s} : p \in \pi_n, p(0) = 1\}.$$

In [1] it is proved that for a given S and n there exists a unique extremal polynomial. The program presented here is a FORTRAN implementation of the Remes algorithm outlined in [1] to calculate the extremal polynomial given S and n.

The extremal polynomial can be used for the determination of the iteration parameters  $\alpha_n$  for Richardson's iteration

$$\mathbf{x}^{n+1} = \mathbf{x}^n - \alpha_n (A\mathbf{x}^n - \mathbf{b})$$

© 1983 ACM 0098-3500/83/0900-0381 \$00.75

ACM Transactions on Mathematical Software, Vol 9, No 3, September 1983, Pages 381-383

Received 11 November 1980; revised 15 December 1982; accepted 29 December 1982

This work was sponsored by the U.S. Army under Contract DAAG29-80-C-0041.

Author's address, Mathematics Research Center, University of Wisconsin, 610 Walnut Street, Madison, WI 53706.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

to solve a linear system of equations  $A\mathbf{x} = \mathbf{b}$ . Defining  $\mathbf{e}^n := \mathbf{x}^n - \mathbf{x}$ , the error in the *n*th iterate, we have

$$\mathbf{e}^{n} = (1 - \alpha_{n-1}A)\mathbf{e}^{n-1} = \cdots = \prod_{j=1}^{n} (1 - \alpha_{j-1}A)\mathbf{e}^{0} = Q_{n}(A)\mathbf{e}^{0},$$

where  $Q_n$  is the polynomial of degree *n*, which vanishes at  $1/\alpha_0, \ldots, 1/\alpha_{n-1}$  and is 1 at 0. If the spectrum of *A* is known to lie in some compact set *S*, then one should choose the parameters so as to minimize  $||Q_n||_S$ ; that is, one should choose the parameters to be the inverse of the roots of the extremal polynomial for *n* and *S*. The resulting polynomial  $P_n$  is then the error in the best Chebyshev approximation on *S* from  $\{\sum_{j=1}^{n} \beta_j t^j\}$ .

A description of how to use this program is given in the comments in subroutine **EXTREM**. In addition, two sample programs are provided. The first is a simple example how **EXTREM** can be called. The second uses Richardson's iteration to solve the linear system  $(B^2 - \mu I)x = b$ , with  $\mu = \sqrt{3}$  and B the tridiagonal matrix with 2 on the diagonal and -1 on the off-diagonals.

## REFERENCES

1. DE BOOR, C., AND RICE, J.R. Extremal polynomials with applications to Richardson iteration for indefinite linear systems. SIAM J. Sci. Stat. Comput. 3, 1 (1982), 47-57.

## ALGORITHM

c c

[A part of the listing is printed here. The complete listing is available from the ACM Distribution Service (see page 385 for order form).]

SUBROUTINE EXTREM(X, N, XINT, NINT, K, IPRINT, MAXIT, EPS2, EPS3, EXT 10 \* EPS4, XTILDE, ALPHA, PROD, PZERO, IERR) EXT 2Ø THIS SUBROUTINE FINDS A POLYNOMIAL WHICH IS AN EXTREMAL OF THE LINEAREXT зø FUNCTIONAL WHICH IS THE POINT EVALUATION AT THE POINT ZERO. THE EXT 4Ø POLYNOMIAL IS CHOSEN FROM THE SPACE OF POLYNOMIALS OF DEGREE LESS 5Ø EXT THAN OR EQUAL TO N, WHERE N IS A PARAMETER SET BY THE USER. THE NORM EXT 6Ø ON THIS SPACE IS DEFINED TO BE: 70 EXT EXT 8Ø NORM(P) = MAX(ABS(P(T)) : T IS IN S)9Ø EXT EXT 100 WHERE S IS THE UNION OF THE CLOSED INTERVALS (XINT(1,I),XINT(2,I)), EXT 110 I = 1, 2, ..., NINT.EXT 120 EXT 130 THE REMES ALGORITHM (OUTLINED BELOW) IS USED TO FIND THIS EXTREMAL EXT 140 POLYNOMIAL. EXT 150 EXT 16Ø EXT 170 PARAMETERS EXT 180 EXT 190 х A REAL ARRAY OF DIMENSION N+1. ON INPUT, X CONTAINS A EXT 200 STRICTLY INCREASING SEQUENCE OF POINTS IN S WITH X(K)=B AND EXT 21Ø X(K+1)=C, WHERE B AND C ARE DEFINED: EXT 22Ø B = MAX (T : T IS IN S, AND T)C = MIN (T : T IS IN S, AND T)EXT .LE. 0) .GE. 0). 230 EXT 24Ø ON RETURN, X CONTAINS THE FINAL SEQUENCE WHICH DEFINES THE EXT 25Ø EXTREMAL POLYNOMIAL WHICH IS THE UNIQUE POLYNOMIAL SUCH THAT: EXT 260 EXT 27Ø (-1) \* \* (K-J)IF J .LE. K EXT 280 P(X(J)) =EXT 290 (-1) \*\* (J+1+K) IF J .GT. K. EXT 300 EXT 31Ø A SUITABLE STARTING SEQUENCE FOR X CAN BE OBTAINED BY 320 EXT CALLING SUBROUTINE SETUP. EXT 330

ACM Transactions on Mathematical Software, Vol 9, No 3, September 1983

Algorithms · 383

				-
С	N	DEGREE OF EXTREMAL POLYNOMIAL. IF THIS IS TOO LARGE	EXT	340
С		OVERFLOW, UNDERFLOW, OR DIVISION BY ZERO ARE LIKELY	EXT	35Ø
С	XINT	TWO DIMENSIONAL ARRAY OF DIMENSION 2 BY NINT. ON INPUT, XINT	EXT	360
С		CONTAINS THE ENDPOINTS OF THE INTERVALS DEFINING S (SEE ABOVE	EEXT	370
č		DEFINITION OF S), XINT MUST BE IN INCREASING ORDER.	EXT	380
č		XINT(1,1) is $XINT(2,1)$ if $XINT(1,2)$ is $XINT(2,2)$	ÈYT	300
č		I = V I M (1) I I M (2) I = V I M (2) I = 0 I M (2) I =	5VM	100
ž		$\mathcal{L}$	EV4	400
ž	1171100	AINI MUSI DE SUCH INAI S CONTAINS AI LEASI NHI POINIS.	EA1	410
C C	NINT	NUMBER OF INTERVALS IN S.	EXT	420
C	ĸ	INTEGER SUCH THAT $X(K) = B$ AND $X(K+1) = C$ .	EXT	430
С	IPRINT	INTEGER WHICH ON INPUT HAS THE FOLLOWING MEANINGS	EXT	440
С		=-1 SUPPRESS ALL PRINTING. PZERO WILL NOT BE CALCULATED	EXT	450
С		= $\emptyset$ PZERO = P( $\emptyset$ ) WILL BE CALCULATED AND PRINTED	EXT	46Ø
С		=1 PZERO AND THE ROOTS OF THE POLYNOMIAL WILL BE	EXT	470
С		CALCULATED AND PRINTED	EXT	480
С		=2 IN ADDITION, FOR EACH ITERATION THE CURRENT VALUE OF	EXT	490
č		THE ARRAY X WILL BE PRINTED	EXT	500
č	MAXIT	MAXIMUM NUMBER OF ITERATIONS ALLOWED	EXT	510
č	EPS2	A TOLERANCE USED AS A STOPPING CRITERION FOR THE MAJOR	EYT	520
č		TTERATION THE BOUTTNE WILL STOD WHEN	EVA	520
č		MAY/ABC/VOID(1) + V(1) + 1 2 Mill Im EDC2	DVM	540
č		$MAA (ADS (AOLD (1)^{-A}(1))) (1^{-1}(2),, (N+1) D1, DFS2$	DVM	540
č	BD03 BT	WHERE AULD IS THE A ARRAY OF THE PREVIOUS THERATION,	EA1	550
8	EPS3,EF	AL TOLERANCE PARAMETERS USED BY SUBROUTINE REGULA. THE	EXT	200
č		BE WITHIN FDS: OF AN ACTUAL DOOR OF ADC(D(VDM)) WITH DE FECC	EV1	5/0
č		THAN FOR VALUES FOR FORS AND FOR A NED NOM BE DOUTDED TH	DVD	500
č		IDENTITY FOUNT TO -1 OF A	EXT	290
Ť		TRAINT IS BOOKD TO -I OK D.	EXT	000
С	XTILDE	ARRAY OF DIMENSION N+1. ON OUTPUT, IF IPRINT IS EQUAL TO 1 OF	EXT	610
С		2, XTILDE WILL CONTAIN THE RECIPROCALS OF THE ROOTS OF THE	EXT	620
С		EXTREMAL POLYNOMIAL. THE RECIPROCALS OF THE ROOTS IN THE	EXT	630
Ċ		INTERVAL (X()), X(N+1)) WILL BE STORED IN THE ELEMENTS	EXT	640
č		XTIDE(2) $XTIDE(3)$ $XTIDE(N)$ THE DECIDENCEL OF THE	EXT	650
č		$\Delta T = \Delta T $	FYT	660
č		TE TH DOC NOW EVICE VMIDES (1) MILL DE STOND IN ATTENDITY.	EYT	670
č	AT DUA	TE IL DUES NUL CAISI, AILDE (I WILD DE SEL TO ADAO.	ENI EV#	600
ž	ALPRA	ARRAI OF DIMENSION NEI USED BI THIS ROUTINE TO STORE THE	DVM	600
č		VALUES (AS CALCULATED BY ALCALC)	EAT	700
C .			EXT	700
C.		N+1	EXT	/10
С		ALPHA(I) = P(X(I)) / PRODUCT (X(I)-X(J))	EXT	720
С		J=1	EXT	730
С		J.NE.I	EXT	740
С			EXT	750
С		WHERE P(X) IS THE VALUE OF THE POLYNOMIAL AT X. IN PARTICULAR,	EXT	760
С			EXT	77Ø
С		(-1)**(I-K) IF I .LE. K	EXT	780
С		P(X(I)) =	EXT	790
с		(-1) * * (I+1-K) IF I .GT. K	EXT	800
С			EXT	81Ø
č		ON RETURN, IF IPRINT IS UNEQUAL TO ~], ALPHA WILL CONTAIN	EXT	820
č		THESE VALUES FOR THE EXTREMAL POLYNOMIAL.	EXT	830
č			EXT	840
č	PROD	ARRAY OF DIMENSION N+1 USED BY THIS ROUTINE TO STORE	EXT	850
č	INOD	THE VALUES (AS CALCULATED BY ALCALC)	FYT	860
č		THE VALUES (AS CALCULATED BI ABCALC)	EV.	070
č		N - 1	EA1	0/10
č		NTI	DVM	000
ž		PROD(1) = PRODUCT (X(1) - X(3))	DVM	0.90
č			DVO	900
C		J.NE.I	EXT	910
č			EXT	920
C		ON RETURN, IF IPRINT IS UNEQUAL TO -1, PROD WILL CONTAIN	EXT	930
C		THESE VALUES FOR THE EXTREMAL POLYNOMIAL.	EXT	940
C			EXT	950
С	PZERO	ON OUTPUT, IF IPRINT IS UNEQUAL TO -1, PZERO WILL CONTAIN	EXT	960
С		THE VALUE OF THE EXTREMAL POLYNOMIAL AT $\emptyset$ .	EXT	970
С			EXT	980
С	IERR	ON RETURN, IERR WILL BE THE SUM OF THE VALUES OF THE	EXT	990
С		FOLLOWING POSSIBLE USER INPUT ERRORS. IERR=Ø IS THE	EXT	1000
С		NORMAL RETURN. IF IERR IS NOT ZERO THE ROUTINE WILL	EXT	1010
С		RETURN IMMEDIATELY WITHOUT CALCULATING THE EXTREMAL	EXT	1020
С		POLYNOMIAL	EXT	1030
С			EXT	1040

ACM Transactions on Mathematical Software, Vol 9, No. 3, September 1983.