

AD-A142 823 A COMPARATIVE STUDY OF LINEAR ARRAY SYNTHESIS TECHNIQUE 1//

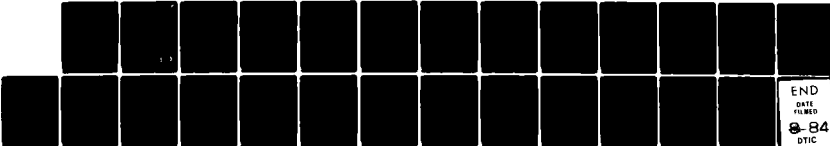
USING A PERSONAL COMPUTER(U) NAVAL RESEARCH LAB
WASHINGTON DC S R LAXPATI ET AL. 29 MAY 84 NRL-MR-5336

UNCLASSIFIED

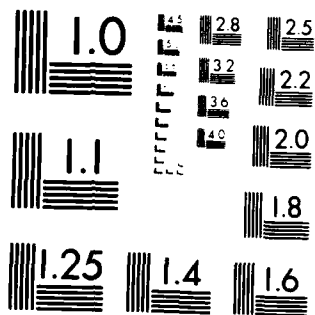
SBI-AD-E000 574

F/G 17/9

NL



END
DATE
FILMED
8-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

ADE 000574

NRL Memorandum Report 5336

A Comparative Study of Linear Array Synthesis Technique Using a Personal Computer

S. R. LAXPATI, J. P. SHELTON,* AND M. A. BURNS**

*Electromagnetics Branch
Radar Division*

**Symmetron, Inc.
Arlington, VA 22201*

***Voice of America
Washington, DC 20547*

AD-A142 823

May 29, 1984

DTIC FILE COPY



NAVAL RESEARCH LABORATORY
Washington, D.C.

DTIC
ELECTE
JUL 02 1984
S D E

Approved for public release; distribution unlimited.

84 06 29 035

Handwritten mark

REPORT DOCUMENTATION PAGE				
1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION AVAILABILITY OF REPORT		
2b DECLASSIFICATION/DOWNGRADING SCHEDULE		Approved for public release; distribution unlimited.		
4 PERFORMING ORGANIZATION REPORT NUMBER NRL Memorandum Report 5336		5 MONITORING ORGANIZATION REPORT NUMBER		
6a NAME OF PERFORMING ORGANIZATION Naval Research Laboratory	6b OFFICE SYMBOL <i>If applicable</i> Code 5370	7a NAME OF MONITORING ORGANIZATION		
6c ADDRESS (City, State and ZIP Code) Washington, DC 20375		7b ADDRESS (City, State and ZIP Code)		
8a NAME OF FUNDING SPONSORING ORGANIZATION Naval Electronic Systems Command	8b OFFICE SYMBOL <i>If applicable</i>	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c ADDRESS (City, State and ZIP Code) Washington, DC 20360		10 SOURCE OF FUNDING NOS.		
11 TITLE (Include Security Classification) (See page ii)		PROGRAM ELEMENT NO. 62712N	PROJECT NO. XF12-141-100	WORK UNIT NO. 53-1854-A4
12 PERSONAL AUTHOR(S) S.R. Laxpati, J.P. Shelton,* and M.A. Burns**				
13a TYPE OF REPORT Interim	13b TIME COVERED FROM TO	14 DATE OF REPORT (Yr. Mo. Day) May 29, 1984	15 PAGE COUNT 26	
16 SUPPLEMENTARY NOTATION *Symmetron, Inc., Arlington, VA 22201 **Voice of America, Washington, DC 20547				
17 COSATI CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB GR	Array synthesis Taylor array computer codes	
			Chebyshev array	
19 ABSTRACT (Continue on reverse if necessary and identify by block number) <p>Five computer programs for synthesizing low-sidelobe sum patterns from linear arrays are evaluated in terms of run time and precision. Three of the programs are based on the Dolph-Chebyshev synthesis procedure, in which all sidelobes are set at the same level. The other two programs are based on a discretized version of the Taylor synthesis procedure, in which far-out sidelobes are allowed to decay. The programs were written for use on small 8- and 16-bit personal computers. It was found that the fastest running programs are also the most precise. The only Chebyshev program that gave satisfactory precision for arrays as large as 100 elements is based on Bresler's nested product algorithm, and the only similarly acceptable Taylor program is based on Shelton's discretized synthesis.</p>				
20 DISTRIBUTION AVAILABILITY OF ABSTRACT UNCLASSIFIED UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a NAME OF RESPONSIBLE INDIVIDUAL S. R. Laxpati		22b TELEPHONE NUMBER (Include Area Code) (202) 767-6277	22c OFFICE SYMBOL Code 5370	

SECURITY CLASSIFICATION OF THIS PAGE

11. TITLE (Include Security Classification)

**A COMPARATIVE STUDY OF LINEAR ARRAY SYNTHESIS TECHNIQUE USING A PERSONAL
COMPUTER**

SECURITY CLASSIFICATION OF THIS PAGE

CONTENTS

INTRODUCTION 1
DOLPH-CHEBYSHEV SYNTHESIS 1
TAYLOR SYNTHESIS 6
CONCLUSIONS 9
REFERENCES 9
APPENDIX 11

Accession For	
NTIS	<input checked="" type="checkbox"/>
DTIC	<input type="checkbox"/>
US	<input type="checkbox"/>
JA	<input type="checkbox"/>
By	
Distribution	
Avail	to

A-1

DTIC
COPY
INSPECTED
2

A COMPARATIVE STUDY OF LINEAR ARRAY SYNTHESIS TECHNIQUE USING A PERSONAL COMPUTER

INTRODUCTION

Procedures for synthesizing the radiation patterns of linear arrays based on the specification of its sidelobe structure are well established. One of these is the technique originally proposed by Dolph¹ and it provides for a uniform sidelobe level. Another technique, although developed for line sources, is due to Taylor² and can be adopted for linear arrays. The latter is popular due to its synthesized aperture distributions which are more readily realizable. A discrete version of the Taylor synthesis procedure is discussed by Shelton³.

Both Dolph-Chebyshev and Taylor synthesis techniques, fundamentally, rely on manipulation of the zeros of the linear array pattern function. The aperture distribution for the desired pattern function usually requires lengthy computations. In case of Dolph-Chebyshev synthesis, this problem has been addressed by several authors⁴⁻⁹ over the past few decades. The Taylor synthesis, in effect, uses a discrete Fourier Transform technique (called Woodward¹⁰ synthesis) to obtain the aperture distribution. These procedures do not have much in common; as a matter of fact, in case of an endfire Chebyshev array, expression for the element excitations are quite different from that for a broadside Chebyshev array. However, the knowledge of the pattern null locations in the above synthesis procedures can be used to develop a simple expression that is suitable for all cases. The expression is readily developed based on the convolution synthesis procedure discussed by Laxpati¹¹ for planar arrays.

With several alternate expressions being available for the aperture distribution of Dolph-Chebyshev and Taylor syntheses, it is desirable to undertake a study to make some recommendations as to the suitability of these expressions in numerical computation. Due to the increasing use of personal computers by antenna engineers, it is felt that an investigation of this nature should be confined to the computation using such small computers. Thus, in this paper, we present the results of a comparative study of various linear array synthesis techniques. In the next section, after a brief discussion of the three basic techniques for evaluation of Chebyshev coefficients, we discuss the accuracy and computation times associated with these techniques. The following section presents the results of the study involving two different techniques (one due to Shelton³ and the other using the convolution procedure) for Taylor synthesis. In the last section, some general observations about the investigation and on the results are offered.

DOLPH-CHEBYSHEV SYNTHESIS

Following Dolph's paper on Chebyshev synthesis, Barbieri⁴, Van Der Maas⁵, Salzev⁶, and Brown^{7,8} reported on alternative means of evaluating aperture distribution for Chebyshev arrays. Although they are not the same, the expressions by Barbieri, Salzev and Brown are

Manuscript approved February 21, 1984.

similar in that they express the current in an element in terms of a finite series of terms involving ratios of factorial functions and with alternating sign. The expression by Elliott¹² is representative of this group and is the one used in this work and is reproduced below. We shall call this the classical expression. Also, although our results are valid for odd or even number of elements, for simplicity, we will present examples of odd number of elements. Thus, all linear arrays discussed in the following have $(2N+1)$ elements; the element numbering scheme is shown in figure 1, where the elements are assumed to have a symmetric excitation leading to the broadside radiation.

Classical Technique:

$$I_n = \sum_{p=n}^N (-1)^{N-p} \frac{N}{N+p} \frac{\Gamma(N+p+1)}{\Gamma(N-p+1)\Gamma(p+n+1)\Gamma(p-n+1)} (u_0)^{2p} \quad (1)$$

$$n=0,1,2,\dots,N.$$

where $T_{2N}(u_0)=R$ and $SLL = 20 \log R$. Here, SLL is the desired sidelobe level in dB, $T_{2N}(x)$ is the Chebyshev polynomial of degree $2N$ and $\Gamma(x)$ is the Gamma function.

In contrast, the expression given by Van Der Maas involves terms of the same sign inside the summation. Bresler⁹ reformulated the expression into a recursive form using nested products. This, we feel is a distinctly different form of representation of the coefficients. Thus, we use this representation (called Nested Product Technique) in our comparison. This expression (in our notation) is shown below.

Nested Product Technique:

$$I_{N-n} = 2N \alpha NP(n, f_m, \alpha). \quad n=0,1,2, \quad (2)$$

$$\text{where } NP(n, f_m, \alpha) = \sum_{m=1}^n \alpha^{n-m} \prod_{j=m}^n f_j; \quad f_n \equiv 1.$$

$$\text{and } f_m = \frac{m(2N-2n+m)}{(n-m)(n+1-m)};$$

$$\text{also } \alpha = 1 - \frac{1}{u_0^2}.$$

The third technique is based on the convolution of three element canonical arrays¹¹. These canonical arrays have outer element excitations of unity, whereas the center element excitation c_j ; for $j=1,2,\dots,N$ is chosen such that the j th canonical array has a pattern null at the location of the j th symmetric zero pair of the Chebyshev

$$U = kd \sin \theta$$

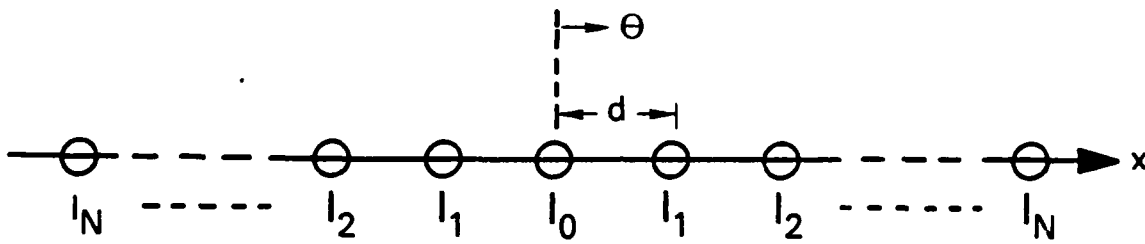


Figure 1 - $(2N + 1)$ Element Linear Array

polynomial. These arrays are then convolved to generate the large array. Convolution Technique:

$$w_{0j} = \cos \frac{(2j-1)}{2N} \pi ; j=1,2,\dots,N.$$

Where w_{0j} are the zeros of the Chebyshev polynomial $T_{2N}(w)$.

$$C_j = -2\cos u_{0j};$$

$$u_{0j} = 2\arccos (w_{0j}/u_0).$$

And the aperture distribution

$$I(x) = \sum_{n=-N}^N I_n \delta(x-nd) = f_1 * f_2 * \dots * f_N \quad (3)$$

where $f_j = \delta(x-d) + C_j \delta(x) + \delta(x+d)$.

Using these three expressions (equations (1), (2) and (3)), computer programs NESTED, CHEB and CONCHEB, respectively, were written to implement the Chebyshev synthesis. Different versions of the program suitable for implementation on different machines were written. These were two personal computers used in the numerical phase; one is an 8-bit Radio Shack TRS-80 Model II which has available an interpretive BASIC language. The other computer is a 16-bit NEC Advanced Personal Computer with BASIC and FORTRAN IV compilers. Also, in order to ascertain the numerical accuracy, some of the programs were run on a 32-bit mainframe computer (Texas Instrument's Advanced Scientific Computer at the Naval Research Laboratory) using double precision (REAL*8) arithmetic.

The computation was carried out for several different array sizes ranging from 15 to 99 elements; although, in principle, there is no limit to the size of arrays that may be synthesized. Furthermore, all designs specified a sidelobe level of 30 dB.

Figure 2 shows the run time, under FORTRAN, for the three aforementioned Chebyshev synthesis programs versus number of elements. The CHEB program was the slowest; but more importantly, the program failed to converge to the correct element excitations beyond 30 elements. Over 21 elements the accuracy of the excitation was only to 2 digits. When the program was run using double precision arithmetic it still failed to converge above 31 elements. This indicates that the classical technique inherently has a limitation as to the largest size of array that may be synthesized.

The convolution synthesis program, CONCHEB, although much faster than CHEB, certainly cannot compete with NESTED program in speed. Also, beyond 61 elements, the CONCHEB program failed to converge. The current version of the program convolves three-element arrays using zeros of the Chebyshev polynomial in an alternating sequence; i.e., the

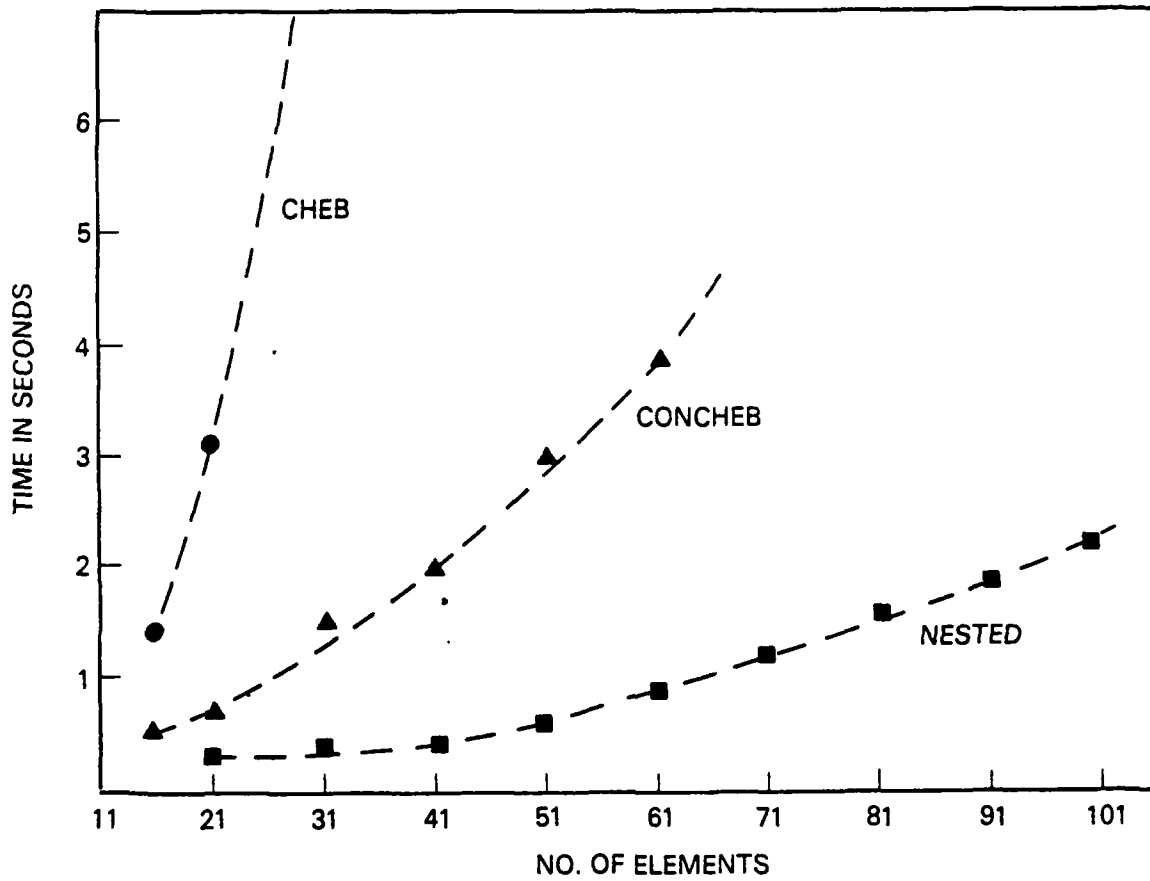


Figure 2 - Run Times for Chebyshev Programs -
 FORTRAN IV on NEC-APC

sequence in j is 1,N,2,N-1,3,---. No attempt was made to modify this convolution process to improve the accuracy; past experience with the convolution process indicates that some improvement may be possible. However, with reference to figure 2, it is obvious that the NESTED program is the most efficient one.

The results of the element excitations indicate that it is an extremely stable algorithm; provides a very good accuracy in single precision (six digit accuracy); and of course, it is very fast. This program, NESTED, was translated into BASIC and run on the TRS-80, Model II computer. The execution time ranged from two seconds for 21 element array to 36 seconds for a 99 element array. Although, the execution times in BASIC are about 15 to 20 times longer than that in FORTRAN, they are not significantly long to be of any major consequence. The NESTED program was also run using double precision (16 significant digits) on the mainframe computer. The total execution time for all 10 different arrays was less than 0.3 seconds!

Our experience with synthesis of various Chebyshev arrays using these three different techniques clearly demonstrates that the most important consideration on small computers is not the speed of execution but the accuracy of the final result. In this sense as well, the nested product algorithm proposed by Resler⁹ is the winner.

TAYLOR SYNTHESIS

Synthesis procedure proposed by Taylor² applies to a continuous aperture. In practice, this procedure is used for discrete aperture (arrays) by properly discretizing the continuous distribution. Shelton³ presented a synthesis procedure for discrete aperture distribution for Taylor type sidelobe structure. He expressed the pattern function in the form of a product function of zeros and then carried out the synthesis exactly analogous to that by Taylor; that is, to use the Woodward synthesis technique. In particular, for a 2N+1 element array, all 2N zeros are explicitly specified in the pattern function. Thus, analogous to the Chebyshev synthesis, this synthesis is amenable to the convolution procedure. In view of this, in the case of Taylor synthesis, we compare the two techniques; one proposed by Shelton and the other being the convolution synthesis. Before presenting and discussing the results of the investigation, the pertinent expressions for the two syntheses are given below. Once again, we will limit out discussion to arrays with odd (2N+1) number of elements.

Discrete Taylor (Shelton³) Technique:

$$u_{on} = \frac{2\pi\bar{n}}{(2N+1)} \frac{\sqrt{A^2+(n-1/2)^2}}{\sqrt{A^2+(\bar{n}-1/2)^2}}, \quad n=1,2,---,\bar{n}-1$$

$$= \frac{2\pi n}{(2N+1)}, \quad n=\bar{n},---,N. \quad (4)$$

where $A = \frac{1}{\pi} \cosh^{-1}(R)$; \bar{n} is equal to the number of near-in zeros that are moved in order to achieve the desired sidelobe ratio R (or equivalently the number of near-in sidelobes that are required at the specified level). The element excitations are

$$I_p = 1 + 2 \sum_{m=1}^{\bar{n}-1} a_m \cos \frac{2mp\pi}{2N+1}, \quad p=0, \dots, N. \quad (5)$$

where

$$a_m = E \left(\frac{2\pi m}{2N+1} \right);$$

$$E(u) = \prod_{n=1}^N \frac{(\cos u - \cos u_{0n})}{(1 - \cos u_{0n})}.$$

For the case of the convolution synthesis procedure, once the symmetric zero pairs are established, the excitation of the center element of a three element canonical array is readily determined. The procedure and expressions are analogous to the case of Chebyshev convolution synthesis. They are

zeros are $\pm u_{0j}$; $j=1, 2, \dots, N$

where u_{0j} are defined through equation (4), and the excitation

$$C_j = -2 \cos u_{0j}.$$

The synthesis of the large array is carried out using the convolution of three element arrays, chosen in the same alternating zero sequence as indicated for the Chebyshev array.

Based on these two procedures, computer codes STAYL and CONTAYL, respectively, were developed in FORTRAN using single precision arithmetic. Run time associated with these codes for $\bar{n} = 6$ and the sidelobe level of 30 dB for various number of elements from 15 to 99 were recorded and are shown in figure 3.

The program CONTAYL failed to converge, once again, for arrays with more than 71 elements and provided only two to three digit accuracy between 31 and 61 elements. These results are similar to the Chebyshev convolution synthesis. Even the run time data is very close.

The computation time associated with STAYL has an interesting behavior with increasing number of elements; it is almost linear. This is to be expected, since the number of computations to be carried out for each element is determined by \bar{n} and not $(2N+1)$. The corresponding growth for CONTAYL is exponential. Thus, for small number of elements CONTAYL may save some computation time but will suffer in accuracy as

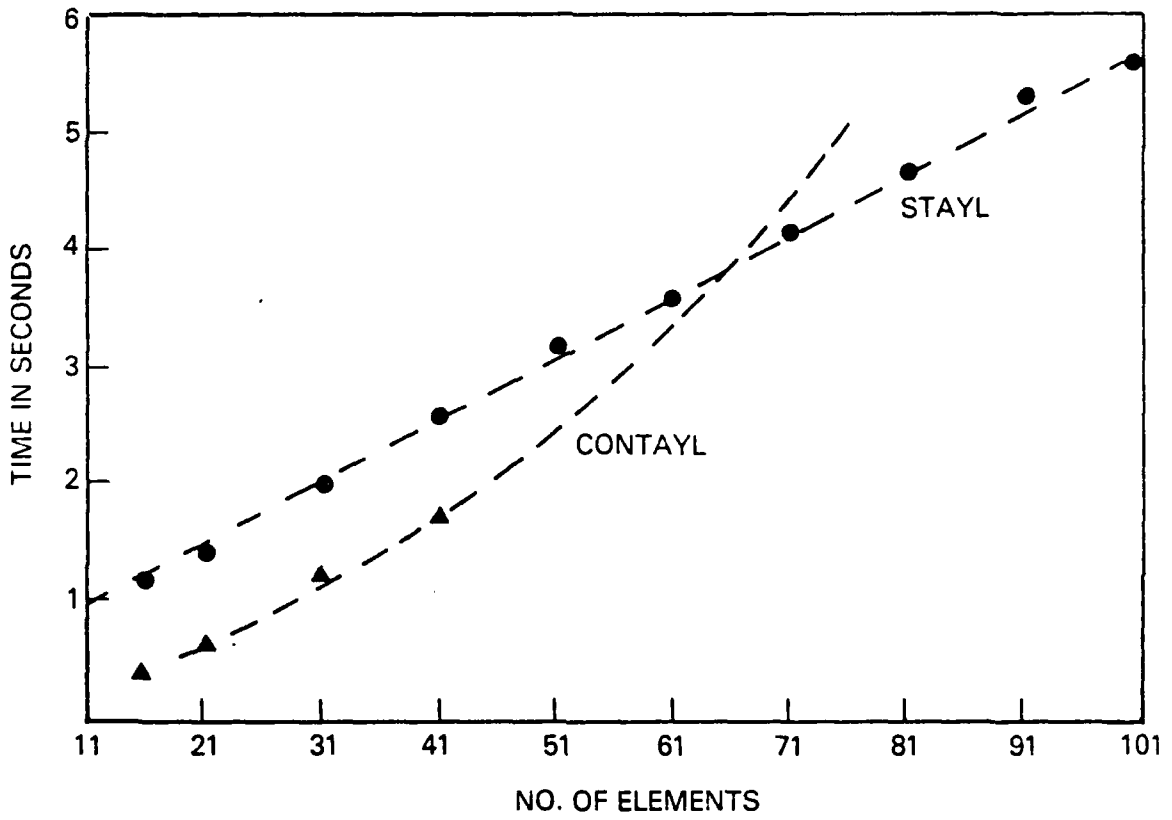


Figure 3 - Run Times for Taylor Programs -
 FORTRAN IV on NEC-APC

the number of elements increases. A check of STAYL program using double precision arithmetic on the mainframe computer indicates that it has five to six digit accuracy in single precision on a small computer.

It should be noted that the STAYL program code was developed by Shelton for the HP-41C, a pocket calculator. On this calculator, one has 10 significant digit capacity and thus the results obtained are more accurate than with a single precision FORTRAN Code. But, as one would expect, the HP-41C is very slow; it took approximately 5 minutes to synthesize a 31 element array.

STAYL Code was also run on NEC-APC using CBASIC, a compiler BASIC. In CBASIC, the computation times were significantly higher, ranging from 30 seconds for a 15 element array to 217 seconds for a 99 element array. However, the computation was carried out to 14 significant figures.

Once again, as with Chebyshev synthesis, we find the overriding consideration in Taylor synthesis is not the computation time, but the accuracy of the results. In this sense, Shelton's procedure is most efficient.

CONCLUSIONS

As is often the case with engineering investigations, the most significant results presented in this paper are not what we were looking for when we began the project. We were originally interested in evaluating computer run times for the various programs. However, two points soon became apparent -- first, most of the programs run fast enough, even on small machines, so that run time is not a major concern, and second, only two of the programs give adequate precision for the range of array size that was investigated. It is concluded that Bresler's nested product algorithm gives excellent results in terms of speed and precision, and also that Shelton's discretized procedure allows precise Taylor synthesis for all sizes of arrays. Finally, it is noted that the programs are very brief; the FORTRAN computer codes for all five programs are included in the appendix and the codes in BASIC are also available from the authors.

REFERENCES

1. C.L. Dolph, "A Current Distribution for Broadside Arrays Which Optimizes the Relationship Between Beam Width and Side-Lobe Level", Proc. IRE, 34, No.6, pp. 335-348, June 1946.
2. T.T. Taylor, "Design of Line-Source Antennas for Narrow Beamwidth and Low Side Lobes", IRE Tran. Antennas and Propagat., AP-3, pp. 16-28, No. 1, January 1955
3. J.P. Shelton, "Synthesis of Taylor and Bayliss Patterns for Linear Antenna Arrays", NRL Report 8511, Naval Research Laboratory, Washington, D.C. 20375, August 31, 1981. (AD-A103 534)

4. D. Barbieri, "A Method of Calculating the Current Distribution of Tschebyscheff Arrays", Proc. IRE, 40, No. 1, pp. 78-82, January 1952.
5. G.J. Van Der Maas, "A Simplified Calculation for Dolph-Tchebycheff Arrays", J. App. Phys., 25, No. 1, pp. 121-124, January 1954.
6. H.E. Salzer, "Note on the Fourier Coefficients for Chebyshev Patterns", Proc. IEE (London), 103C, pp. 286-288, February 1956.
7. J.L. Brown, Jr., "A Simplified Derivation of the Fourier Coefficients for Chebyshev Patterns", Proc. IEE (London), 105C, pp. 167-168, November 1957.
8. J.L. Brown, Jr., "On the Determination of Excitation Coefficients for a Tchebycheff Pattern", IRE. Tran. Antennas and Propagat., AP-10, pp. 215-216, March 1962.
9. A.D. Bresler, "A New Algorithm for Calculating the Current Distributions of Dolph-Chebyshev Arrays", IEEE Tran. Antennas and Propagat., AP-28, No. 6, November 1980. pp. 951-952.
10. P.M. Woodward, "A Method of Calculating the Field Over A Plane Aperture Required to Produce A Given Polar Diagram", Journal of IEE (London), Pt. III A, 93, pp. 1554-1558, 1946.
11. Sharad R. Laxpati, "Planar Array Synthesis with Prescribed Pattern Nulls", IEEE Trans. on Antennas and Propagat., AP-30, No. 6, pp. 1176-1183, November 1982.
12. R.S. Elliott, "Antenna Theory and Design", Prentice-Hall, Inc., pp. 143-147, 1981.

APPENDIX

In this appendix the FORTRAN IV computer codes for NEC-APC with supersoft FORTRAN compiler are listed. As noted in the main body of the report, the programs are brief; there are a number of "comment" statements in the listing and thus are easy to follow. No sample inputs or outputs are included.

Programs listed in the following are NESTED, CHEB, CONCHEB, STAYL and CONTAYL.

```

001      C                      PROGRAM NESTED
002      C  REVISED 01/28/84
002      C  CHEBYSHEV ARRAY ALGORITHM USING NESTED PRODUCTS FORMULATION
003      C  INPUT IS M = NO. OF ELEMENTS; SLL = SIDELobe LEVEL IN DB.
004          REAL I(100), NP, C(100)
005          WRITE (1,100)
006      100  FORMAT (' ENTER DATA: M,SLL')
007          READ (1,200) M,SLL
008      200  FORMAT ( IO,F0.0)
009          N = M / 2
010          TEST = (-1)**M
011          IF(TEST.GT.0) GO TO 10
012          N = (M - 1) / 2
013      10   R = 10. ** ( SLL / 20. )
014          ARCOSH = ALOG ( R + SQRT ( R**2. - 1 ) )
015          A = ARCOSH / (M-1)
016          ALPHA = ( TANH ( A ) ) ** 2.
017          I(N+1) = 1.0
018          I(N) = (M-1) * ALPHA
019          DO 30  K = 2, N
020              NP = 1.0
021              DO 20  J = 1, K-1
022                  FN = J * (M-1-2*K+J)
023                  FD = (K-J) * (K+1-J)
024                  F = FN/FD
025                  NP = NP * ALPHA * F + 1.
026      20   CONTINUE
027          I(N+1-K) = ( M-1 ) * ALPHA * NP
028      30   CONTINUE
029          DO 40  L = 1, N+1
030              C(N+L) = I(L)
031              C(N+2-L) = I(L)
032      40   CONTINUE
033          WRITE (4, 50)
034      50   FORMAT ( '                      CURRENTS' )
035          DO 60  L = 1, M
036              WRITE (4,70) L, C(L)
037      60   CONTINUE
038      70   FORMAT (10X, I2, 10X, F10.6)
039          STOP
040          END

```

```

001      C          PROGRAM CHEB
002      C REVISED 01/28/84
003      C BASED ON A CLASSIC METHOD OF COMPUTATION OF CHEBYSHEV
004      C EXCITATION VOLTAGES.
005      C REFERENCE ANTENNA THEORY AND DESIGN; ELLIOTT.
006      C ODD NUMBER OF ELEMENTS ONLY
007          REAL CC(100),C(100),CRNT(100)
008          WRITE (1,100)
009      100  FORMAT (' ENTER DATA: N, SLL')
010          READ (1,200) N,SLL
011      200  FORMAT (IO,FO.0)
012          M = N-1
013          MM = M/2
014          NN = (N+1)/2
015          PI = 3.1415927
016          R = 10. ** (SLL/20.0)
017          U = COSH (RCOSH (R)/ FLOAT (M) )
018          DO 10  I = 1,NN
019              II = I -1
020              C(I) = 0.0
021          DO 20  J = I,NN
022              JJ = J - 1
023              A = FLOAT (NN + JJ)
024              GA = GAMALN (NN + JJ)
025              GB = GAMALN (NN - JJ)
026              GE = GAMALN (J - II)
027              GD = GAMALN (J + II)
028              UP = U ** (2*JJ)
029              SIGN = (-1) ** (NN-J)
030              TL = EXP (GA - GB -GE - GD)
031              TN = UP*SIGN*(2.*NN-1.)/(2.*A)
032              T = TL * TN
033              C(I) = C(I) + T
034      20   CONTINUE
035      10   CONTINUE
036          DO 12 J = 1,NN
037              CC(J) = C(J)/ C(NN)
038      12   CONTINUE
039          DO 13 J = 1,NN
040              CRNT (NN-1+J) = CC(J)
041              CRNT (NN+1-J) = CC(J)
042      13   CONTINUE
043          WRITE (4,30)
044      30   FORMAT ('          CURRENTS')
045          DO 14 I = 1,N
046              WRITE (4,300) I, CRNT(I)
047      14   CONTINUE
048      300  FORMAT (10X,I2,15X,F12.8)
049          STOP
050          END

```

```

051 C*****
052 C INVERSE HYPERBOLIC FUNCTION
053 C*****
054     FUNCTION RCOSH (R)
055         RCOSH = ALOG(R + SQRT(R*R - 1.0))
056     RETURN
057     END
058 C*****
059 C HYPERBOLIC FUNCTION
060 C*****
061     FUNCTION COSH (R)
062         Y = EXP (R)
063         COSH = (Y + (1.0/Y) ) /2.
064     RETURN
065     END
066 C*****
067 C     GAMALN FUNCTION
068 C*****
069     FUNCTION GAMALN (K)
070         GAMALN = 0.0
071         IF (K .EQ. 0) RETURN
072         FACT = 0.0
073         TPL = 0.91893853
074         AL = K
075     10  IF (AL .GE. 10.0) GO TO 20
076         FACT = FACT + ALOG (AL)
077         AL = AL + 1.0
078         GO TO 10
079     20  TERM = (AL - 0.5) * ALOG(AL) - AL + TPL
080         1 + 1.0/(12.*AL) - 1.0/(360.0 * AL**3) + 1.0/(1260.*
081         2 AL**5) - 1.0/ (1680. * AL**7)
082         GAMALN = TERM - FACT
083     RETURN
084     END

```

```

001             PROGRAM CONCHEB
002 C*****
003 C* LINEAR      ARRAY SYNTHESIS USING CONVOLUTION METHOD.
004 C* CHEBYSHEV SIDELobe DESIGN
005 C* ODD NUMBER OF ELEMENTS.
006 C*****
007         REAL PSI(100)
008         REAL C(100),AA(100),A1(100),A2(100),A3(100),CONV(100)
009         DATA A1/100*1./, A3/100*1./, CONV/100*0./, AA/100*0./
010 C*****
011 C* N = NUMBER OF ELEMENTS IN THE ARRAY. MUST BE ODD!!
012 C* SLL = SIDE LOBE LEVEL IN DBS.
013 C*****
014         PI = 3.1415297
015         WRITE (1,100)
016 100  FORMAT (' ENTER DATA: N,SLL')
017         READ (1,200) N,SLL
018 200  FORMAT (IO,FO.0)
019         M = N-1
020         NN = (N+1)/2
021         MM = (N-1)/2
022         MD = (MM/2) + 1
023 20  CALL CHEBX(PI,M,NN,SLL,MM,PSI)
024 C*****
025 C* CREATE THREE ELEMENT ARRAYS
026 C*****
027 30  DO 40 I = 1,MM
028     A2(I) = -2. * COS(PSI(I))
029 40  CONTINUE
030 C*****
031 C* REPEATED CONVOLUTION OF 3-ELEMENT ARRAYS
032 C*****
033         CONV(1) = A1(1)
034         CONV(2) = A2(1)
035         CONV(3) = A3(1)
036         L = 1
037         K = 5
038         LX = 0
039 50  LL = NN-L
040         LX = LX+1
041 60  L = LL
042         C(1) = A1(L)
043         C(2) = A2(L)
044         C(3) = A3(L)
045         DO 70 I = 1,3
046         DO 70 J = I,K
047             JJ = J-I+1
048             AA(J) = AA(J) + CONV(JJ)*C(I)
049 70  CONTINUE
050         DO 80 I = 1,K
051             CONV(I) = AA(I)
052             AA(I) = 0.0
053 80  CONTINUE
054         K = K+2

```

```

055         IF (L.EQ.MD) GO TO 90
056         LL = NN+1-L
057         LSUM = L+LX
059         IF (LSUM.EQ.NN) GO TO 60
060         GO TO 50
061     90    CONTINUE
062         WRITE (4,600)
063     600   FORMAT ( '          CURRENTS'/)
064         DO 120 I = 1,N
065           WRITE (4,700) I,CONV(I)
066     120   CONTINUE
067     700   FORMAT (10X,I2,10X,F10.6)
068         STOP
069         END
070         FUNCTION COSH(R)
071         Y = EXP(R)
072         COSH = (Y + (1.0/Y))/2.
073         RETURN
074         END
075     C*****
076     C* INVERSE HYPERBOLIC COSINE FUNCTION
077     C*****
078         FUNCTION RCOSH(R)
079         RCOSH = ALOG(R + SQRT(R*R - 1.0))
080         RETURN
081         END
082     C*****
083     C* CHEBYSHEV ZEROS
084     C*****
085         SUBROUTINE CHEBX(PI,M,NN,SLL,MM,PSI)
086         REAL X(50),PSI(100)
087         R = 10.0 ** (SLL/20.)
088         B = COSH(RCOSH(R)/M)
089         DO 10 I = 1,NN
090           J = I-1
091           X(I) = COS(PI*(2.*J + 1.)/(2.*M))
092     10    CONTINUE
093         DO 20 J = 1,MM
094           II = NN-1+J
095           JJ = NN-J
096           Y = X(J) / B
097           PSI(II) = 2.*ATAN(SQRT(1-Y*Y)/Y)
098           PSI(JJ) = PSI(II)
099     20    CONTINUE
100         RETURN
101         END

```

```

001      C          PROGRAM STAYL
002      C REVISED 01/28/84
003      C THIS PROGRAM COMPUTES ELEMENT EXCITATIONS FOR TAYLOR
004      C TYPE SIDELOBES USING SYNTHESIS EXPRESSIONS OF SHELTON
005      DIMENSION Z(100),AM(100),EN(100),EX(100)
006      WRITE (1,100)
007      100  FORMAT (' ENTER N,NBAR,SIDELOBE LEVEL FOR TAYLOR'
008             1  , ' SYNTHESIS' )
009      READ (1,200) N, NBAR, SLL
010      200  FORMAT (2I0,F0.0)
011      WRITE (4,300) N, NBAR, SLL
012      300  FORMAT (' TAYLOR SYNTHESIS - SHELTON' / ' N=',I5,
013             1  2X,'NBAR=',I5,2X,'SIDELOBE=', F5.2)
014      AL2 = 0.30102999566398
015      ALE = 0.43429448190325
016      PI = 3.14159265358979
017      M = (N-1)/2 + 0.1
018      IE = 1
019      IF (N .EQ. (2*M+1)) IE = 0
020      A = (SLL + 20.0*AL2) / (20.0*PI*ALE)
021      XN = FLOAT (N)
022      XN12 = FLOAT (NBAR) - 0.5
023      N1 = NBAR - 1
024      ALPHA = SQRT (A*A + XN12 * XN12)
025      DO 1 I=1,N1
026          XI12 = FLOAT (I) - 0.5
027          BETA = SQRT (A*A + XI12 * XI12)
028          Z(I) = ((2.0*PI/XN)/ALPHA) * FLOAT (NBAR) * BETA
029      1  CONTINUE
030      DO 2 I=NBAR,M,1
031          Z(I) = ( FLOAT (I) * 2.0*PI)/XN
032      2  CONTINUE
033      EO = 1.0
034      DO 3 I=1,M
035      3  EO = EO * (1.0-COS(Z(I)))
036      DO 5 I=1,N1
037          AM(I) = 1.0
038          DELTA = (2.0*PI * FLOAT(I))/XN
039          IF (IE .EQ. 1) AM(I) = COS (DELTA/2.0)
040          DO 4 J=1,M
041      4  AM(I) = AM(I) * (COS(DELTA) - COS(Z(J)))
042          AM(I) = AM(I)/EO
043      5  CONTINUE
044      DO 6 I=1,M+1
045          XI = 2*I - 2
046          IF (IE .EQ. 1) XI = XI + 1
047          EN(I) = 0.0
048          DO 7 J=1,N1
049          XJ = FLOAT (J)
050          EN(I) = AM(J) * COS((PI*XI*XJ)/XN) + EN(I)
051      7  CONTINUE
052          EN(I) = 2.0 * EN(I) + 1.0
053      6  CONTINUE
054      DO 50 K=1,M+1

```

```
055          L = N+1-K
056          EX(K) = EN(M+2-K)/EN(M+1)
057      50    EX(L) = EX(K)
058          WRITE (4,301)
059          WRITE (4,55) (I,EX(I), I=1,N)
060      301   FORMAT (' ELEM. NO.', 3X, 'EXCITATION')
061      55   FORMAT (5X,I2,5X,F14.7)
062          STOP
063          END
```



```

001      C                      PROGRAM CONTAYL
002      C*****
003      C* LINEAR          ARRAY SYNTHESIS USING CONVOLUTION METHOD.
004      C* TAYLOR SIDELobe DESIGN
005      C* ODD NUMBER OF ELEMENTS.
006      C*****
007          REAL PSI(100)
008          REAL C(100),AA(100),A1(100),A2(100),A3(100),CONV(100)
009          DATA A1/100*1./, A3/100*1./, CONV/100*0./, AA/100*0./
010      C*****
011      C* N = NUMBER OF ELEMENTS IN THE ARRAY. MUST BE ODD!!
012      C* SLL = SIDE LOBE LEVEL IN DBS.
013      C*****
014          PI = 3.1415297
015          WRITE (1,100)
016      100  FORMAT (' ENTER DATA: N,NBAR,SLL')
017          READ (1,200) N,NBAR,SLL
018      200  FORMAT (2I0,F0.0)
019          M = N-1
020          NN = (N+1)/2
021          MM = (N-1)/2
022          MD = (MM/2) + 1
023      10  CALL TAYLX(MM,SLL,PI,PSI,N,NN,M,NBAR)
024      C*****
025      C* CREATE THREE ELEMENT ARRAYS
026      C*****
027      30  DO 40 I = 1,MM
028          A2(I) = -2. * COS(PSI(I))
029      40  CONTINUE
030      C*****
031      C* REPEATED CONVOLUTION OF 3-ELEMENT ARRAYS
032      C*****
033          CONV(1) = A1(1)
034          CONV(2) = A2(1)
035          CONV(3) = A3(1)
036          L = 1
037          K = 5
038          LX = 0
039      50  LL = NN-L
040          LX = LX+1
041      60  L = LL
042          C(1) = A1(L)
043          C(2) = A2(L)
044          C(3) = A3(L)
045          DO 70 I = 1,3
046              DO 70 J = I,K
047                  JJ = J-I+1
048                  AA(J) = AA(J) + CONV(JJ)*C(I)
049      70  CONTINUE
050          DO 80 I = 1,K
051              CONV(I) = AA(I)
052              AA(I) = 0.0
053      80  CONTINUE
054          K = K+2

```

```

055         IF (L.EQ.MD) GO TO 90
056         LL = NN+1-L
057         LSUM = L+LX
058         IF (LSUM.EQ.NN) GO TO 60
059         GO TO 50
060     90     CONTINUE
061     C     WRITE (4,401)
062     C 401  FORMAT ( '           PSI ZEROS'//)
063     C     DO 110 I = 1,M
064     C       WRITE (4,500) I,PSI(I)
065     C 110  CONTINUE
066     500   FORMAT (10X,I2,10X,F10.6)
067     WRITE (4,600)
068     600   FORMAT ( '           CURRENTS'//)
069     DO 120 I = 1,N
070     WRITE (4,700) I,CONV(I)
071     120   CONTINUE
072     700   FORMAT (10X,I2,10X,F10.6)
073     STOP
074     END
075     FUNCTION COSH(R)
076     Y = EXP(R)
077     COSH = (Y + (1.0/Y))/2.
078     RETURN
079     END
080     C*****
081     C* INVERSE HYPERBOLIC COSINE FUNCTION
082     C*****
083     FUNCTION RCOSH(R)
084     RCOSH = ALOG(R + SQRT(R*R - 1.0))
085     RETURN
086     END
087     C*****
088     C* COMPUTATION OF TAYLOR ZEROS
089     C*****
090     SUBROUTINE TAYLX(MM,SLL,PI,ZEROS,N,NN,M,NBAR)
091     REAL ZERO(50), ZEROS(100), MEMA,MEMB
092     NBAR1 = NBAR-1
093     A = (SLL + 6.0202)/27.2875
094     C*****
095     C* COMPUTE ZEROS FROM 1 TO NBAR
096     C*****
097     DO 10 I = 1,NBAR1
098     RI = I
099     MEMA = (A*A) + ((RI-.5)**2.)
100     MEMB = (A*A) + ((NBAR-.5)**2.)
101     ZERO(I) = (((2.*PI)*NBAR)/N)*((SQRT(MEMA))/(SQRT(MEMB)))
102     10    CONTINUE
103     C*****
104     C* COMPUTE ZEROS FROM NBAR TO M
105     C*****
106     DO 20 I = NBAR,MM
107     RI = I
108     ZERO(I) = (2*PI*RI)/N

```

```
109      20  CONTINUE
110      C   WRITE (4,100)
111      C 100  FORMAT ( '                ZEROS' )
112      C       DO 30 I = 1,MM
113      C           WRITE (4,200) I,ZERO(I)
114      C 30   CONTINUE
115      C 200  FORMAT (10X,I2,10X,F10.6)
116      C       DO 40 J = 1,NN
117      C           ZEROS(NN-1+J) = ZERO(J)
118      C           ZEROS(NN-J) = ZERO(J)
119      C 40   CONTINUE
120      C       RETURN
121      C       END
```

DATE
FILME

