```
                DO;
                  BASIC=BASIC+1;
                  SUBSTR(B(BASIC),A(MAX(W,PREV(W)),
                    MIN(W,PREV(W))),1)='1'B;
                  SUBSTR(B(BASIC),A(MAX(Z,W),MIN(Z,W)),
                    1)='1'B;
                  A(MIN(Z,W),MAX(Z,W))=0;
                  J=Z;
                  DO WHILE(J¬=PREV(W));
                    SUBSTR(B(BASIC),A(MAX(PREV(J),J),
                      MIN(PREV(J),J)),1)='1'B;
                    J=PREV(J);
                  END;
                END;
              ELSE
/*  THE EDGE CONNECTING Z AND W SHOULD BE PLACED IN THE
    TREE.                                                       */
                DO;
                  PREV(W)=Z;
                  SUBSTR(T,W,1)='1'B;
                  ALLOCATE STACK;
                  STACK=W;
                  A(MIN(Z,W),MAX(Z,W))=0;
                END;
            END;
        END;
        GO TO NEW_Z;
    END;
END BASIC_GENERATOR;
```

Remark and Certification on Algorithm 446

Ten Subroutines for the Manipulation of Chebyshev Series [C1] [R. Broucke, *Comm. ACM 16* (Apr. 1973), 254–265]

Robert Piessens and Irene Mertens [Recd 11 Jan. 1974] Applied Mathematics and Programming Division, University of Leuven, B-3030, Heverlee (Belgium)

1. Two corrections are needed in the subroutine *CHEBY*:

(i) The statement after statement 50 must be changed into:

LM = MOD(L−1)*(J−1), 2*N) + 1

(ii) formulas (1) and (3) for the computation of Chebyshev series coefficients c_1 do not agree with the exact formulas given by Fox and Parker [1, p. 66]. Indeed the last coefficient must be halved. This can be accomplished in the routine by replacing the five statement before *RETURN* by

```
      DO 100 J = 1, NF
        DO 90 K = 1, NPL
          X(K, J) = FAC*X(K, J)
 90     CONTINUE
        X(NPL, J) = 0.5 DO*X(NPL, J)
100   CONTINUE
```

2. Moreover, the number of cosine-evaluations in *CHEBY* can be reduced by a factor 4 if the *DO*-loop:

```
      DO 30 K = 1, N2
        ⋮
 30   CONTINUE
```

is replaced by

```
      NN = (NPL+1)/2
      DO 30 K = 1, NN
        FK = K − 1
        GC(K) = DCOS(FK*PEN)
        NPLK = NPL+1 − K
        GC(NPLK) = −GC(K)
 30   CONTINUE
      DO 35 K = 1, N
        NPLK = NPL + K
        GC(NPLK) = −GC(K+1)
 35   CONTINUE
```

3. In subroutine *MLTPLY*, the *DO*-loop

```
      DO 10 K = 1, NPL
        ⋮
 10   CONTINUE
```

may be deleted.

We have tested *INVERT* and *BINOM* by calculating

$$[T_0(x) + aT_1(x)]^{-1},$$

and *BINOM*, *XALFA*2 and *XALFA*3 by calculating

$$\left[\left(1+\frac{a^2}{2}\right)T_0(x) + 2aT_1(x) + \frac{a^2}{2}T_2(x)\right]^{-1/2}$$

The results are compared with the exact Chebyshev series expansion

$$(1+ax)^{-1} = \sum_{k=0}^{\infty}{}' a_k T_k(x)$$

where

$$a_k = \frac{2}{(1-a^2)^{\frac{1}{2}}}\left(\frac{(1-a^2)^{\frac{1}{2}}-1}{a}\right)^k, \qquad |a| < 1.$$

The rate of convergence of this series depends strongly on the value of a. For this reason, we have given a the values 0.1, 0.2, . . . , 0.9.

We have noted that, especially in the case of slowly converging series, *INVERT*, *XALFA*2 and *XALFA*3 are more efficient than *BINOM*. Moreover, in order to have convergence, *BINOM* requires more accurate initial approximations than the other routines.

Reference

1. Fox, L., and Parker, I.B. *Chebyshev Polynomials in Numerical Analysis*. Oxford University Press, London, 1968.

Remark on Algorithm 475 [J6]

Visible Surface Plotting Program [Thomas Wright, *Comm. ACM 17* (Mar. 1974), 152–155]

R.G. Mashburn [Recd 9 Dec. 1974] Computer Sciences Division at Oak Ridge National Laboratory Union Carbide Corporation, Nuclear Division* Oak Ridge, TN 37830

* Prime contractor for the U.S. Energy Research and Development Administration.

The Visible Surface Plotting Program, Algorithm 475, has been modified to run on IBM 360 hardware using the Fortran IV (level H) compiler. Using a modifid version of the demonstration program supplied with the algorithm, the two sample plots were successfully produced. The following documents the changes that were required to convert the programs from CDC 6000 or 7000 programs to IBM 360 programs. In addition to the changes listed below it was, of course, necessary to include a *FRAME* subroutine, a *LINE* subroutine, and other calls to plotting subroutines which support locally available plotting equipment. However, since plotting equipment and its software support vary from one installation to another, only those changes pertinent to the IBM 360 are listed here.

Demonstration program:

1. Remove the *PROGRAM* statement.
2. Change the first *DIMENSION* statement from:

   DIMENSION EYE(3), S(4), ST1(80, 80, 2), IS2(3, 160)

   to:

   DIMENSION EYE(3), S(4), ST1(80, 80, 2), IS2(5, 160)

   Note. The comments in the program indicate the first extent *LX* of the array *IS*2 is calculated as follows:

   LX = 1 + NX/NBPW

This is true so long as NX is not an integral multiple of $NBPW$. However, in this case NX is 160 and $NBPW$ (the number of bits per word) is 32 for the IBM 360. Thus NX is an integral multiple of $NBPW$, and LX is calculated simply as $NX/NBPW$ In general use

```
LX = 1 + (NX-1)/NBPW.
```

3. Change the call to the *INIT3D* subroutine to:

```
CALL INIT3D (EYE, 80, 80, 80, ST1, 5, 160, IS2, 9, S)
```

4. Change the two calls to *DANDR* (one after statement 40, the other after statement 110) to:

```
CALL DANDR (80, 80, ST1, 5, 160, 160, IS2, 9, S, IOBJ
*80)
```

5. Change the *DO* statement following the *REWIND* 9 statement from:

```
DO 70 I = 1, 3   to:   DO 70 I = 1, 5
```

INIT3D subroutine: No changes required.
SETORG subroutine:

1. Because no standard exists for referencing arc cosine, the three statements containing references to the arc cosine subroutine were changed from:

```
AL = ACOS(COSAL) to: AL = ARCOS(COSAL)
BE = ACOS(COSBE)     BE = ARCOS(COSBE)
GA = ACOS(COSBA)     GA = ARCOS(COSGA)
```

2. Because no standard exists for *ENTRY* statements and their syntax differs among compilers, it was necessary to change the *ENTRY* statement from:

```
ENTRY PERSPEC   to:
                ENTRY PERSPC(X, Y, Z, XT, YT, ZT)
```

DANDR subroutine:

1. The *DIMENSION* statement should be changed from:

```
DIMENSION MASK (60)   to:   DIMENSION MASK (32)
```

2. The two *DATA* statements following the *DIMENSION* statement should be changed from:

```
DATA NBPW/60/
DATA MASK/1B, 2B, 4B, 10B, 20B, 40B, 100B, 200B, 400B,
1000B,
* 2000B, 4000B, 10000B, 20000B, etc.,
to:
DATA NBPW/32/
DATA MASK/Z1, Z2, Z4, Z8, Z10, Z20, Z40, Z80, Z100,
* Z200, Z400, Z800, Z1000, Z2000, Z4000, Z8000, Z10000,
* Z20000, Z40000, Z80000, Z100000, Z200000, Z400000,
* Z800000, Z1000000, Z2000000, Z4000000, Z8000000
* Z10000000, Z20000000, Z40000000, Z80000000/
```

3. The two uses of the *.AND.* masking operation and the one use of the *.OR.* masking operation were changed to call assembly language function subprograms *IAND* and *IOR* (programs written locally for the ORNL computing center Fortran library) which return an *INTEGER**4 value which is the logical *AND* and logical *OR* respectively of the two arguments given them.
Change the two *.AND.* statements from:

```
IV = IS2(IX, IY).AND.MASK (IBIT)   to:
IV = IAND(IS2(IX, IY), MASK (IBIT))
```

Change the *.OR.* statement from:

```
IS2(X, IY) = IS2(IX, IY).OR.MASK (IBIT)   to:
IS2(IX, IY) = IOR(IS2(IX, IY), MASK(IBIT))
```

Note. In the original program listing of subroutine *DANDR*, the comment card immediately preceding statement 60 reads:
C UPPER-LEFT but should say: C UPPER-RIGHT.

Remark on Algorithm 475[J6]

Visible Surface Plotting Program [Thomas Wright, *Comm. ACM 17* (Mar. 1974), 152–155]

C.J. Doran [Recd 22 Oct. 1974], Physics Department, University of Nottingham, England

Algorithm 475 has been successfully implemented on a D.G. Nova 1220 minicomputer and an I.C.L. 1906A, making substitutions for the nonstandard features of the original algorithm.

ENTRY statements are permitted in 1900 Fortran but not by Data General. *SETORG* and *PERSPC* were therefore written as separate subroutines linked by a labelled common area declared as:

COMMON/CSETORG/JUMP, EX, EY, EZ, AX, AY, AZ, D, R, COSBE, COSAL, COSGA

JUMP being declared as a *LOGICAL* variable. The assigned *GO TO* statement in *PERSPC* then becomes

IF (JUMP) GO TO 30

with *JUMP* = *.FALSE.* replacing the first *ASSIGN* statement in *SETORG*, and *JUMP* = *.TRUE.* replacing the second.

The *DATA* statement in *DANDR* may easily be standardized by writing decimal literals, but most compilers will not accept an integer 2^{NBPW}. *NBPW* should then be redefined as one less than the number of bits per word.

Logical operations between integers may be performed by portable Fortran functions *IAND* and *IOR* as:

```
FUNCTION IAND(I, J)
LOGICAL BI, BJ
EQUIVALENCE (BI, II), (BJ, JJ)
II = I
JJ = J
BI = BI . AND . BJ
IAND = II
RETURN
END
```

with equivalent coding for *IOR*. The first two masking operations then become:
IV = IAND(IS2(IX, IY), MASK(IBIT))

and the third becomes:

IS2(IX, IY) = IOR(IS2(IX, IY), MASK(IBIT))

Remark on Algorithm 478[F4]

Solution of an Overdetermined System of Equations in the l_1 Norm [I. Barrodale and F.D.K. Roberts, *Comm. ACM 17* (June 1974), 319–320]

Fred N. Fritsch and Alan C. Hindmarsh [Recd 23 Sept. 1974], Numerical Mathematics Group, Lawrence Livermore Laboratory, University of California, Livermore, CA 94550

This note is to point out an error in the "Footnote to Algorithm 478." To correspond to the published listing, the statement numbers in (i) of the second paragraph of the footnote should be 210 and 230, rather than 20 and 22. To be consistent with the published statement numbering, we would also recommend that statement number 22 be changed to 220 in the three places it occurs in the replacement coding of (ii).