U.S. Department of the Interior

U.S. Geological Survey

A Method for Designing Film Transfer Functions for Use in an Image Processing Laboratory

by

Joseph S. Duval

Open-File Report 82-705
1982

A METHOD FOR DESIGNING FILM TRANSFER FUNCTIONS FOR USE IN AN IMAGE PROCESSING LABORATORY

BY

JOSEPH S. DUVAL

INTRODUCTION

The U.S. Geological Survey (USGS) image processing laboratory in Denver, Colorado uses several film writing machines which transform digital data to a photographic film format. The transformation process is nonlinear and frequently produces images that are subjectively judged to be unacceptable. In order to provide the user with the means to improve the images, the film writing machines include the capability to utilize user-defined transfer functions. This paper presents an analytical method for designing such transfer functions and contains appendices with listings of computer programs that can be used to define the transfer functions.

FILM CHARACTERISTICS

Before we can define a transfer function to modify the normal operation of the film writing machines, we must first define the unmodified characteristics of the images. This requires some understanding of photographic film properties and the chemical processing of film.

Photographic film normally consists of a film base with one or more emulsion layers made of silver halide crystals suspended in gelatin. Black and white film emulsions are sensitive to most of the visible light spectrum whereas color films normally have three emulsions with sensitivity to selected parts of the spectrum. Exposure to an appropriate light source produces a latent image of the light source in the emulsions and development chemicals are used to make the latent images visible. See Jacobson and others (1978) for details on the development process.

The effects of the film exposure and development can be quantified by measuring film density as a functin of exposure. The film density is defined as the logarithm of the ratio of light incident on the film to light transmitted through the film. Curves of film density versus the logarithm of the exposure are called DlogE curves and are frequently used to define the characteristics of a film. The minimum achievable film density of a film is detemined by the transmission properties of the film base. The minimum density actually achieved is a function of the exposure and the development process. For black and white films the maximum film density increases as the developing time increases and the minimum density also increases to a lesser extent. For color reversal films the maximum film density decreases as the developing time increases and the minimum density also decreases to a lesser extent. Usually the chemical process must be modified to achieve desired results for minimum and maximum film densities with reasonable development times.

The USGS Denver film writing machines transform digital data (0-255 range) to light intensity that is proportional to the exponential of the negative of the data value. The data values are, therefore, proportional to the logarithm of the film exposure and a curve of data values [denoted as density numbers (DN's)] versus film density is a characteristic curve for the image making process. Figure 1 presents a characteristic curve for black and white film. A similar curve for color film is measured using an image produced by overlaying identical red, green, and blue data sets. If the film writing machine is color balanced relative to the film type and chemical process used, the resulting image consists of neutral (grey) tones. The measurement of density is made using a film densitometer equipped with a visual filter. According to Jacobson and others (1978), the visual filter is designed to approximate the spectral response of the human eye. Figure 2 presents a characteristic curve for color film.

TRANSFER FUNCTION DESIGN

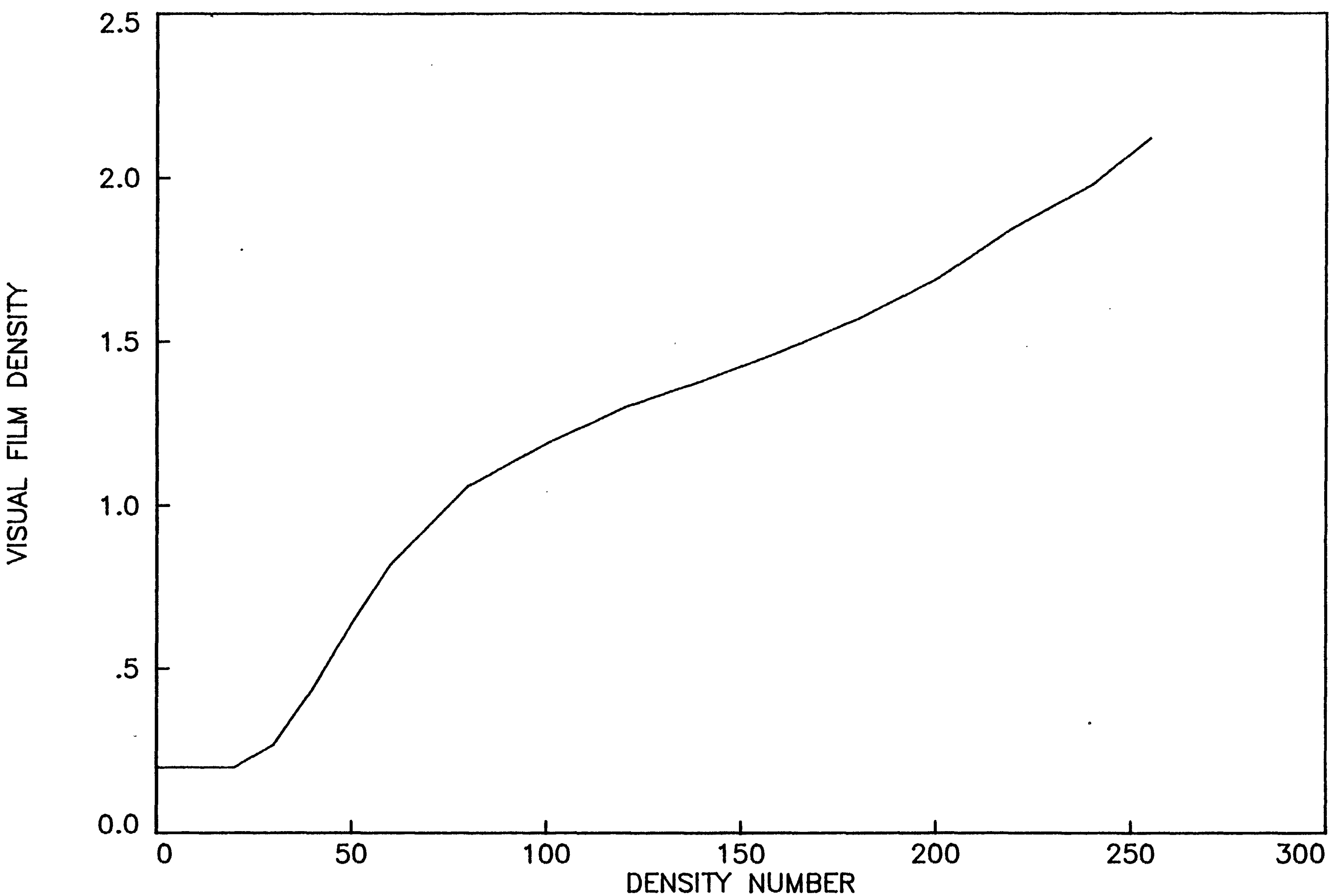
Given the characteristic curve for an image making process, transfer functions can be designed using a two-step process. The first step is to obtain an analytical equation which reproduces the characteristic curve. Because the characteristic curves are smooth and slowly varying as a function of the density number, a polynomial function expressed as powers of the density number can accurately reproduce the curve. A least-squares fitting technique is a convenient way to calculate the polynomial equation and the computer program PWRFIT which is listed in Appendix I performs the necessary calculations. Using PWRFIT, the equation for the curve in Figure 1 is

$$\text{(1)} \quad f(X)=.19-6.46E{-3}*X+5.3E{-4}*X^{2}-5.61E{-6}*X^{3}+2.33E{-8}*X^{4}-3.38E{-11}*X^{5}$$

and the equation for the curve in Figure 2 is

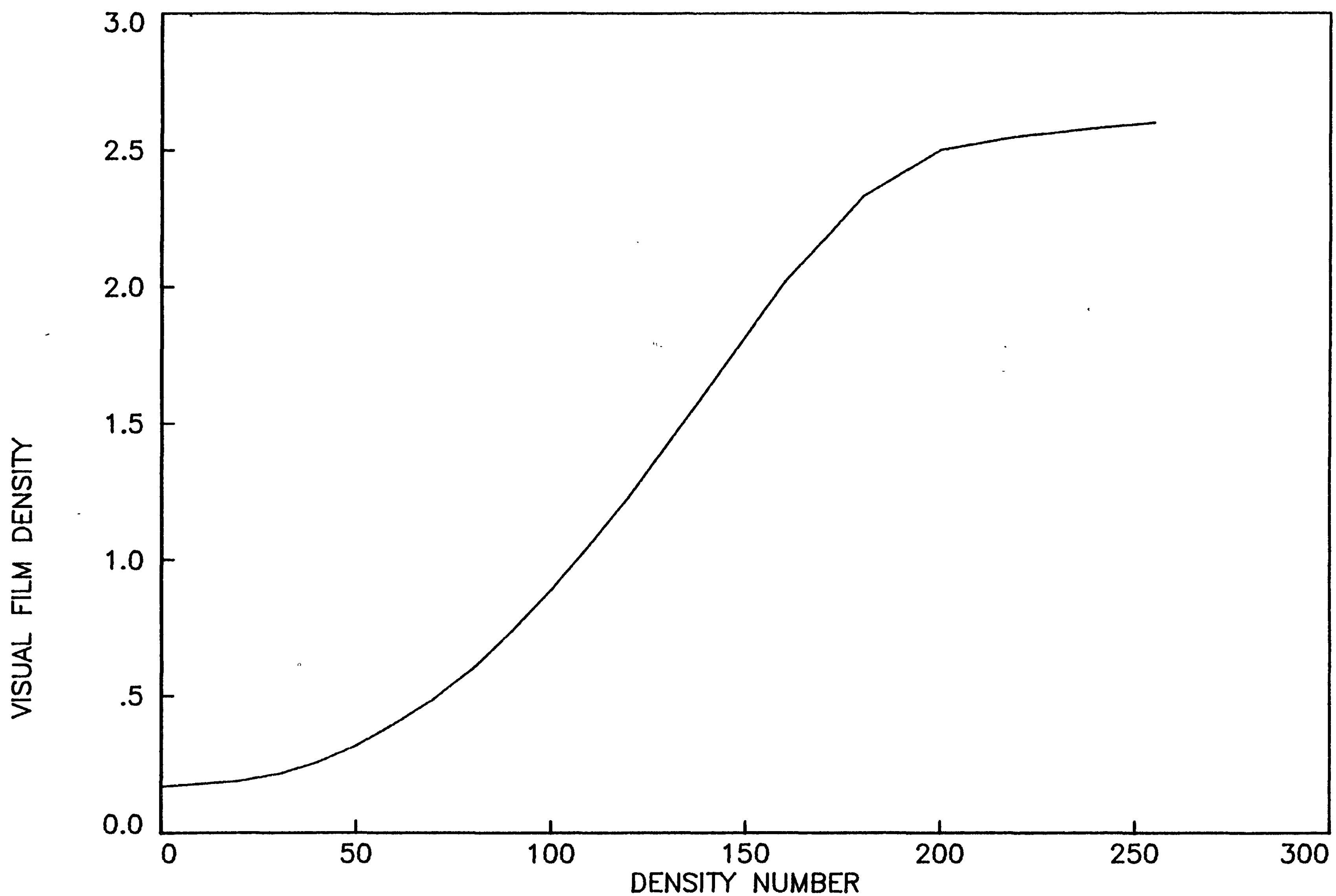
$$\text{(1)} \quad f(X)=.15+6.0E{-3}*X-1.88E{-4}*X^{2}+3.5E{-6}*X^{3}-1.73E{-8}*X^{4}+2.63E{-11}*X^{5}$$

FIGURE 1. CHARACTERISTIC CURVE OF VISUAL FILM DENSITY VERSUS DENSITY NUMBER FOR BLACK AND WHITE IMAGE MAKING PROCESS.



2A

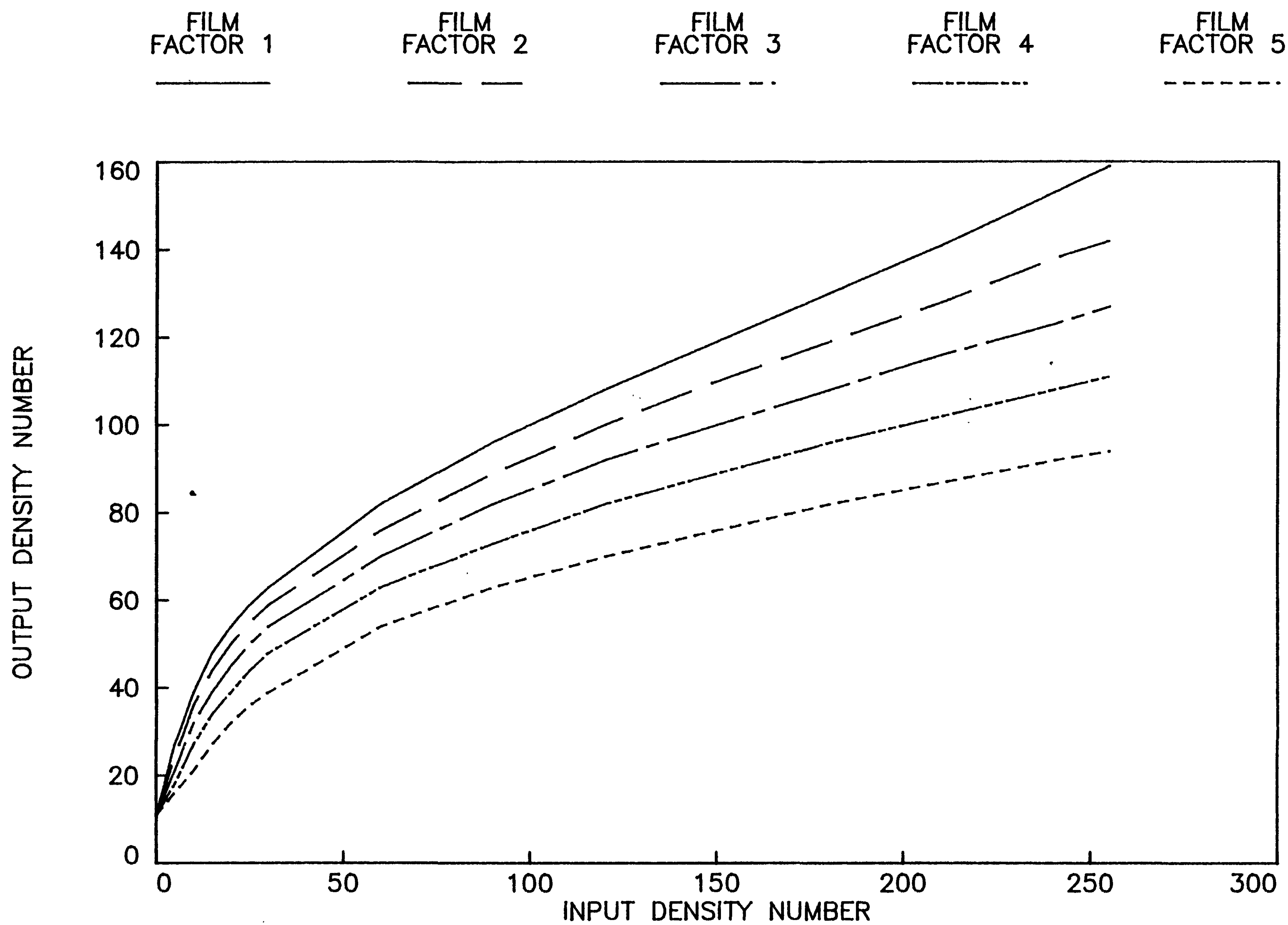
FIGURE 2. CHARACTERISTIC CURVE OF VISUAL FILM DENSITY VERSUS DENSITY NUMBER FOR COLOR IMAGE MAKING PROCESS.



The second step in the definition of a transfer function is to use the analytical equation of the characteristic curve to calculate a curve of density numbers that will produce a desired curve of film densities. TCCALC, which is listed in Appendix II, is a computer program designed to calculate transfer functions that produce linearly varying film densities between specified minimum and maximum values corresponding to specified starting and ending density numbers. Figure 3 presents a series of transfer functions (denoted as film factors) designed for color film using equation (2). Film factor 1 is intended to produce a linear variation of film density in the range 0.2-2.0, film factor 2 in the range 0.2-1.7, film factor 3 in the range 0.2-1.4, film factor 4 in the range 0.2-1.1, and film factor 5 in the range 0.2-0.8. Figure 4 presents characteristic curves produced using the film factors.

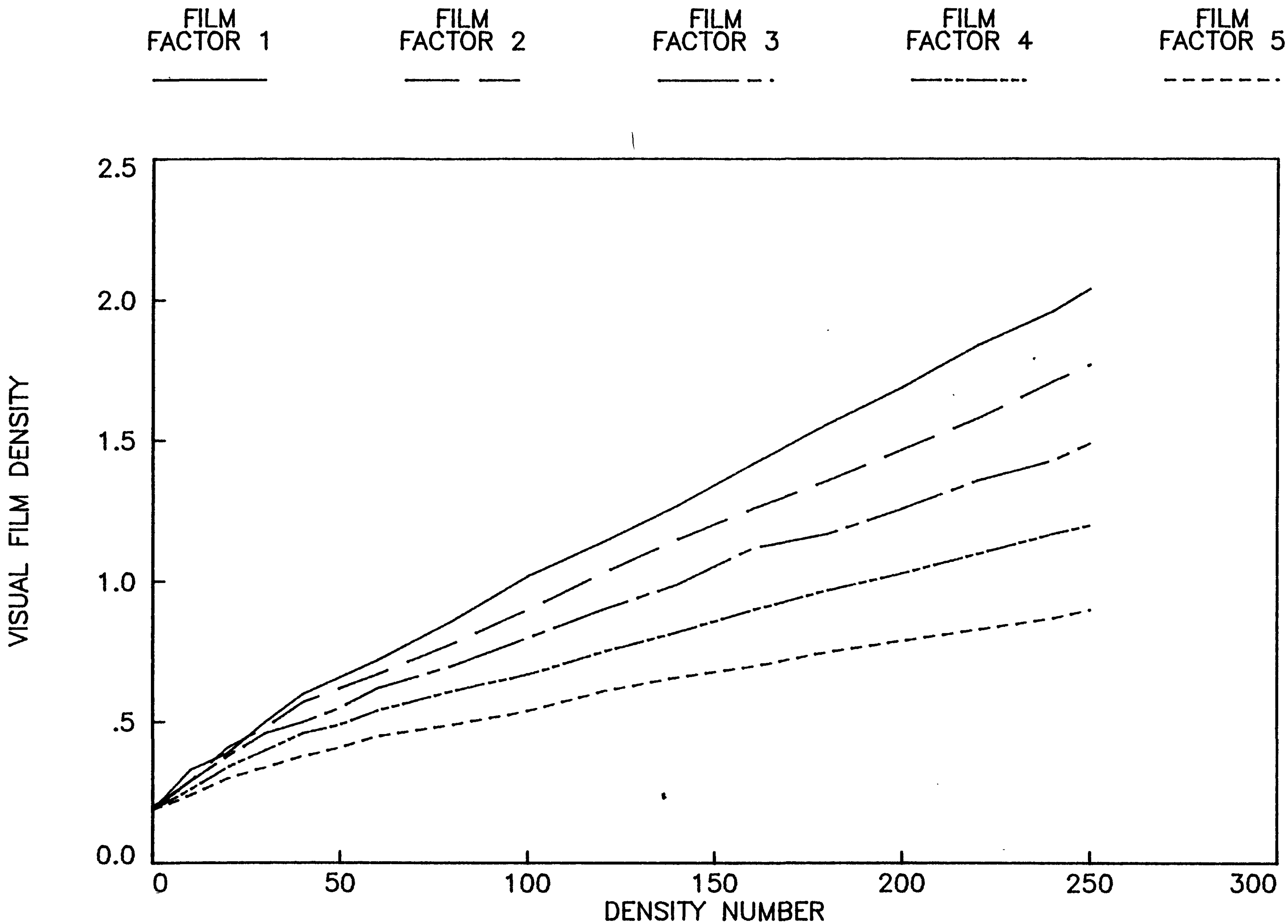
The use of transfer functions that produce a linear relationship between the density numbers and the actual film density is an arbitrary choice based upon a personal prejudice that data transformations should be linear. Experience in the USGS Denver image processing laboratory is that these transfer functions produce images that are judged to be better than images made with no transfer function. Nonlinear transfer functions might be even better because the response of the human eye to changes in film density as measured by a film densitometer is nonlinear (Jacobson and others, 1978). The determination of an appropriate nonlinear function is, however, beyond the scope of this paper.

FIGURE 3. TRANSFER FUNCTIONS WHICH MODIFY THE NORMAL COLOR IMAGE MAKING PROCESS BY MAPPING INPUT DATA TO AN OUTPUT DENSITY NUMBER CURVE.



3A

FIGURE 4. CHARACTERISTIC CURVES OF VISUAL FILM DENSITY VERSUS DENSITY NUMBER MEASURED ON IMAGES PRODUCED USING THE FILM FACTORS OF FIGURE 3.



REFERENCES

Jacobson, R.E., Ray, S.F., Attridge, G.G., and Axford, N.R., 1978, The manual of photography: London, Focal Press, 628 p.

APPENDIX I

The program PWRFIT which is listed below is a routine that uses an ordinary least-squares fitting technique to calculate a polynomial power series equation for an arbitrary data set. The program is written in standard FORTRAN. Input/output statements are site dependent.

```
C************************** PWRFIT ****************************
C***
C*** THIS PROGRAM CALCULATES A POWER SERIES FIT TO A SET OF
C*** DATA ASSUMING THE FORM:
C***
C***    F(X) = A0 + A1*X + A2*X**2 + ... + AN*X**N
C***
C*** INPUT DATA REQUIRED ARE:
C***
C*** NPTS = THE NUMBER OF DATA POINTS TO BE USED IN THE
C***        CALCULATION
C***
C*** NT = THE NUMBER OF TERMS IN THE POWER SERIES
C***
C*** ILBL = AN 80 CHARACTER LABEL
C***
C*** YOU CAN USE AS MANY AS 100 DATA POINTS WITH AS MANY AS
C*** 20 TERMS IN THE SERIES.
C***
C*************************************************************
C
      COMMON XS(100),YS(100),WS(100),ILBL(40)
      DOUBLE PRECISION H(20,20),B(20),SOLN(20)
      DIMENSION X(100),Y(100),W(100)
      DIMENSION ERR(100),EC(100),COF(100,20)
      OUTPUT ' ',' ','THIS PROGRAM CALCULATES A LEAST SQUARES FIT'
      OUTPUT 'TO A POWER SERIES IN X.  YOU ARE RESTRICTED TO'
      OUTPUT 'MAXIMUMS OF 100 DATA POINTS AND 20 TERMS IN THE'
      OUTPUT 'POWER SERIES'
      OUTPUT 'NEVERLESS, MORE THAN 10 TERMS ARE NOT RECOMMENDED'
      OUTPUT 'BECAUSE OF PROBLEMS WITH ACCURACY'
100   OUTPUT ' ',' ','HOW MANY DATA POINTS TO BE ENTERED ?'
      INPUT NPTS
      OUTPUT ' ',' ','HOW MANY TERMS IN THE POWER SERIES ?'
      INPUT NT
      OUTPUT ' ',' ','ENTER A ONE LINE LABEL.'
      READ(5,1) ILBL
1     FORMAT(40A2)
      IF(NPTS.EQ.0) GOTO 1000
200   CALL PWRDVR(H,B,SOLN,X,Y,W,ERR,EC,COF,NPTS,NT,NREP)
      IF(NREP.EQ.0) GOTO 1000
      IF(NREP.EQ.2) GOTO 200
```

```
      GOTO 100
1000  OUTPUT ' ',' ','PROGRAM FINISHED'
      STOP
      END
C********************** PWRDVR ********************************
C***
C*** THIS SUBROUTINE IS THE DRIVER ROUTINE USED WITH PWRFIT.
C***
C*** PRIMARY FUNCTION IS I/O.
C***
C***************************************************************
C
      SUBROUTINE PWRDVR(H,B,SOLN,X,Y,W,ERR,EC,COF,NPTS,NT,NREP)
      COMMON XS(100),YS(100),WS(100),ILBL(40)
      DOUBLE PRECISION H(NT,NT),B(NT),SOLN(NT)
      DIMENSION X(NPTS),Y(NPTS),W(NPTS)
      DIMENSION ERR(NPTS),EC(NPTS),COF(NPTS,NT)
      IF(NREP.NE.2) GOTO 100
      DO 90 I=1,NPTS
      X(I)=XS(I)
      Y(I)=YS(I)
90    W(I)=WS(I)
      GOTO 200
100   OUTPUT ' ',' ','NOW YOU ARE ASKED TO ENTER THE X-VALUE,'
      OUTPUT 'Y-VALUE, AND THE WEIGHT OR ERROR ASSOCIATED WITH Y.'
      DO 110 I=1,NPTS
      OUTPUT ' ',' ENTER VALUES OF X,Y, AND WT FOR DATA POINT # ',I
      INPUT X(I),Y(I),W(I)
      XS(I)=X(I)
      YS(I)=Y(I)
110   WS(I)=W(I)
200   WRITE(6,1) ILBL
1     FORMAT(/////,40A2)
      OUTPUT ' ','VALUES ENTERED FOR X,Y, AND WT ARE:'
      DO 210 I=1,NPTS
210   OUTPUT I,X(I),Y(I),W(I)
220   OUTPUT ' ','WOULD YOU LIKE TO MAKE ANY CHANGES ?'
      READ(5,2) IANS
2     FORMAT(A1)
      IF(IANS.NE.'Y'.AND.IANS.NE.'N') GO TO 220
      IF (IANS.EQ.'N') GOTO 300
230   OUTPUT ' ','WHICH DATA POINT NEEDS TO BE CHANGED ?'
      INPUT IW
      IF(IW.GE.1.AND.IW.LE.NPTS) GOTO 240
      OUTPUT 'BAD NUMBER -- TRY AGAIN'
      GOTO 230
240   OUTPUT 'WHAT NEEDS TO BE FIXED - X,Y,WT, OR ALL ? (X/Y/W/A)'
      READ(5,2) IANS
      IF(IANS.EQ.'X'.OR.IANS.EQ.'A') GOTO 250
      IF(IANS.EQ.'Y') GOTO 260
      IF(IANS.EQ.'W') GOTO 270
      OUTPUT 'BAD ANSWER -- TRY AGAIN'
```

```
      GOTO 240
250   WRITE(6,3) IW
3     FORMAT('NEW VALUE FOR X(',I3,') ?')
      INPUT X(IW)
      XS(IW)=X(IW)
      IF(IANS.NE.'A') GOTO 271
260   WRITE(6,4) IW
4     FORMAT('NEW VALUE FOR Y(',I3,') ?')
      INPUT Y(IW)
      YS(IW)=Y(IW)
      IF(IANS.NE.'A') GOTO 271
270   WRITE(6,5) IW
5     FORMAT('NEW VALUE FOR WT(',I3,') ?')
      INPUT W(IW)
      WS(IW)=W(IW)
271   OUTPUT ' ','WOULD YOU LIKE TO MAKE ANY OTHER CHANGES ?'
      READ(5,2) IANS
      IF(IANS.EQ.'Y') GOTO 230
      IF(IANS.EQ.'N') GOTO 200
      OUTPUT ' ','ANSWER YES OR NO ! (Y/N)'
      GOTO 271
C***
CC*** CALCULATE THE COEFFICIENTS OF THE POWER SERIES
C***
300   DO 310 I=1,NPTS
      COF(I,1)=1.
      XPWR=1.
      DO 310 J=2,NT
      COF(I,J)=XPWR*X(I)
310   XPWR=XPWR*X(I)
C***
C*** CALCULATE THE MATRIX H (SQUARE MATRIX OF COEFFICIENTS)
C***
      DO 320 I=1,NT
      DO 320 J=1,NT
      H(I,J)=0.
      DO 320 K=1,NPTS
320   H(I,J)=H(I,J)+COF(K,I)*COF(K,J)/W(K)**2
C***
C*** CALCULATE THE MATRIX B (VECTOR MATRIX OF KNOWNS)
C***
      DO 330 I=1,NT
      B(I)=0.
      DO 330 J=1,NPTS
330   B(I)=B(I)+Y(J)*COF(J,I)/W(J)**2
C***
C*** SOLVE THE EQUATION
C***
      DO 335 I=1,NT
335   SOLN(I)=B(I)
      CALL MATINV(H,NT,SOLN,NT,1,DET)
X     OUTPUT ' DETERMINANT = ',DET
```

```
      IF(ABS(DET).GT.1E-30) GOTO 340
      OUTPUT ' ',' ','SYSTEM HAS NO SOLUTION'
      GOTO 350
340   CHISQ=0.
      DO 342 I=1,NPTS
      EC(I)=0.
      DO 341 J=1,NT
341   EC(I)=EC(I)+COF(I,J)*SOLN(J)
      ERR(I)=Y(I)-EC(I)
342   CHISQ=CHISQ+ERR(I)**2
      WRITE(6,1) ILBL
      OUTPUT ' ','HERE ARE THE COEFFICIENTS OF THE POWER SERIES'
      DO 343 I=1,NT
343   WRITE(6,6) I,SOLN(I)
6     FORMAT('COEFFICIENT # ',I2,' = ',1PE12.5)
      OUTPUT ' ','HERE ARE VALUES CALCULATED AND ERRORS:'
      DO 344 I=1,NPTS
344   WRITE(6,7)  I,Y(I),I,EC(I),I,ERR(I)
7     FORMAT(' Y(',I2,') = ',1PE12.5,' YCALC(',I2,') = ',
     &1PE12.5,' ERROR(',I2,') = ',1PE12.5)
350   OUTPUT ' ','DO YOU WANT TO RUN THE PROGRAM AGAIN ? (Y/N)'
      READ(5,2) IANS
      IF(IANS.EQ.'Y') GOTO 400
      IF(IANS.EQ.'N') GOTO 450
      OUTPUT 'BAD ANSWER -- TRY AGAIN'
      GOTO 350
400   OUTPUT ' ','DO YOU WANT TO MODIFY THE PREVIOUS DATA OR'
      OUTPUT 'ENTER NEW DATA ? (P/N)'
      READ(5,2) IANS
      IF(IANS.EQ.'P') GOTO 410
      IF (IANS.EQ.'N') GOTO 420
      OUTPUT ' BAD ANSWER -- TRY AGAIN'
      GOTO 400
410   OUTPUT ' ',' NUMBER OF TERMS IN THE SERIES ?'
      INPUT NT
      NREP=2
      RETURN
420   NREP=1
      RETURN
450   NREP=0
       RETURN
      END
C**************************** MATINV ****************************
C***
C*** THIS SUBROUTINE SOLVES THE MATRIX EQUATION A X = B WHERE
C***' A' IS A SQUARE MATRIX OF COEFFICIENTS AND 'B' IS A
C*** MATRIX OF CONSTANT VECTORS.
C***
C*** A = ON INPUT 'A' IS A SQUARE MATRIX OF COEFFICIENTS
C***     ON OUTPUT 'A' IS THE INVERSE OF THE ORIGINAL MATRIX
C***
C*** B = ON INPUT, 'B' IS THE CONSTANT VECTOR MATRIX
```

```
C***      ON OUTPUT, 'B' IS THE SOLUTION MATRIX
C***
C*** N = NUMBER OF ROWS IN 'A'
C***
C*** NX = THE MAXIMUM NUMBER OF ROWS OR COLUMNS DIMENSIONED FOR
C***      'A' IN THE CALLING PROGRAM
C***
C*** M = NUMBER OF COLUMNS IN 'B'
C***     IF M=0, THEN ONLY THE INVERSE OF 'A' IS COMPUTED AND NO
C***     CONSTANT VECTORS ARE ASSUMED
C***
C*** DET = DETERMINANT OF 'A'
C***
C*****************************************************************
C
      SUBROUTINE MATINV(A,N,B,NX,M,DET)
      DOUBLE PRECISION A(1),B(1),PIVOT(20),SWAP,T,AMAX
      DIMENSION IX1(20),IX2(20) ,IPIV(20)
C***
C***  INITIALIZATION
C***
10    DET=1.
      IFLAG=0
      DO 20 J=1,N
20    IPIV(J)=0
X     OUTPUT 'N, NX, M ARE:',N,NX,M
X     DO 500 I=1,N
X     DO 500 J=1,N
X     L=I+(J-1)*NX
X500  WRITE(6,501) I,J,A(L)
X501  FORMAT(' ROW ',I2,' COLUMN ',I2,' ELEMENT = ',E12.5)
      DO 210 I=1,N
C***
C*** SEARCH FOR PIVOT ELEMENT
C***
C*** NOTE THAT COLUMNWISE STORAGE IS ASSUMED
C***
      AMAX=0.
      DO 70 J=1,N
      IF(IPIV(J)-1) 30,70,30
30    DO 60 K=1,N
      IF(IPIV(K)-1) 40,60,250
40    LL=J+(K-1)*NX
      IF(DABS(AMAX)-DABS(A(LL))) 50,60,60
50    IROW=J
      ICOL=K
      AMAX=A(LL)
60    CONTINUE
70    CONTINUE
      IPIV(ICOL)=IPIV(ICOL)+1
C***
C*** INTERCHAGE ROWS TO PUT ELEMENT ON DIAGONAL
```

```
C***
      IF(IROW-ICOL) 80,120,80
80    DET=-DET
      DO 90 L=1,N
      NROW=IROW+(L-1)*NX
      NCOL=ICOL+(L-1)*NX
      SWAP=A(NROW)
      A(NROW)=A(NCOL)
90    A(NCOL)=SWAP
      IF(M) 120,120,100
100   DO 110 L=1,M
      NROW=IROW+(L-1)*NX
      NCOL=ICOL+(L-1)*NX
      SWAP=B(NROW)
      B(NROW)=B(NCOL)
110   B(NCOL)=SWAP
120   IX1(I)=IROW
      IX2(I)=ICOL
      LL=ICOL+(ICOL-1)*NX
      PIVOT(I)=A(LL)
      IF(ALOG10(ABS(DET))+DLOG10(DABS(PIVOT(I))).LE.30.) GOTO 121
      IFLAG=1
      GOTO 122
121   DET=DET*PIVOT(I)
122   CONTINUE
C***
C*** DIVIDE PIVOT ROW BY PIVOT ELEMENT
C***
      A(LL)=1.
      DO 130 L=1,N
      NCOL=ICOL+(L-1)*NX
130   A(NCOL)=A(NCOL)/PIVOT(I)
      IF(M) 160,160,140
140   DO 150 L=1,M
      NCOL=ICOL+(L-1)*M
150   B(NCOL)=B(NCOL)/PIVOT(I)
C***
C*** REDUCE NON-PIVOT ROWS
C***
160   JCON=(ICOL-1)*NX
      DO 210 L1=1,N
      IF(L1-ICOL) 170,210,170
170   LL=JCON+L1
      T=A(LL)
      A(LL)=0.
      DO 180 L=1,N
      NL1=L1+(L-1)*NX
      NCOL=ICOL+(L-1)*NX
180   A(NL1)=A(NL1)-A(NCOL)*T
      IF(M) 210,210,190
190   DO 200 L=1,M
      NL1=L1+(L-1)*NX
```

```
      NCOL=ICOL+(L-1)*NX
200   B(NL1)=B(NL1)-B(NCOL)*T
210   CONTINUE
C***
C***
C*** INTERCHANGE COLUMNS
C***
      DO 240 I=1,N
      L=N+1-I
      IF(IX1(L)-IX2(L)) 220,240,220
220   JROW=IX1(L)
      JCOL=IX2(L)
      DO 230 K=1,N
      NROW=K+(JROW-1)*NX
      NCOL=K+(JCOL-1)*NX
      SWAP=A(NROW)
      A(NROW)=A(NCOL)
230   A(NCOL)=SWAP
240   CONTINUE
X      OUTPUT ' ',' INVERSE MATRIX:'
X      DO 510 I=1,N
X      DO 510 J=1,N
X      L=I+(J-1)*NX
X510   WRITE(6,501) I,J,A(L)
250   IF(IFLAG.NE.0) OUTPUT 'DETERMINANT EXCEEDS OVERFLOW LIMIT'
260   RETURN
      END
```

APPENDIX II

The program TCCALC which is listed below is a routine that uses a polynomial equation representing a film characteristic curve to calculate a transfer function to modify the image making process. All of the program and subroutines except PDEL are written in standard FORTRAN, and PDEL is a specialized subroutine written in assembler to punch a delete code on a paper tape punch. The output procedures provide for outputing the transfer function either to a floppy disk or a paper tape punch. These procedures are site specific.

```
C************************ TCCALC ********************************
C***
C***  TRANSFER CHARACTERISTIC CALCULATION PROGRAM
C***
C*** THIS PROGRAM IS USED TO CALCULATE TRANSFER CHARACTERISTICS
C*** GIVEN THE FOLLOWING DATA:
C***
C*** FD1,DN1,FD2,DN2 - WHERE FD1, DN1 ARE THE FILM DENSITY AND DN
C*** NUMBER FOR THE START OF A LINEAR FUNCTION THAT GOES TO THE
C*** VALUE FD2 AT DN2.  THIS LINEAR FUNCTION IS USED TO CALCULATE
C*** THE DESIRED FILM DENSITY AS A FUNCTION OF THE DENSITY NUMBER
C*** IF DN1 IS NOT EQUAL TO 0 AND/OR DN2 IS NOT EQUAL TO 255,
C*** THE FILM DENSITIES FOR VALUES BELOW DN1 WILL BE SET TO FD1
C*** AND/OR THE VALUES ABOVE DN2 WILL BE SET TO FD2.
C***
C*** A(I), I=1,6 - THE COEFFICIENTS A(I) ARE THE COEFFICIENTS
C***  DETERMINED BY A SIX TERM POWER SERIES FIT TO THE GREY
C***  FILM DENSITY VERSUS THE INPUT DN.  THE POWER SERIES FIT
C***  IS DONE USING 'PWRFIT'.
C***
C*** METHOD:  THE FUNCTION DEFINED BY THE POWER SERIES IS USED TO
C***       CALCULATE THE VALUE OF DN WHICH GIVES THE DESIRED FILM
C***       DENSITY.  THIS IS DONE USING A SIMPLE ROOT FINDING
C***       PROGRAM CALLED 'BISECT'.
C***
C**************************************************************
C
      COMMON A,B,MAXBI,TOL,DELTAX,FD(256),LBL(40),COEF(6)
      DIMENSION ROOT(10),F(10),ERR(10)
      OUTPUT ' ',' ENTER A ONE LINE TITLE.'
      READ(5,1) LBL
1     FORMAT(40A2)
      OUTPUT ' ',' ENTER FD1, DN1, FD2, AND DN2.'
      INPUT FD1,DN1,FD2,DN2
C***
C*** CALCULATE THE PARAMETERS OF THE LINEAR EQUATION
C***
      A2=(FD1-FD2)/(DN1-DN2)
```

```
      A1=FD1-A2*DN1
      DN=0
      EXPFAC=0.
C***
C*** EXPFAC CAN BE USED TO CREATE EXPOSURE FACTORS WHERE THE
C*** MAXIMUM FILM DENSITY IS REDUCED AND THE BOTTOM END OF THE
C*** DN SCALE ARE ALL SET TO THE SAME VALUE.
C***
C*** FOR THIS PROGRAM AS IS THE EXPFAC IS SET TO ZERO.
C***
      DO 100 I=1,256
      FD(I)=A1+A2*DN-EXPFAC*ABS(FD2-FD1)
      IF(FD(I).LT.AMIN1(FD1,FD2)) FD(I)=AMIN1(FD1,FD2)
      IF(FD(I).GT.AMAX1(FD1,FD2)) FD(I)=AMAX1(FD1,FD2)
      IF(DN.LT.DN1) FD(I)=FD1
      IF(DN.GT.DN2) FD(I)=FD2
100   DN=DN+1
      MAXBI=1
      A=0
      B=255
      DELTAX=0.5
      TOL=AMAX1(FD1,FD2)/100.
      OUTPUT ' ',' ENTER THE 6 COEFFICIENTS OF THE POWER SERIES.'
      INPUT COEF
      NROOTS=1
C***
C*** CALL BISEC1 TO DYNAMICALLY SET UP THE ARRAYS
C***
X     OUTPUT ' ','CALLING BISEC1: A&B AND DELTAX = ',A,B,DELTAX
      CALL BISEC1(ROOT,F,ERR,NROOTS)
      STOP
      END
C************************** BISEC1 ****************************
C***
C*** BISEC1 IS USED TO DYNAMICALLY SET UP ARRAYS
C***      AND TO PRINT RESULTS
C***
C*************************************************************
      SUBROUTINE BISEC1(ROOT,F,ERR,NROOTS)
      COMMON AA,BB,MAXBI,TOL,DELTAX,FD(256),LBL(40),COEF(6)
      DIMENSION ROOT(NROOTS),F(NROOTS),ERR(NROOTS),IODN(256),INDN(256)
C***
C*** ALL PARAMETERS HAVE BEEN ESTABLISHED.  CALL BISECT.
C***
      DO 100 J=1,256
      JFLAG=0
90    C=FD(J)
      A=AA
      IF(J.GT.1.AND.ROOT(1).LT.9.9E29) A=AINT(ROOT(1))-2
      IF(A.LT.0.) A=0.
      B=BB
X     OUTPUT ' ','BISEC1 CALLING BISECT: AA&BB ARE',AA,BB
```

```
X       OUTPUT ' A&B ARE:',A,B
        CALL BISECT(A,B,MAXBI,TOL,DELTAX,ROOT,F,ERR,NROOTS,C,COEF)
        IF(ROOT(1).LT.9.9E29) GOTO 95
        JFLAG=JFLAG+1
        IF(JFLAG.EQ.1) GOTO 90
        OUTPUT ' ','NO ROOT FOR J = ',J,' AND FD(J) = ',FD(J)
        OUTPUT 'ROOT(1) = ',ROOT(1),' F(1) = ',F(1)
X        OUTPUT ' COEF = ',COEF,' C = ',C,' A&B = ',A,B
        GOTO 100
C***
C*** IF COMPILED WITH XON, PRINT OUT THE ROOT, FUNCTION VALUE,
C*** AND ERROR.
C***
C*** NOTE THAT 9.99999E29 IS STORED AS THE ROOT IF NONE IS FOUND.
C***
95      CONTINUE
X       OUTPUT ' ','      ROOT   FUNCTION VALUE         ACCURACY'
X       DO 99 I=1,NROOTS
X       WRITE(6,98) ROOT(I),F(I),ERR(I)
X98     FORMAT(3X,F7.2,4X,F7.2,10X,F7.2)
X99     CONTINUE
        IODN(J)=AINT(ROOT(1))
100     INDN(J)=J-1
        WRITE(6,1) LBL
1       FORMAT(/////,1X,40A2)
        OUTPUT ' ','OUTPUT DN SEQUENCE:'
        I1=1
        I2=15
        DO 110 I=1,256,15
        I3=I1-1
        WRITE(6,2) I3,(IODN(J),J=I1,I2)
2       FORMAT(2X,I3,3X,15(1X,I3))
        I1=I1+15
        I2=I2+15
110     IF(I2.GT.256) I2=256
        WRITE(6,3)
3       FORMAT(//////////)
        OUTPUT ' ','TC TO DISK/PAPER TAPE ? (D/T)'
        READ(5,4) IDT
4       FORMAT(A1)
        IF(IDT.EQ.'D') GOTO 200
        OUTPUT ' ','READY PUNCH, TYPE <RETURN>'
        INPUT IANS
        GOTO 201
200     OUTPUT ' ','ASSIGN UNIT 7 TO DIRECT TC FILE'
        OUTPUT ' ','ASSIGN UNIT 8  TO INVERSE TC FILE, THEN CONTINUE'
        PAUSE
201     CONTINUE
        IF(IDT.EQ.'T') CALL  PUNCH(IODN)
        IF(IDT.EQ.'D') CALL DWRITE(IODN,7)
        I1=256
        DO 210 I=1,256
```

```
      INDN(I1)=IODN(I)
210   I1=I1-1
      IF(IDT.EQ.'T') CALL PUNCH(INDN)
      IF(IDT.EQ.'D') CALL DWRITE(INDN,8)
      RETURN
      END
C************************** BISECT ******************************
C***
C*** ROOTFINDER - BISECTION METHOD
C***
C*****************************************************************
C
      SUBROUTINE BISECT(A,B,MAXBI,TOL,DELTAX,ROOT,F,ERR,NROOTS,CF,COEF)
      DIMENSION ROOT(1),F(1),ERR(1),COEF(1)
      REAL LEFT
      INTEGER C
50    IBAD=1
X     OUTPUT ' ','ENTRY INTO BISECT: A&B&DELTAX ARE ',A,B,DELTAX
      IF(A.GE.B.OR.MAXBI.LE.0.OR.TOL.LE.0) IBAD=0
      IF(DELTAX.LE.0.OR.NROOTS.LE.0) IBAD=0
      IF(IBAD.NE.0) GOTO 100
C***
C*** USER MAY CORRECT BAD DATA AND CONTINUE
C***
      OUTPUT ' ','BAD DATA IN SUBROUTINE BISECT'
      OUTPUT ' A = ',A,' B = ',B,' MAXBI = ',MAXBI
      OUTPUT ' TOL = ',TOL,' DELTAX = ',DELTAX,' NROOTS = ',NROOTS
      OUTPUT ' ','ENTER A NEW SET OF VALUES'
      INPUT A,B,MAXBI,TOL,DELTAX,NROOTS
      GOTO 50
C***
C*** BEGIN SUBPROGRAM
C***
C*** INITIALIZE LOCAL VARIABLES AND SET ALL ROOTS TO 9.99999E29
C***
100   N=0
      NOROOT=NROOTS
      DO 200 I=1,NROOTS
      ROOT(I)=9.99999E29
      F(I)=ROOT(I)
200   ERR(I)=F(I)
C***
C*** FUNCTIONAL VALUE OF LEFT BOUND
C***
210   X=A
C***
C*** IF DESIRED NUMBER OF ROOTS HAVE BEEN FOUND, RETURN
C***
      IF(N.GE.NOROOT) RETURN
C***
C*** SEARCH FOR NEW ROOT
C***
```

```
      N=N+1
      Y=FNF(X,CF)
250   FF=Y
C***
C*** ADVANCE TO NEXT SEARCH INTERVAL
C***
      A=A+DELTAX
C***
C*** IF GREATER THAN UPPER BOUND , RETURN
C***
      IF(A.GT.B) RETURN
      X=A
      Y=FNF(X,CF)
      PROD=FF*Y
C***
C*** IF PRODUCT IS POSITIVE, SEARCH NEXT INTERVAL
C*** IF PRODUCT IS NEGATIVE, LOOK FOR ROOT.
C***
      IF(PROD.GT.0) GOTO 250
      IF(PROD.LT.0) GOTO 300
      IF(FF.NE.0.) GOTO 260
C***
C*** EXACT ROOT HAS BEEN FOUND
C***
      X=A-DELTAX
      Y=FF
260   ROOT(N)=X
      F(N)=Y
      A=A+DELTAX
      SIZE=1.E-12
      ERR(N)=SIZE
C***
C*** SEARCH NEXT INTERVAL FOR REMAINING ROOTS
C***
      GOTO 210
C***
C*** ROOT HAS BEEN BRACKETED.  COMPUTE MIDPOINT AND ITS
C*** FUNCTIONAL VALUE.
C***
300   LEFT=A-DELTAX
      RIGHT=A
C***
C*** 'C' IS NUMBER OF ITERATIONS.
C***
      C=0
310   X=(LEFT+RIGHT)/2.
      Y=FNF(X,CF)
      C=C+1
X     OUTPUT ' ','AT 300: X,Y,CF,C ARE',X,Y,CF,C
C***
C*** CHECK FOR MAXIMUM NUMBER OF INTERATIONS
C*** AND PRINT MESSAGE IF EXCEEDED
```

```
C***
      IF(C.GT.MAXBI) GOTO 340
C***
C*** CHECK WHETHER ERROR TOLERANCE IS SATISFIED.
C***
      IF(ABS(Y).LT.TOL ) GOTO 330
C***
C*** IF PRODUCT IS POSITIVE LOOK ON RIGHT INTERVAL
C***
      PROD=FF*Y
      IF(PROD.LE.0.) GOTO 320
      LEFT=X
      GOTO 310
C***
C*** CHECK IF ROOT HAS BEEN FOUND
C***
320   IF(PROD.EQ.0.) GOTO 330
C***
C*** SEARCH ON LEFT INTERVAL
C***
      RIGHT=X
      GOTO 310
C***
C*** ROOT HAS BEEN FOUND
C***
330   ROOT(N)=X
      F(N)=Y
      SIZE=RIGHT-LEFT
      ERR(N)=SIZE
      GOTO 210
C***
C*** MAX # OF ITERATIONS EXCEEDED, PRINT WARNING MESSAGE.
C*** PRINT APPROXIMATE ROOT AND ACCURACY
C***
340   OUTPUT ' ','MAX # BISECTIONS REACHED ON ROOT #',N
      OUTPUT' X BETWEEN ',LEFT,' AND ',RIGHT
      OUTPUT ' F(X) = ',Y
      SIZE=RIGHT-LEFT
      OUTPUT ' ACCURACY TO ',SIZE
      OUTPUT 'AVERAGE VALUE STORED AS APPROXIMATION'
      ROOT(N)=(LEFT+RIGHT)/2.
      F(N)=Y
      ERR(N)=SIZE
C***
C*** SEARCH NEXT INTERVAL FOR REMAINING ROOTS
C***
      GOTO 210
      END
C******************** END ************************************
C***
C*** FUNCTION SUBPROGRAM TO CALCULATE DESIRED FILM DENSITIES
C***
```

```
C*** USAGE: X=FNF(X,C) , WHERE 'X' IS THE DN VALUE AT WHICH
C***        THE FUNCTION IS TO BE EVALUATED AND 'C' IS THE
C***        REFERENCE VALUE.
C***
C***********************************************************
C
      FUNCTION FNF(X,C)
      COMMON A,B,MAXBI,TOL,DX,FD(256),LBL(40),COEF(6)
C***
C*** COEF ARE THE COEFFICIENTS OF THE EQUATION
C***
      FNF=-C
      XPWR=1.
      DO 100 I=1,6
      FNF=FNF+COEF(I)*XPWR
100   XPWR=XPWR*X
      RETURN
      END
C*********************** PACK *******************************
C***
C*** THIS SUBROUTINE PACKS INTEGER WORDS WITH VALUES IN THE
C*** RANGE 0-255 INTO SINGLE WORDS.
C***
C*** USAGE: CALL PACK(IBUF,N) - WHERE IBUF IS THE INPUT AND OUTPUT
C*** ARRAY AND 'N' IS THE NUMBER OF NUMBERS TO BE PACKED
C***
C***********************************************************
C
      SUBROUTINE PACK(IBUF,N)
      INTEGER IBUF(1),PADR,UPADR
C
      ASSEMBLER
C SET WORD ADDR
      LDA IBUF
      STA UPADR
C SET BYTE ADDR
      LLA 1
      STA PADR
C
      FORTRAN
      DO 1 I=1,N
      ASSEMBLER
C LOAD WORD
      LDX UPADR
      LDA @0
C INCR WORD ADDR
      DATA :128
      STX UPADR
C STORE BYTE
      LDX PADR
      SBM
      STA @0
```

```
      SWM
C INCR BYTE COUNTER
      DATA :128
      STX PADR
      FORTRAN
C
1     CONTINUE
      RETURN
      END
C********************** DWRITE ********************************
C***
C*** DISK WRITING SUBROUTINE TO WRITE TRANSFER CHARACTERISTICS
C*** OUT TO DISK
C***
C**************************************************************
C
      SUBROUTINE DWRITE(IBUF,N)
      INTEGER IBUF(256)
      CALL PACK(IBUF,256)
      WRITE(N) IBUF
      RETURN
      END
C*********************** PUNCH *********************************
C***
C*** THIIS PROGRAM IS A SUBROUTINE FOR PUNCHING PAPER TAPES
C***
C*** USAGE: CALL PUNCH(IBUF) - WHERE IBUF IS AN INTEGER ARRAY OF
C***      256 NUMBERS IN THE RANGE 0-255.
C***
C****************************************************************
C
      SUBROUTINE PUNCH(BUF)
      INTEGER BUF(256),DELETE,NULLS(50)
      DATA NULLS/50*:0000/,DELETE/:FFFF/
C.....PUNCH BUFFER TO TAPE
C
      WRITE(4,96) NULLS
96    FORMAT(50A2,A1)
      CALL PDEL
C
      IF(BUF(1).LT.1) N = 0
      IF(BUF(1).GT.0) N = ALOG10(FLOAT(BUF(1))+.1)
      J1 = 1
      DO 2 I=2,256
      IF(BUF(1).LT.1) M = 0
      IF(BUF(I).GT.0) M = ALOG10(FLOAT(BUF(I))+.1)
      IF(M.EQ.N) GOTO 2
      J2 = I-1
      CALL WRITR(J1,J2,N,BUF)
      N = M
      J1 = I
2     CONTINUE
```

```
      CALL WRITR(J1,256,N,BUF)
C
      WRITE(4,96) NULLS
C
       RETURN
      END
C************************* WRITR *************************
C***
C***  SUBROUTINE CALLED BY PUNCH TO DO THE ACTUAL PUNCHING
C***
C*****************************************************************
C
      SUBROUTINE WRITR(J1,J2,N,BUF)
      INTEGER BUF(256)
C
      IF(J1.NE.1) GOTO 3
      IF(N.EQ.0) WRITE(4,90) (BUF(J),J=J1,J2)
      IF(N.EQ.1) WRITE(4,91) (BUF(J),J=J1,J2)
      IF(N.EQ.2) WRITE(4,92) (BUF(J),J=J1,J2)
      GOTO 4
3      IF(N.EQ.0) WRITE(4,93) (BUF(J),J=J1,J2)
      IF(N.EQ.1) WRITE(4,94) (BUF(J),J=J1,J2)
      IF(N.EQ.2) WRITE(4,95) (BUF(J),J=J1,J2)
4     RETURN
C  FORMATS
C
93    FORMAT(36I2)
90    FORMAT(I1,(36I2))
91    FORMAT(I2,(24I3))
94    FORMAT(24I3)
92    FORMAT(I3,(18I4))
95    FORMAT(18I4)
C
      END
C********************** PDEL ********************************
C***
C***  PDEL IS A SUBROUTINE WHICH SENDS A DELETE CHARACTER TO
C***  THE PAPER TAPE PUNCH
C***
C**************************************************************
 TITL PDEL
 NAM PDEL
PPDEVA EQU :FC
PPINT EQU :130
PDEL ENT
 LDA DELADR
 LLA 1
 SAI 1
 STA PPINT+2
 LAM 1
 STA PPINT+1
 LDA EOBADR
```

```
 STA PPINT+5
 DIN
 LDA =:210
 OTA PPDEVA+1
 EIN
 WAIT
EOBADR DATA $+1
EOB ENT
 RTN PDEL
DELADR DATA $+1
DEL DATA :FF00
 LPOOL
 END
```