APPENDIX C: The Computer Program and Flow Chart

Program Wagner implements the procedure given in Appendix B for solving the integral equation derived in Appendix A. Flexibility is obtained by using appropriate versions of three subroutines:

(1) TERRANE, which returns the height, slope, and ground constants (σ, ϵ_r) as a function of distance, x. By writing appropriate statements in this subroutine the user can define any propagation path he needs. The general form of the subroutine TERRANE is shown on page 49, and two particular implementations used for examples in this report are listed on pages 50 and 51 .

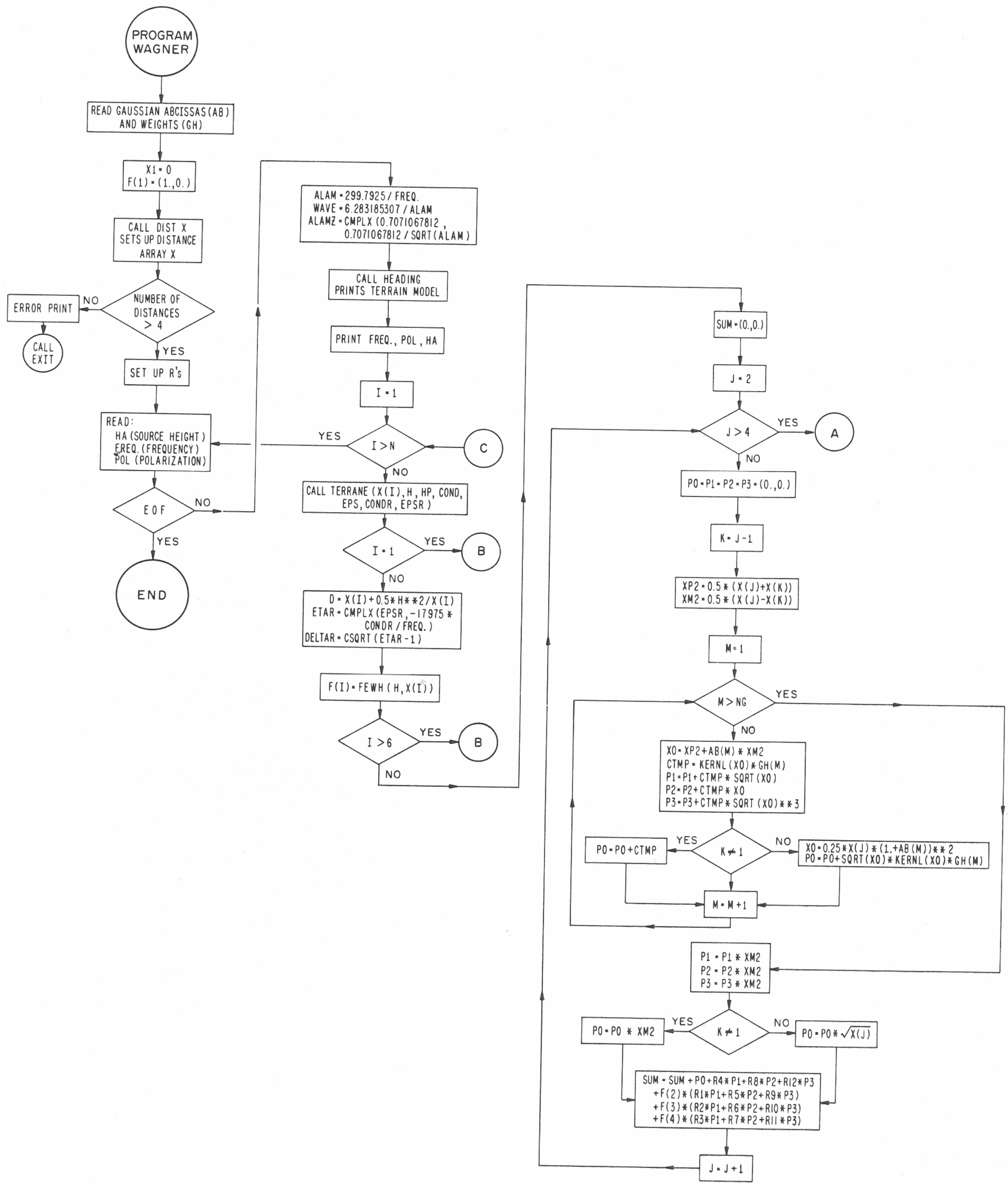
(2) DISTX, which returns the set of distances x(I) at which the function F(x) will be calculated. The general form of DISTX is shown on page 45, and two particular implementations are shown on pages 46 and 47 .

(3) KERNL, which computes the kernel of the integral equation. Program Wagner can be used to solve other integral equations of the form (B-1) if the kernel includes the factor $[s(x-s)]^{-\frac{1}{2}}$ by modifying subroutine KERNL. For example, WAGNER can solve Hufford's integral equation.

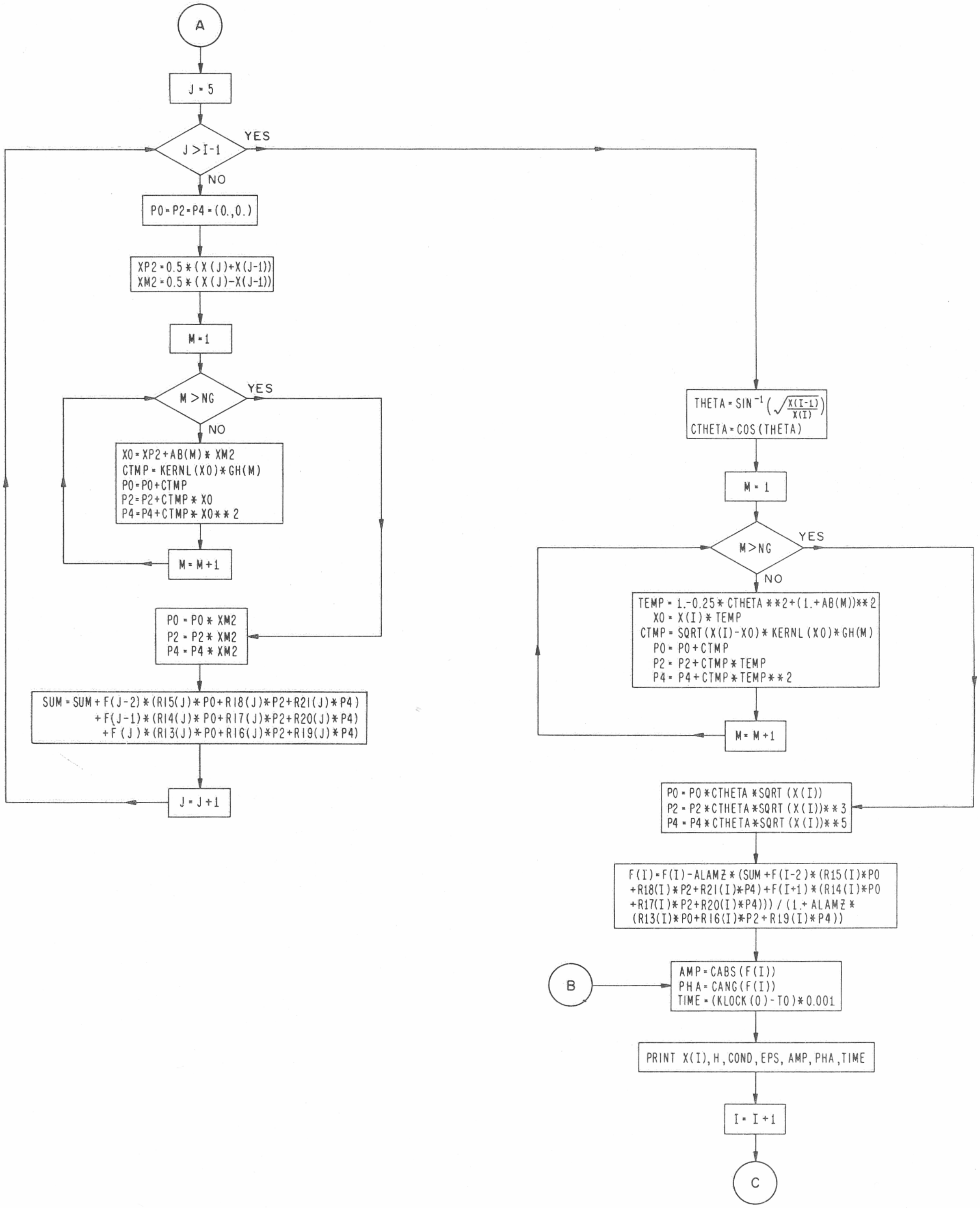
Comment cards in the listings that follow explain the program's operation. The input card sequence for Program WAGNER is

Card	Cols.	Description
1	1-10	The number of Gaussian quadrature abcissas and weights (5 recommended)
2 through 4	3-33 & 36-66	Values for the Gaussian weights and abcissas.
5 through N+4	1-10	The N points at which the attenuation function is to be calculated. These distances are read in kilometers, by DISTX.
N + 5		A blank card which signals the end of the distance deck when the form of DISTX is that given on page 47. When DISTX takes the form given on page 46, no blank card is required.
N + 6	1-10	Source height in kilometers.
	11-20	Frequency in Megahertz.
	21-30	Polarization, 1. = vertical, 2. = horizontal.

Following is a flow chart together with a statement listing (Fortran 3800) of the computer program, and a sample output.



Flow chart for computer program



Flow chart for computer program

```
      PROGRAM WAGNER
C
C       A PROGRAM TO COMPUTE HF GROUND WAVE ATTENUATION
C       IRREGULAR, INHOMOGENEOUS TERRAIN.  REFERENCE:
C       TELECOMMUNICATIONS RESEARCH REPORT, No. 7 , 1970.
C
      DIMENSION IPOL(2)
      COMMON /0/ F(2000),R13(2000),R14(2000),R15(2000),R16(2000),
     1 R17(2000),R18(2000),R19(2000),R20(2000),R21(2000)
      COMMON /1/ HA
      COMMON /2/ D,H,HP
      COMMON /3/ DELTAR,WAVE
      COMMON /4/ FREQ,POL
      COMMON /5/ NG,AB(48),GH(48)
      COMMON /6/ N,X(2001),I
      TYPE DOUBLE DAB,DGH
      COMPLEX FEWH,F,ALAMZ,SUM,DELTAR,ETAR
      COMPLEX KERNL,P0,P1,P2,P3,P4,CTMP
      IPOL(1)=8H  VERTIC  $  IPOL(2)=8HHORIZONT
C
C          READ GAUSSIAN QUADRATURE ABCISSAS AND WEIGHTS
C
      READ 1000, NG
 1000 FORMAT(I10)
      NR=(NG+1)/2
      DO 1 L=1,NR
      READ 1010, DAB,DGH
 1010 FORMAT(2D33.25)
      J=NG-L+1
      AB(L)=DAB
      AB(J)=-AB(L)
      GH(L)=DGH
    1 GH(J)=GH(L)
C
C         CALL SUBROUTINE TO SET UP DISTANCE ARRAY X IN METERS
C         START WITH X(2).  X(1)=0. HAS ALREADY BEEN SET.
C         THE DISTANCES DO NOT HAVE TO BE EQUALLY SPACED.
C         SUBROUTINE DISTX SHOULD MAKE SURE N < 2000
C
      X(1)=0.
      F(1)=(1.,0.)
      CALL DISTX
C
C         MAKE SURE THERE ARE AT LEAST 4 DISTANCES
      IF (N.GE.4) GO TO 4
      PRINT 1040
 1040 FORMAT (*0NUMBER OF DISTANCES 0 4*)
      CALL EXIT
C
    4 SQRTX2=SQRT(X(2))
      SQRTX3=SQRT(X(3))
      SQRTX4=SQRT(X(4))
      D1=SQRT(X(2)*X(3)*X(4))*(X(2)*(SQRTX4-SQRTX3)+X(3)*(SQRTX2-SQRTX4)
     1 +X(4)*(SQRTX3-SQRTX2))
      R1=X(3)*X(4)*(SQRTX4-SQRTX3)/D1
      R2=X(2)*X(4)*(SQRTX2-SQRTX4)/D1
      R3=X(2)*X(3)*(SQRTX3-SQRTX2)/D1
      R4=(X(2)*(SQRTX4**3-SQRTX3**3)+X(3)*(SQRTX2**3-SQRTX4**3)
     1 +X(4)*(SQRTX3**3-SQRTX2**3))/D1
```

```
      R5=SQRT(X(3)*X(4))*(X(3)-X(4))/D1
      R6=SQRT(X(2)*X(4))*(X(4)-X(2))/D1
      R7=SQRT(X(2)*X(3))*(X(2)-X(3))/D1
      R8=(SQRTX2*(SQRTX3**3-SQRTX4**3)+SQRTX3*(SQRTX4**3-SQRTX2**3)
     1 +SQRTX4*(SQRTX2**3-SQRTX3**3))/D1
      R9=SQRT(X(3)*X(4))*(SQRTX4-SQRTX3)/D1
      R10=SQRT(X(2)*X(4))*(SQRTX2-SQRTX4)/D1
      R11=SQRT(X(2)*X(3))*(SQRTX3-SQRTX2)/D1
      R12=(SQRTX2*(X(4)-X(3))+SQRTX3*(X(2)-X(4))+SQRTX4*(X(3)-X(2)))/D1
      DO 10 M=5,N
      M1=M-1
      M2=M-2
      D2=(X(M2)-X(M1))*(X(M)**2-X(M)*(X(M1)+X(M2))+X(M1)*X(M2))
      R13(M)=X(M1)*X(M2)*(X(M2)-X(M1))/D2
      R14(M)=X(M)*X(M2)*(X(M)-X(M2))/D2
      R15(M)=X(M)*X(M1)*(X(M1)-X(M))/D2
      R16(M)=(X(M1)**2-X(M2)**2)/D2
      R17(M)=(X(M2)**2-X(M)**2)/D2
      R18(M)=(X(M)**2-X(M1)**2)/D2
      R19(M)=(X(M2)-X(M1))/D2
      R20(M)=(X(M)-X(M2))/D2
   10 R21(M)=(X(M1)-X(M))/D2
C
C           READ SOURCE HEIGHT, FREQUENCY, AND POLARIZATION
C     COL          DESCRIPTION
C     1-10       SOURCE HEIGHT, KM
C    11-20       FREQUENCY, MHZ
C    21-30       POLARIZATION, 1. = VERTICAL, 2. = HORIZONTAL
C
   20 READ 2000, HA,FREQ,POL
 2000 FORMAT (3F10.4)
      IF (EOF,60) 999,22
   22 HA=HA*1.E3
      KPOL=POL
      ALAM=2.997925E2/FREQ
      WAVE=6.283185307/ALAM
      ALAMZ = ((0.7071067812,0.7071067812)/SQRTF(ALAM))
      CALL HEADING
      PRINT 2500, FREQ,IPOL(KPOL),HA
 2500 FORMAT (*0FREQUENCY =*,F10.2,10X,A8,*AL POLARIZATION*,10X,*ANTENNA
     1 HEIGHT =*,F6.2,* METERS*//
     2 9X,*X*,14X,*Z*,10X,*CONDUCTIVITY*,3X,*DIELECTRIC*,15X,*F(X)*,22X,
     3 *TIMING*/8X,*(M)*,12X,*(M)*,12X,*(MHO/M)*,6X,*CONSTANT*,8X,*MAG*,
     4 13X,*ARG*,16X,*(SEC)*)
      T0=KLOCK(0)
C
C           LOOP ON DISTANCE
C
      DO 100 I=1,N
      CALL TERRANE (X(I),H,HP,COND,EPS,CONDR,EPSR)
      IF (I.EQ.1) GO TO 75
      D=X(I)+(H**2)/(2.*X(I))
      ETAR = CMPLX(EPSR,-17975.*CONDR/FREQ)
      DELTAR = CSQRT(ETAR - 1.)
```

```
      IF(KPOL.EQ.1) DELTAR = DELTAR/ETAR
      F(I)=FEWH(H,X(I))
      IF (I.LE.6) GO TO 75
C
C                 J = 2 THROUGH 4
C
      SUM=(0.,0.)
      DO 40 J=2,4
      P0=P1=P2=P3=(0.,0.)
      K=J-1
      XP2=0.5*(X(J)+X(K))
      XM2=0.5*(X(J)-X(K))
      DO 35 M=1,NG
      X0=XP2+AB(M)*XM2
      CTMP=KERNL(X0)*GH(M)
      P1=P1+CTMP*SQRT(X0)
      P2=P2+CTMP*X0
      P3=P3+CTMP*SQRT(X0)**3
      IF (K.NE.1) GO TO 33
      X0=0.25*X(J)*(1.+AB(M))**2
      P0=P0+SQRT(X0)*KERNL(X0)*GH(M)
      GO TO 35
   33 P0=P0+CTMP
   35 CONTINUE
      P1=P1*XM2
      P2=P2*XM2
      P3=P3*XM2
      IF (K.NE.1) GO TO 38
      P0=P0*SQRT(X(J))
      GO TO 40
   38 P0=P0*XM2
   40 SUM=SUM+P0+R4*P1+R8*P2+R12*P3 +F(2)*(R1*P1+R5*P2+R9*P3)
     1     +F(3)*(R2*P1+R6*P2+R10*P3)+F(4)*(R3*P1+R7*P2+R11*P3)
C
C                 J = 5 THROUGH I-1
C
      I1=I-1
      DO 50 J=5,I1
      P0=P2=P4=(0.,0.)
      XP2=0.5*(X(J)+X(J-1))
      XM2=0.5*(X(J)-X(J-1))
      DO 45 M=1,NG
      X0=XP2+AB(M)*XM2
      CTMP=KERNL(X0)*GH(M)
      P0=P0+CTMP
      P2=P2+CTMP*X0
   45 P4=P4+CTMP*X0**2
      P0=P0*XM2
      P2=P2*XM2
      P4=P4*XM2
   50 SUM=SUM+F(J-2)*(R15(J)*P0+R18(J)*P2+R21(J)*P4)
     1         +F(J-1)*(R14(J)*P0+R17(J)*P2+R20(J)*P4)
     2         +F(J)  *(R13(J)*P0+R16(J)*P2+R19(J)*P4)
C
```

```
C                   J=I
C
      THETA=ASINF(SQRT(X(I1)/X(I)))
      CTHETA=COSF(THETA)
      P0=P2=P4=(0.,0.)
      DO 55 M=1,NG
      TEMP=1.-0.25*CTHETA**2*(1.+AB(M))**2
      X0=X(I)*TEMP
      CTMP=SQRT(X(I)-X0)*KERNL(X0)*GH(M)
      P0=P0+CTMP
      P2=P2+CTMP*TEMP
   55 P4=P4+CTMP*TEMP**2
      P0=P0*CTHETA*SQRT(X(I))
      P2=P2*CTHETA*SQRT(X(I))**3
      P4=P4*CTHETA*SQRT(X(I))**5
C
C     EQUATION (B-11)
C
      F(I)=(F(I)-ALAMZ*(SUM+F(I-2)*(R15(I)*P0+R18(I)*P2+R21(I)*P4)
     1 +F(I1)*(R14(I)*P0+R17(I)*P2+R20(I)*P4)))/(1.+ALAMZ*(R13(I)*P0
     2 +R16(I)*P2+R19(I)*P4))
   75 AMP = CABS(F(I))
      PHA = CANG(F(I))
      TIME=(KLOCK(0)-T0)*0.001
      PRINT 8000, X(I),H,COND,EPS,AMP,PHA,TIME
 8000 FORMAT (*0*,F12.2,F18.9,F14.6,F13.4,E18.8,E16.8,F15.3)
  100 CONTINUE
C
      GO TO 20
  999 CALL EXIT
          END
```

```
      SUBROUTINE DISTX
C          READ DISTANCES IN KM AND CONVERTS THEM TO METERS
C          (A DISTANCE OF ZERO SIGNALS END OF DISTANCE DECK)
      COMMON /6/ N,X(2001),I
C
C     IN THIS SUBROUTINE THE USER MUST FILL
C     THE X(I) ARRAY WITH N VARIABLES.
C
      RETURN
      END
```

```
      SUBROUTINE DISTX
C          COMPUTES EQUALLY SPACED DISTANCES
      COMMON /6/ N,X(2001),I
C              INPUT
C          DMIN -- FIRST DISTANCE IN KM
C          DMAX -- MAXIMUM DISTANCE IN KM
C          DINC -- INCREMENT ON DISTANCE IN KM
C
      READ 1000, DMIN,DMAX,DINC
 1000 FORMAT (3F10.2)
      IF (DMIN.EQ.0.) DMIN=DMIN+DINC
      N=(DMAX-DMIN)/DINC+2
      DO 10 I=2,N
      X(I)=(DMIN+(I-2)*DINC)*1.E3
   10 CONTINUE
      RETURN
         END
```

Note, this is an example of subroutine DISTX.