6. Software - zdrojový program modelu KINFIL (FORTRAN)

INFIL – data TŘEBSÍN

```
$DEBUG
$NOTRUNCATE
      PROGRAM EINFIL
C     VERSION OF INFIL MODEL FOR EROSION
C     - KBUK FES CULS PRAGUE, 2014
C     NAME OF FILE EINFIL.FOR
      COMMON/A1/DELT,QU,N,P,TD,CN,II
      COMMON/A2/KT,TP,RP,WP,K
      COMMON/A3/MO,DAY,YEAR,SFFC,S,SO
      COMMON/A4/T(250),R(250),RE(250),RER(250)
      COMMON/A5/AR(5)
      COMMON/A6/SF,RETEN
      COMMON/A7/CUMP(250)
      COMMON/A8/AREA,L,Y,NN,NF
      COMMON/A9/DELTA(250),CNM
      COMMON/A10/DELP(250),QA(250),TM(250),QM(250)
      COMMON/A11/SUBPT1,SUBPT2,SUBPT3,SUBPT4
      COMMON/A12/TL
      COMMON/A13/CNAME(20)
      INTEGER QU,SUBPT1,SUBPT2,SUBPT3,SUBPT4,DAY,YEAR
      REAL KT,L
      CHARACTER*12 NDFILE
      CHARACTER*8 FNAME$
      OPEN(UNIT=5,FILE='treb4sim.dta',STATUS='OLD')
      OPEN(UNIT=7,FILE='treb4sim.out',STATUS='UNKNOWN')
      OPEN(UNIT=1,FILE='NDFILE',STATUS='UNKNOWN')
   62 FORMAT(2X,56(1H*))
C
   63 FORMAT(2X,'T H E    I N F I L T R A T I O N    M O D E L '/)
   64 FORMAT(2X,'THE KW-INFILTRATION MODEL COMPUTES NET RAINFALL FROM GR
     *OSS ONE USING'/2X,'MEIN - LARSON AND MOREL - SEYTOUX INFILTRATION
     *FORMULAE')
   61 FORMAT(2X,'FLOOD HYDROGRAPH CAN BE COMPUTED USING UH CONVOLUTION')
      WRITE(6,63)
      WRITE(6,62)
      WRITE(6,64)
      WRITE(6,61)
      WRITE(6,51)
      WRITE(7,63)
      WRITE(7,62)
      WRITE(7,64)
      WRITE(7,61)
      WRITE(7,51)
   51 FORMAT(//,2X,'IF SUBOPT1=1, VARIABLE RAINFALL RATES IS USED',/
     *2X,'IF SUBPT1=0, CONSTANT RAINFALL IS USED',
     */,2X,'IF SUBPT2=0, USER INPUTS KT AND SFFC'/
     *2X,'IF SUBPT2=1, KT AND SFFC ARE COMPUTED FROM CN'/
     *2X,'IF SUBPT3=1, MEASURED DISCHARGES SHOULD BE READ'/
     *2X,'IF SUBPT4=0, EROSION PROGRAM IS IMPLEMENTED'//)
C
   76 FORMAT(20A4)
  100 READ(5,76)CNAME
C
   65 FORMAT(2X,'NAME OF CATCHMENT:'/2X,20A4/2X,60(1H*))
      WRITE(6,65)CNAME
      WRITE(7,65)CNAME
      READ(5,38)SUBPT1,SUBPT2,SUBPT3,SUBPT4
   38 FORMAT(I2,2X,I2,2X,I2,2X,I2)
      READ(5,72)IDENT
   72 FORMAT(I1)
      WRITE(6,39)SUBPT1,SUBPT2,SUBPT3,SUBPT4
      WRITE(7,39)SUBPT1,SUBPT2,SUBPT3,SUBPT4
   39 FORMAT(2X,'SUBOPT1=',I2,3X,'SUBOPT2=',I2,3X,'SUBOPT3=',I2/
     *2X,'SUBOPT4='I2)
      IF(SUBPT2.EQ.0)GO TO 40
      READ(5,41)P,TD,CN,CNM
   41 FORMAT(4F10.3)
      WRITE(6,42)P,TD,CN
      WRITE(7,42)P,TD,CN
   42 FORMAT(2X,'STORM DEPTH P =',F10.3,2X,'MM'/2X,'STORM DURATION TD ='
     *,F10.3,2X,'HR'/2X,'CURVE NUMBER CN=',F10.3,2X,'(-)')
```

```
C     READ '(A12)',NDFILE
      CALL TABLE
      GO TO 43
   40 READ(5,2)KT,SO,P,TD,CN
    2 FORMAT(5F10.3)
      SFFC=(SO**2)/(2.0*KT)
      WRITE(6,15)KT,SO,SFFC,P,TD
      WRITE(7,15)KT,SO,SFFC,P,TD
   15 FORMAT(/,2X,'HYDRAULIC CONDUCTIVITY, KT =',F8.3,1X,'MM/HR',/,2X,
     *'SORPTIVITY, SO =',F8.3,1X,'MM/HR**0.5',/2X,
     *'STORAGE SUCTION FACTOR, SFFC =',F8.3,1X,'MM',/,
     *2X,'TOTAL PRECIP, P =',F8.3,1X,'MM',/,2X,'DURATION TIME,TD =',F8.3
     *,1X,'HR',)
      GO TO 8
   43 READ(5,27)AREA,L,Y,RETEN
   27 FORMAT(4F10.5)
      WRITE(6,28)AREA,L,Y
      WRITE(7,28)AREA,L,Y
   28 FORMAT(2X,'AREA=',F10.5,2X,'KM2',
     */,2X,'LENGTH TO DIVIDE=',F10.2,2X,'M'/2X,
     *'AVG CATCHMENT SLOPE=',F10.2,2X,'PERCENT')
    8 READ(5,9)N,II,DELT
    9 FORMAT(I3,2X,I3,2X,F10.5)
C     WRITE(6,20)N,II,DELT
C     WRITE(7,20)N,II,DELT
   20 FORMAT(/,2X,'N=',I3,5X,'II =',I3,5X,'DELT =',F10.5/)
C
C     WRITE(6,49)
C     WRITE(7,49)
   49 FORMAT(2X,'N IS NUMBER OF TIME STEPS IN USERS STORM'/
     *2X,'II IS NUMBER OF TIME STEPS IN REQUIRED HYDROGRAPH'/
     *2X,'DELT IS LENGHT OF TIME STEP IN HR'/
     *2X,'INPUT RAIN IS IN MM/HR',/)
C
      IF(IDENT.NE.1)GOTO 33
C
      READ(5,11)DAY,MO,YEAR
   11 FORMAT(I2,2X,I2,2X,I2)
      READ(5,16)(AR(I),I=1,5)
   16 FORMAT(5F8.0)
C
      WRITE(6,13)DAY,MO,YEAR
      WRITE(7,13)DAY,MO,YEAR
   13 FORMAT(2X,'DATE OF EVENT:',I2,'/',I2,'/','19',I2,/)
   14 FORMAT(2X,'ANTECEDENT DAILY RAIN AR(I):'/2X,5F7.1/)
      WRITE(6,14)(AR(I),I=1,5)
      WRITE(7,14)(AR(I),I=1,5)
      READ(5,31)QM(1)
   31 FORMAT(F10.3)
      WRITE(6,48)QM(1)
      WRITE(7,48)QM(1)
   48 FORMAT(2X,'INITIAL DISCHARGE OBSERVED: QM(1)=',F10.3,1X,'M3/S,'/)
   33 CONTINUE
C     SCREEN CONTROL
C     READ '(A12)',NDFILE
   99 T(1)=DELT
      DO 22 I=2,II
   22 T(I)=T(I-1)+DELT
      READ(5,10)(R(I),I=1,N)
   10 FORMAT(10F8.0)
    3 FORMAT(2X,10F7.2,/,2X,10F7.2,/,2X,10F7.2,/,2X,10F7.2,/,2X,10F7.2,/
     *,2X,10F7.2,/,2X,10F7.2,/2X,10F7.2,/2X,10F7.2,/2X,10F7.2,/2X,10F7.2
     */2X,10F7.2,/2X,10F7.2,/2X,10F7.2,/2X,10F7.2,/)
   32 FORMAT(2X,10F6.3,/,2X,10F6.3,/,2X,10F6.3,/,2X,10F6.3,/,2X,10F6.3,/
     *,2X,10F6.3,/,2X,10F6.3,/2X,10F6.3,/2X,10F6.3,/2X,10F6.3,/2X,10F6.3
     */2X,10F6.3,/2X,10F6.3,/2X,10F6.3,/2X,10F6.3,/)
   52 CONTINUE
C     WRITE(6,36)
C     WRITE(7,36)
   36 FORMAT(2X,'TIME STEPS, HOURS')
C     WRITE(6,3) (T(I),I=1,II)
   45 CONTINUE
C     WRITE(6,23)
C     WRITE(7,3) (T(I),I=1,II)
C     WRITE(7,23)
   23 FORMAT(2X,'RAINFALL INTENSITIES, MM/HR')
C     WRITE(6,3)(R(I),I=1,N)
```

```
C     WRITE(7,3)(R(I),I=1,N)
C     COMPUTE CUMULATIVE STEP PRECIP,  CUMP
      DO 24 I=1,N
      IF(I.EQ.1)GO TO 25
      CUMP(I)=CUMP(I-1)+R(I)*(T(I)-T(I-1))
      GO TO 24
   25 CUMP(I)=R(I)*T(I)
   24 CONTINUE
C     WRITE(6,26)
C     WRITE(7,26)
   26 FORMAT(2X,'STEPS OF CUMULATIVE PRECIP (MM)')
C     WRITE(6,3)(CUMP(I),I=1,N)
C     WRITE(7,3)(CUMP(I),I=1,N)
      IF(SUBPT3.EQ.0)GO TO 80
      READ(5,56)(QM(I),I=1,II)
   56 FORMAT(10F8.0)
      WRITE(6,57)
      WRITE(7,57)
   57 FORMAT(2X,'THE OBSERVED DISCHARGE VALUES IN M3/S',/)
C     IF(IDENT.EQ.0)GO TO 74
C     DO 75 I=1,II
C  75 QM(I)=QM(I)*0.001
   74 CONTINUE
      WRITE(6,32)(QM(I),I=1,II)
      WRITE(7,32)(QM(I),I=1,II)
   80 CONTINUE
    5 READ(5,73)NEXT
   73 FORMAT(I1)
      SF=SFFC
C     SCREEN CONTROL
C     READ '(A12)',NDFILE
      IF(SUBPT1.EQ.1)GO TO 30     !
      CALL CONST
      GO TO 66
   30 CONTINUE
      CALL PONTI
C     READ '(A12)',NDFILE
      CALL PPINF
   66 CONTINUE
      PRINT *,'**** END OF INFILTRATION PART OF MODEL ****'
      IF(SUBPT4.EQ.0)GO TO 6
      PRINT *,'**** BEGINING OF ROUTING PART OF MODEL ****'
      SRER=0.0
      DO 7 I=1,N
    7 SRER=SRER+RER(I)
      IF(SRER.LE.0.0)GO TO 6
   12 CALL UH
C     SCREEN CONTROL
C     READ '(A12)',NDFILE
      CALL ROUTE
C     SCREEN CONTROL
C     READ '(A12)',NDFILE
      IF(IDENT.EQ.1)CALL GODFI
C     SCREEN CONTROL
      READ '(A12)',NDFILE
      write(*,'(A\)')' NAME OF EVENT DATA:'
      read(*,'(A)')FNAME$
      open(unit=8,file=FNAME$)
C     write(8,'(2x,i3,3f8.3)')(i,r(i),rer(i),qa(i),i=1,ii)
      write(8,'(2x,i3,f8.3)')(i,qa(i),i=1,ii)
      DO 4 I=1,II
    4 QA(I)=0.0
      close(8)
    6 CONTINUE
      IF(NEXT.NE.0) GO TO 100
   70 STOP
      END
      SUBROUTINE PONTI
      COMMON/A1/DELT,QU,N,P,TD,CN,II
      COMMON/A2/KT,TP,RP,WP,K
      COMMON/A4/T(250),R(250),RE(250),RER(250)
      COMMON/A6/SF,RETEN
      DIMENSION PT(250)
      REAL KT
C
C     THIS SUBROUTINE CALCULATES PONDING TIME FOR A VARIABLE RAINFALL
C     INTENSITY EVENT
```

```
c     WRITE(6,33)
c     WRITE(7,33)
   33 FORMAT(/2X,60(1H*),/)
      WRITE(6,27)
      WRITE(7,27)
   27 FORMAT(/,2X,'OUTPUT OF SUBROUTINE PONTI',/)
      I=0
   10 I=I+1
      IF(I.GT.N)GOTO 18
      IF(R(I).LE.KT)GOTO 10
      IF(I.EQ.1)GOTO 11
      II=I-1
      SUMP=0.
      DO 12 J=1,II
      IF(J.EQ.1)GOTO 14
      SUMP=SUMP+R(J)*(T(J)-T(J-1))
      GOTO 12
   14 SUMP=SUMP+R(J)+T(J)
   12 CONTINUE
      PT(I)=T(I-1)+(1./R(I))*((SF/((R(I)/KT)-1.))-SUMP)
C     TEST COMPUTED PONDING TIME AGAINST PREVIOUS TIME STEP
      IF(PT(I)-T(I-1))13,13,17
   13 TP=T(I-1)
      RP=R(I)
      GOTO 23
C     TEST COMPUTED PONDING TIME AGAINST TIME STEP OF CONSIDERATION
   17 IF(PT(I)-T(I))15,15,10
   11 PT(I)=(1./R(I))*(SF/((R(I)/KT)-1.))
      IF(PT(I).GT.T(I))GO TO 10
   15 TP=PT(I)
      RP=R(I)
   23 K=0
      WP=0.0
      DO 20 J=1,N
      IF(T(J).GT.TP)GO TO 20
      IF(J.EQ.1)GO TO 21
      WP=WP+R(J)*(T(J)-T(J-1))
      GO TO 22
   21 WP=WP+R(J)*T(J)
   22 K=K+1
   20 CONTINUE
      IF(K.EQ.0)GO TO 25
      WP=WP+RP*(TP-T(K))
      GO TO 26
   25 WP=RP*TP
   26 WRITE(6,24)TP,RP,WP
      WRITE(7,24)TP,RP,WP
   24 FORMAT(2X,'PONDING TIME=',F8.3,1X,'HR',5X,'PONDING RAINFALL=',
     *F8.3,2X,'MM/HR'/2X,'DEPTH OF RAIN INFILTRATED PREVIOUS TO PONDING=
     *',F8.3,1X,'MM')
      IF(K.GT.0)GO TO 28
      GO TO 16
   28 WRITE(6,29)K,T(K)
      WRITE(7,29)K,T(K)
   29 FORMAT(2X,'LAST FULL TIME STEP T(',I3,')=',F8.3,1X,'HR')
      GO TO 16
   18 WRITE(6,19)
      WRITE(7,19)
   19 FORMAT(2X,'PONDING NEVER OCCURS')
   16 continue
      RETURN
      END
      SUBROUTINE CONST
      COMMON/A1/DELT,QU,N,P,TD,CN,II
      COMMON/A2/KT,TP,RP,WP,K
      COMMON/A4/T(250),R(250),RE(250),RER(250)
      COMMON/A6/SF,RETEN
      COMMON/A8/AREA,L,Y,NN,NF
      COMMON/A10/DELP(250),QA(250),TM(250),QM(250)
      DIMENSION W(250),DELW(250),IR(250)
      REAL KT,IR
C
C     THIS SUBROUTINE COMPUTES EXCESS RAINFALL BY INFILTRATION EQUATION
C     FOR A CONSTANT INTENSITY EVENT
C
c     WRITE(6,33)
c     WRITE(7,33)
```

```
   33 FORMAT(/2X,60(1H*),/)
      WRITE(6,21)
      WRITE(7,21)
   21 FORMAT(/,2X,'OUTPUT OF SUBROUTINE CONST, CONSTANT RAINFALL '/
     *2X,'BY INFILTRATION APPROACH',)
      CR=P/TD
      SORP =SQRT(2.*KT*SF)
      RSTAR=CR/KT
      IF(RSTAR.LE.1.)GO TO 4
C     COMPUTE MEIN AND LARSON PONDING TIME
      TP=SF/(CR*(RSTAR-1.0))
      Write(6,35)TP
      Write(7,35)TP
   35 format(2x,'PONDING TIME TP= ',2X,f8.3,2X,'HR')
      IF(TP.GE.TD)GO TO 4
      KK=0
      DO 22 I=1,N
      IF(T(I).GE.TP)GO TO 22
      KK=KK+1
   22 CONTINUE
      M=KK+1
      RATIO=RSTAR/(RSTAR-1.0)
      WP=CR*TP
      B=0.5*TP*(RATIO**3)
      DO 20 I=M,N
      W(I)=WP+SORP*RATIO*(SQRT(T(I)-TP+B)-SQRT(B))+KT*(T(I)-TP)
      IF(I.EQ.M)GO TO 11
      DELW(I)=W(I)-W(I-1)
      IR(I)=DELW(I)/(T(I)-T(I-1))
      GO TO 12
   11 DELW(I)=W(I)-WP
      IR(I)=DELW(I)/(T(I)-TP)
   12 CONTINUE
      IF(CR-IR(I))13,13,14
   13 IR(I)=CR
      RE(I)=0.0
      IF(I.EQ.M)GO TO 15
      DELW(I)=CR*(T(I)-T(I-1))
      W(I)=W(I-1)+DELW(I)
      GO TO 20
   15 DELW(I)=CR*(T(I)-TP)
      W(I)=W(I-1)+DELW(I)
      GO TO 20
   14 RE(I)=CR-IR(I)
   20 CONTINUE
C     SUBTRACT RETENTION
      RET=RETEN
      DO 27 I=M,N
      IF(I.EQ.M)GO TO 23
      PS=RE(I)*(T(I)-T(I-1))
      IF(RET-PS)26,25,25
   26 RER(I)=(PS-RET)/(T(I)-T(I-1))
      RET=0.0
      GO TO 27
   23 PS=RE(I)*(T(I)-TP)
      IF(RET-PS)24,25,25
   24 RER(I)=(PS-RET)/(T(I)-TP)
      RET=0.0
      GO TO 27
   25 RER(I)=0.0
      RET=RET-PS
   27 CONTINUE
    5 IFLAG=0
      DO 28 I=M,N
      IF(RER(I).EQ.0.0.AND.IFLAG.EQ.0)GO TO 28
      IFLAG=IFLAG+1
      DELP(IFLAG)=RER(I)*DELT
      TM(IFLAG)=T(I)
   28 CONTINUE
      NK=N+1    !
      DO 36 I=1,NK
      RE(I)=RE(I)*DELT
      RER(I)=RER(I)*DELT
   36 CONTINUE
      NIN=INT(TD/DELT)+1
c     DO 1 I=NIN,N
c     R(I)=0.0
```

```
c     RE(I)=0.0
c     RER(I)=0.0
    1 CONTINUE
C     READ '(A12)',NDFILE
      NF=IFLAG
      WRITE(6,17)
      WRITE(7,17)
   17 FORMAT(5X,'T(HR)',6X,'W(MM)',3X,'DELW(MM)',1X,'IR(MM/HR)',2X,'R(MM
     */HR)',5X,'RE(MM/HR)',2X,'RER(MM/HR)',/)
      WRITE(6,18)TP,WP,WP
      WRITE(7,18)TP,WP,WP
   18 FORMAT(1X,3F10.3)
      DO 16 I=1,NK
      WRITE(6,19)T(I),W(I),DELW(I),IR(I),R(I),RE(I),RER(I)
      WRITE(7,19)T(I),W(I),DELW(I),IR(I),R(I),RE(I),RER(I)
   19 FORMAT(1X,7F10.3)
   16 CONTINUE
C     READ '(A12)',NDFILE
C     CHECK MASS BALANCE
      WRITE(6,7)
      WRITE(7,7)
    7 FORMAT(/,2X,'MASS BALANCE CHECK',)
      PECONS=P-W(N)
    6 RET=RETEN-RET
      WRITE(6,10)PECONS,W(N),RET,P
      WRITE(7,10)PECONS,W(N),RET,P
   10 FORMAT(2X,'EXCESS PRECIP=',F8.3,2X,'MM',/,2X,
     *  'CUMULATIVE INFILTRATION=',F8.3,2X,'MM',/,
     *2X,'RETENTION=',F8.3,2X,'MM',/,
     *2X,'TOTAL PRECIP=',F8.3,2X,'MM'//)
      IF(NF.EQ.0)GO TO 4
      GO TO 2
    4 WRITE(6,3)
      WRITE(7,3)
    3 FORMAT(/,5X,'ALL RAINFALL INFILTRATES - NO RUNOFF IS PRODUCED',//)
C     SCREEN CONTROL
    2 CONTINUE
C     READ '(A12)',NDFILE
      RETURN
      END
      SUBROUTINE TABLE
      COMMON/A1/DELT,QU,N,P,TD,CN,II
      COMMON/A2/KT,TP,RP,WP,K
      COMMON/A3/MO,DAY,YEAR,SFFC,S,SO
      COMMON/A11/SUBPT1,SUBPT2,SUBPT3,SUBPT4
      INTEGER DAY,YEAR
      REAL KT
      WRITE(6,33)
      WRITE(7,33)
   33 FORMAT(/2X,60(1H*),/)
      WRITE(6,28)
      WRITE(7,28)
   28 FORMAT(/,2X,'OUTPUT OF SUBROUTINE TABLE',/)
      IF(CN.LE.75.)GO TO 11
      KT=(100.-CN)/12.4
      GO TO 12
   11 IF(CN.LE.36.)GO TO 13
      KT=31.394-0.391*CN
      GO TO 12
   13 KT=47.066-0.823*CN
   12 CONTINUE
      IF(CN.LE.65.)GOTO 14
      SORP=(100.-CN)/2.512
      GO TO 15
   14 SORP=30.251-0.146*CN
   15 CONTINUE
      SFFC=(SORP**2)/(2.*KT)
      WRITE(6,19)KT,SFFC
      WRITE(7,19)KT,SFFC
   19 FORMAT(2X,'HYDRAULIC CONDUCTIVITY KT=',F10.3,2X,'MM/HR',/,2X,'STOR
     *AGE SUCTION FACTOR AT FIELD CAPACITY SFFC=',F10.3,2X,'MM',/)
      WRITE(6,33)
      WRITE(7,33)
      RETURN
      END
      SUBROUTINE PPINF
      COMMON/A1/DELT,QU,N,P,TD,CN,II
```

```
      COMMON/A2/KT,TP,RP,WP,K
      COMMON/A3/MO,DAY,YEAR,SFFC,S,SO
      COMMON/A4/T(250),R(250),RE(250),RER(250)
      COMMON/A6/SF,RETEN
      COMMON/A8/AREA,L,Y,NN,NF
      COMMON/A10/DELP(250),QA(250),TM(250),QM(250)
      DIMENSION W(250),DELW(250),IR(250)
      INTEGER DAY,YEAR
      REAL IR,KT
C
C     THIS SUBROUTINE COMPUTES POST-PONDING INFILTRATION FOR A VARIABLE
C     INTENSITY RAINFALL EVENT
      WRITE(6,33)
      WRITE(7,33)
   33 FORMAT(/2X,60(1H*),/)
      WRITE(6,21)
      WRITE(7,21)
   21 FORMAT(/2X,'OUTPUT OF SUBROUTINE PPINF, VARIABLE RAINFALL'/
     *2X,'INFILTRATION APPROACH'/)
      RSORP=SQRT(2.*KT*((SF+WP)**2)/SF)
      RSTARP=RP/KT
      B=0.5*((SF+WP)**2)/(KT*SF*((RSTARP-1.)**2))
      M=K+1
      DO 10 I=M,N
      W(I)=WP+RSORP*(SQRT(T(I)-TP+B)-SQRT(B))+KT*(T(I)-TP)
      IF(I.EQ.M)GO TO 11
      DELW(I)=W(I)-W(I-1)
      IR(I)=DELW(I)/(T(I)-T(I-1))
      GO TO 12
   11 DELW(I)=W(I)-WP
      IR(I)=DELW(I)/(T(I)-TP)
   12 CONTINUE
      IF(R(I)-IR(I))13,13,14
   13 IR(I)=R(I)
      RE(I)=0.0
      IF(I.EQ.M)GO TO 15
      DELW(I)=R(I)*(T(I)-T(I-1))
      W(I)=W(I-1)+DELW(I)
      GO TO 10
   15 DELW(I)=R(I)*(T(I)-TP)
      W(I)=W(I-1)+DELW(I)
      GO TO 10
   14 RE(I)=R(I)-IR(I)
   10 CONTINUE
C     SUBTRACT RETENTION
      RET=RETEN
      DO 27 I=M,N
      IF(I.EQ.M)GO TO 23
      PS=RE(I)*(T(I)-T(I-1))
      IF(RET-PS)26,25,25
   26 RER(I)=(PS-RET)/(T(I)-T(I-1))
      RET=0.0
      GO TO 27
   23 PS=RE(I)*(T(I)-TP)
      IF(RET-PS)24,25,25
   24 RER(I)=(PS-RET)/(T(I)-TP)
      RET=0.0
      GO TO 27
   25 RER(I)=0.0
      RET=RET-PS
   27 CONTINUE
      IFLAG=0
      DO 28 I=M,N
      IF(RER(I).EQ.0.0.AND.IFLAG.EQ.0)GO TO 28
      IFLAG=IFLAG+1
      DELP(IFLAG)=RER(I)*DELT
      TM(IFLAG)=T(I)
   28 CONTINUE
C     READ '(A12)',NDFILE
      NF=IFLAG
      WRITE(6,9)
      WRITE(7,9)
    9 FORMAT(5X,'T(HR) = TIME'/,15X,'W(MM) = CUMULATIVE INFILTRATION'/,
     *15X,'DELW(MM) = INCREMENTAL INFILTRATION'/,
     *15X,'IR(MM/HR) = INFILTRATION RATE'/,
     *15X,'R(MM/HR) = RAINFALL RATE'/,
     *15X,'RE(MM/HR) = RAINFALL RATE AFTER INFILTR. SUBTRACTED'/,
```

```
     *15X,'RER(MM/HR) = NET RAINFALL AFTER RETENTION SUBTRACTED'/)
      WRITE(6,17)
      WRITE(7,17)
   17 FORMAT(5X,'T(HR)',6X,'W(MM)',3X,'DELW(MM)',1X,'IR(MM/HR) ','R(MM/
     *HR)',5X,'RE(MM/HR)',2X,'RER(MM/HR)',/)
      WRITE(6,18)TP,WP,WP
      WRITE(7,18)TP,WP,WP
   18 FORMAT(1X,3F10.3)
      DO 16 I=M,N
      WRITE(6,19)T(I),W(I),DELW(I),IR(I),R(I),RE(I),RER(I)
      WRITE(7,19)T(I),W(I),DELW(I),IR(I),R(I),RE(I),RER(I)
   19 FORMAT(1X,7F10.3)
   16 CONTINUE
C     READ '(A12)',NDFILE
C     CHECK MASS BALANCE
      WRITE(6,7)
      WRITE(7,7)
    7 FORMAT(/,2X,'MASS BALANCE CHECK',/)
      PE=0.0
      IF(NF.EQ.0)GO TO 6
      DO 8 I=1,NF
    8 PE=PE+DELP(I)
    6 RET=RETEN-RET
      WRITE(6,20)PE,W(N),RET,P
      WRITE(7,20)PE,W(N),RET,P
   20 FORMAT(2X,'EXCESS PRECIP=',F8.3,2X,'MM',/,
     *2X,'CUMULATIVE INFILTRATION=',F8.3,2X,'MM',/,
     *2X,'RETENTION=',F8.3,2X,'MM',/,
     *2X,'TOTAL PRECIP=',F8.3,2X,'MM',/)
      IF(NF.EQ.0)GO TO 5
      GO TO 3
    5 WRITE(6,4)
      WRITE(7,4)
    4 FORMAT(5X,'ALL RAINFALL INFILTRATES - NO RUNOFF IS PRODUCED',/)
C     SCREEN CONTROL
    3 CONTINUE
C     READ '(A12)',NDFILE
      RETURN
      END
      SUBROUTINE UH
      COMMON/A1/DELT,QU,N,P,TD,CN,II
      COMMON/A4/T(250),R(250),RE(250),RER(250)
      COMMON/A8/AREA,L,Y,NN,NF
      COMMON/A9/DELTA(250),CNM
      COMMON/A10/DELP(250),QA(250),TM(250),QM(250)
      COMMON/A12/TL
      DIMENSION RATIOT(20),RATIOQ(20),QQT(250)
      REAL L
C
      WRITE(6,33)
      WRITE(7,33)
   33 FORMAT(2X,60(1H*),/)
      WRITE(6,20)
   20 FORMAT(/,2X,'OUTPUT OF SUBROUTINE UH',/)
C
      S=25.4*((1000./(CN-5.0))-10.)
      TL=(((3.28*L)**0.8)*(((0.04*S)+1.0)**0.7))/(1900.0*(Y**0.5))
    5 TTP=(DELT/2.)+TL
      D=0.25*TTP
      IF(DELT.LE.D)GO TO 9
      WRITE(6,6)DELT,D
      WRITE(7,6)DELT,D
    6 FORMAT(2X,'THE TIME STEP OF',F5.2,1X,
     *'HR IS GREATER THAN 0.25 TIME TO PEAK',/2x,' WHICH IS',F5.2,
     *1X,'HR',2X,'SO THE RESULTING HYDROGRAPH MAY BE JAGGED',/)
    9 CONTINUE
      WRITE(6,24)TL,TTP
      WRITE(7,24)TL,TTP
   24 FORMAT(2X,'WATERSHED LAG TIME =',F8.3,'HR',/,2X,'TIME TO PEAK=',
     *F8.3,'HR')
      DO 10 I=1,20
      IF(I.EQ.1)GO TO 11
      RATIOQ(I)=RATIOQ(I-1)+0.05
      GO TO 10
   11 RATIOQ(I)=0.05
   10 CONTINUE
      RATIOT(1)=.47
```

```
   RATIOT(2)=.60
   RATIOT(3)=.69
   RATIOT(4)=.78
   RATIOT(5)=.85
   RATIOT(6)=.92
   RATIOT(7)=.97
   RATIOT(8)=1.02
   RATIOT(9)=1.08
   RATIOT(10)=1.16
   RATIOT(11)=1.24
   RATIOT(12)=1.32
   RATIOT(13)=1.41
   RATIOT(14)=1.51
   RATIOT(15)=1.62
   RATIOT(16)=1.75
   RATIOT(17)=1.91
   RATIOT(18)=2.15
   RATIOT(19)=2.60
   RATIOT(20)=5.00
   NN=INT(5*TTP/DELT)
   XNN=5*TTP/DELT
   IF(XNN.GT.FLOAT(NN))NN=NN+1
   IF(NN.LE.N)GO TO 27
   NPLUS=N+1
   DO 28 I=NPLUS,NN
28 T(I)=T(I-1)+DELT
27 CONTINUE
   DO 12 I=1,NN
   TTTP=T(I)/TTP
   IF(TTTP.GE.5.)GO TO 15
   IFLAG=1
   DO 13 J=1,20
   IF(TTTP.LE.RATIOT(J))GO TO 13
   IFLAG=IFLAG+1
13 CONTINUE
   IF(IFLAG.GT.1)GO TO 14
   QQT(I)=(TTTP/RATIOT(1))*RATIOQ(1)
   GO TO 12
14 QQT(I)=((TTTP-RATIOT(IFLAG-1))/(RATIOT(IFLAG)-RATIOT(IFLAG-1)))
   QQT(I)=QQT(I)*(RATIOQ(IFLAG)-RATIOQ(IFLAG-1))
   QQT(I)=QQT(I)+RATIOQ(IFLAG-1)
   GO TO 12
15 QQT(I)=1.0
12 CONTINUE
   DO 16 I=1,NN
   IF(I.EQ.1)GO TO 25
   DELTA(I)=(QQT(I)-QQT(I-1))
   GO TO 16
25 DELTA(I)=QQT(I)
16 CONTINUE
   SUMDEL=0.0
   DO 21 I=1,NN
21 SUMDEL=SUMDEL+DELTA(I)
   DO 23 I=1,NN
23 DELTA(I)=DELTA(I)/SUMDEL
   WRITE(6,8)
   WRITE(7,8)
 8 FORMAT(/,10X,'UNIT HYDROGRAPH',/,3X,'TIME(HR)',10X,'DIMENSIONLESS
  *ORDINATES'/)
   DO 22 I=1,NN
   WRITE(6,17)T(I),DELTA(I)
   WRITE(7,17)T(I),DELTA(I)
17 FORMAT(1X,F10.3,17X,F10.3)
22 CONTINUE
   RETURN
   END
   SUBROUTINE ROUTE
   COMMON/A1/DELT,QU,N,P,TD,CN,II
   COMMON/A2/KT,TP,RP,WP,K
   COMMON/A4/T(250),R(250),RE(250),RER(250)
   COMMON/A8/AREA,L,Y,NN,NF
   COMMON/A9/DELTA(250),CNM
   COMMON/A10/DELP(250),QA(250),TM(250),QM(250)
   COMMON/A11/SUBPT1,SUBPT2,SUBPT3,SUBPT4
   COMMON/A12/TL
   DIMENSION DD(250)
   MM=NN+NF-1
```

```
      NPLUS=NF+1
      IF(NPLUS.LE.1)NPLUS=2
      IF(MM.LE.1)MM=3
      DO 10 I=NPLUS,MM
      TM(I)=TM(I-1)+DELT
   10 DELP(I)=0.0
      NPLUS=NN+1
      IF(MM.LT.150)GOTO 25
      MM=150
      NPLUS=149
   25 CONTINUE
      DO 11 I=NPLUS,MM
   11 DELTA(I)=0.0
      WRITE(6,17)
      WRITE(7,17)
   17 FORMAT(/,1X,'F L O O D  H Y D R O G R A P H  C O M P U T A T I O N
     *'/10X,' B Y  C N - I N F I L  M O D E L ',/)
      WRITE(6,20)
      WRITE(7,20)
   20 FORMAT(1X,65(1H*))
      WRITE(6,21)
      WRITE(7,21)
   21 FORMAT(/,2X,'T(HR) = TIME AT THE END OF EACH STEP'/
     *2X,'DD(MM/HR) = RATE OF RAINFALL EXCESS'/
     *2X,'DELP(MM) = INCREMENTAL DEPTH OF EXCESS RAINFALL'/
     *2X,'QA(M3/S) = DISCHARGES COMPUTED'/
     *2X,'QM(M3/S) = DISCHARGES OBSERVED (IF SO)'///)
      WRITE(6,40)
      WRITE(7,40)
   22 FORMAT(1X,F5.2,4X,F8.2,4X,F8.2,4X,F8.3,4X,F8.3)
   23 FORMAT(1X,F5.2,4X,F8.2,4X,F8.2,4X,F8.3)
   24 FORMAT(1X,60(1H-))
      WRITE(6,24)
      WRITE(7,24)
      IF(SUBPT3.EQ.0)GO TO 29
      IF(II.LT.MM)GO TO 29
      DO 27 I=II,MM
   27 QM(I)=QM(II)
   29 CONTINUE
      DO 14 I=1,MM
   14 DD(I)=DELP(I)/DELT
C     CONVOLUTION OPERATION
      DO 13 I=1,MM
      QA(I)=0.0
      DO 13 J=1,I
      QA(I)=QA(I)+DELP(J)*DELTA(I-J+1)
   13 CONTINUE
C     CONVERT FROM MM/HR TO M3/S
      DO 30 I=1,MM
      QA(I)=(AREA/3.6)*QA(I)/DELT
      QA(I)=QA(I)*CNM
   30 CONTINUE
      IF(SUBPT3.EQ.0)GO TO 31
   40 FORMAT(2X,'T(HR)',3X,'DD(MM/HR)',3X,'DELP(MM)',4X,'QA(M3/S)',4X,
     *'QM(M3/S)'/)
      DO 2 I=1,MM
    2 QM(I)=QM(I)-QM(1)
      DO 9 I=1,MM
      IJ=K+I
      QM(I)=QM(IJ)
      IF(QM(I).LE.0.0)QM(I)=0.0
      WRITE(6,22)TM(I),DD(I),DELP(I),QA(I),QM(I)
      WRITE(7,22)TM(I),DD(I),DELP(I),QA(I),QM(I)
    9 CONTINUE
      GO TO 42
   31 CONTINUE
      DO 8 I=1,MM
      WRITE(6,23)TM(I),DD(I),DELP(I),QA(I)
      WRITE(7,23)TM(I),DD(I),DELP(I),QA(I)
    8 CONTINUE
   42 CONTINUE
      RETURN
      END
      SUBROUTINE GODFI
      COMMON/A1/DELT,QU,N,P,TD,CN,II
      COMMON/A4/T(250),R(250),RE(250),RER(250)
      COMMON/A7/CUMP(250)
```

```
      COMMON/A8/AREA,L,Y,NN,NF
      COMMON/A10/DELP(250),QA(250),TM(250),QM(250)
      COMMON/A13/CNAME(20)
      INTEGER QU
      REAL L
C
C     EVALUATION OF RESULTS
C
      II=N
      SQ2=0.
      SQM2=0.
      SUMC=0.
      SUMQ=0.
      SMQ=0.
      DO 1 I=1,N
    1 SMQ=SMQ+QM(I)
      SMQ=SMQ/N
      DO 2 I=1,N
      SQ2=SQ2+(QM(I)-QA(I))**2
      SQM2=SQM2+(QM(I)-SMQ)**2
    2 CONTINUE
      RI=(SQM2-SQ2)/SQM2
      PE=(SQRT(SQ2/N))/SMQ
      F1=0.
      F2=0.
      QMAX=0.
      QAMAX=0.
      DO 3 I=1,N
      F2=QA(I)-QM(I)
      F2=ABS(F2)*QM(I)
      F1=F1+F2
    3 CONTINUE
      DO 4 I=1,N
      IF(QM(I).GT.QMAX)QMAX=QM(I)
      IF(QA(I). GT.QAMAX)QAMAX=QA(I)
      SUMQ=SUMQ+QM(I)
      SUMC=SUMC+QA(I)
    4 CONTINUE
      DEV=(F1*200.0)/(FLOAT(N)*(QMAX*QMAX))
      PEAK=(QMAX-QAMAX)/QMAX
      PEAK=PEAK*100.
      TVOL=(SUMQ-SUMC)/SUMQ
      TVOL=TVOL*100.
C     PRINT EVALUTION OF RESULTS
      WRITE(6,20)
   20 FORMAT(2X//)
      WRITE(6,21)
      WRITE(7,21)
   21 FORMAT(2X,'THE GOODNES OF FITTING CRITERIA',/2X,32(1H-)/)
   22 FORMAT(2X,'COEFFICIENT OF DETERMINATION RE=',F7.2/
     *2X,'COEFFICIENT OF VARIATION PE=',F7.2/
     *2X,'SCHULTZE HYDROL.DEVIATION DEV=',F7.2/
     *2X,'PEAK ERROR PEAK(PERC)=',F7.2/
     *2X,'TOTAL RUNOFF ERROR TVOL(PERC)=',F7.2//)
      WRITE(6,22)RI,PE,DEV,PEAK,TVOL
      WRITE(6,20)
      WRITE(7,22)RI,PE,DEV,PEAK,TVOL
      WRITE(7,20)
      RETURN
      END
```

KIN – data TŘEBSÍN

```
$DEBUG
$NOTRUNCATE
      PROGRAM CPLANEKS
C     PROGRAM PLANEK SERVES FOR DESIGN DISCHARGES COMPUTATION
C     IT IS A VERSION OF CPLANE WITH SUMMATION OF PARTIAL PLANES
C     AND SEGMENTS AS AN INPUT TO CSTREAM OR MUSK PROGRAMMES
C     NAME OF FILE: CPLANEKS.FOR
      COMMON/A1/DELT,PP,N,P,TD,CN,II,IT,IL,IK,TTM,TDR
      COMMON/A4/T(150),R(150),RE(150),RER(150),QAB(150)
      COMMON/A5/SO(10),DLN(10),WI(10),WID(10),AK1(10),ALPHA(10),AM(10)
      COMMON/A8/AREA,L,Y,NN,NF,TIM,DT
      COMMON/A9/FRNM(10),FRIC(10),DX(10),PBAC(10),AR(10),AL(10),PR(10)
      COMMON/A10/DELP(150),QA(150),TM(150),QM(150),TIND(150),QAA(150)
```

```
      COMMON/A12/H(11,150),VE(11,150),TAU(11,150),VSTR(11,150)
      COMMON/A13/CNAME(20)
      COMMON/A14/NPL,QAW(20,150),qcw(20,150)
      INTEGER PP,PPP,PM,FLAG,J,I,IT,IL,N,II,NPL,K
      REAL HKIN,HBAC,HCEN,HFOR,RO,GA,DEL,TIME,QL,AK11,AM1,PBAC1,PR1
      REAL PEFF,EFR,Q0
      DIMENSION QC(150)
      PARAMETER(PM=150)
      PARAMETER(PPP=10)
C     PARAMETER(PP=3)
      CHARACTER*12 NDFILE
C     CHARACTER*30 START
      CHARACTER*8 FNAME$
C
C     PURPOSE: TO SOLVE THE KINEMATIC WAVE EQUATION FOR OVERLAND FLOW
C              ON A CASCADE OF PLANES UNDER RAINFALL, USING THE SECOND
C              ORDER EXPLICIT FINITE-DIFFERENCE LAX-WENDROFF SCHEME.
C              THE BACKWARDS FINITE DIFFERENCE SCHEME IS USED AT THE
C              END OF EACH PLANE. THE VARIABLE WIDTHS OF PLANES CAN
C              ALSO BE USED FOR A SEGMENT FLOW SIMULATION.
C
C
      RO=1000.0
      GA=9.81
      K=0
      OPEN(UNIT=5,FILE='TREB9.DTA',STATUS='OLD')
      OPEN(UNIT=7,FILE='TREB9.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=1,FILE='NDFILE',STATUS='UNKNOWN')
C     INPUT OF DATA
C     READ INITIAL DISCHARGE (for design discharges obviously Q0=0.0)
   97 FORMAT(F10.4)
      READ(5,97)Q0
C     beginning of cycle data reading
  100 PRINT *,'GIVE THE NAME OF DATA FILE:      '
C     READ '(A12)',NDFILE
C     READ IN DATA FILE
   99 FORMAT(20A4)
      READ(5,99)CNAME
   98 FORMAT(2X,60(1H*))
C     READ NUMBER OF GEOMETRIC ELEMENTS (PLANES, SEGMENTS) TO BE SUMMED
      READ(5,105)NPL
C     READ THE NUMBER OF PLANES IN A CASCADE, PP
      READ(5,105)PP
C     READ THE PLANE PARAMETERS:
C     SLOPE SO(J), LENGTH DLN(J), WIDTH WID(J), OBSTACKLES AR(J),
C     MANNING ROUGHNESS FRNM(J), FRICTION FRIC(J), FLOW TYPE AM(J).
  101 FORMAT(7F10.4)
      DO 2 J=1,PP
      READ(5,101)SO(J),DLN(J),WID(J),AR(J),FRNM(J),FRIC(J),AM(J)
    2 CONTINUE
  102 FORMAT(2F10.2)
  103 FORMAT(10F8.3)
      READ(5,102)DELT,TTM
      RCO=3600.
  105 FORMAT(I3)
      READ(5,105)N
      READ(5,103)(RER(I),I=1,N)
      TDLN=0.
      SI=0.
      DO 17 I=1,N
      RER(I)=RER(I)/0.01665    ! 1 min
C     RER(I)=RER(I)/0.08325    ! 5 min
C     RER(I)=RER(I)/0.1665     ! 10 min
C     RER(I)=RER(I)/0.5        ! 30 min
      SI=SI+RER(I)
   17 CONTINUE
      SI=SI*(DELT/RCO)
      DO 18 J=1,PP
   18 TDLN=TDLN+DLN(J)
      PAR=(TDLN/1000.)*(WID(1)/1000.)
C     TRANSFER FROM MM/HR TO M/S
      RER(1)=RER(1)/RCO/1000.0
      T(1)=DELT
      DO 3 I=2,N
      RER(I)=RER(I)/RCO/1000.0
      T(I)=T(I-1)+DELT
    3 CONTINUE
```

```
  112 FORMAT(I1)
      READ(5,112)NEXT
C
    4 PRINT *,'GIVE TIME INCREMENT (DT):      '
      READ *,DT
C     PRINT *,'GIVE TIME INTERVAL FOR WRITING INTERMEDIATE RESULTS:    '
C     PRINT *,'RECOMMENDATION:TIM SHOULD EQUAL PREFERABLY TO DELT, '
C     PRINT *,'BUT NOT NECESSARILLY'
C     READ *,TIM
C
C     INITIAL CONDITIONS
      TIM=DELT
      TIME=TIM
      TTIM=30.0
      IL=11
      IK=0
      DO 6 J=1,PP
      FRNM(J)=FRNM(J)*1.0   !
      AL(J)=ATAN(SO(J))
      AK1(J)=SQRT(8.0*GA*SO(J)/FRIC(J))
      ALPHA(J)=SQRT(SO(J))/FRNM(J)
      DX(J)=DLN(J)/10.0
      FRNM(J)=FRNM(J)/1.0   !
      IF(AR(J).LE.1.0)THEN
      WI(J)=1.0-AR(J)
      ELSE
      WI(J)=1.0*AR(J)
      ENDIF
      DO 5 I=1,IL
      H(J,I)=0.0
    5 CONTINUE
    6 CONTINUE
      IT=0
C     WRITING INITIAL DATA
  113 FORMAT(2X,'NAME OF CATCHMENT:',2X,20A4/2X,60(1H*))
      WRITE(6,113)CNAME
      WRITE(7,113)CNAME
  107 FORMAT(2X,'OUTPUT OF SUBROUTINE CPLANE:'/2X,28(1H*))
C     WRITE(6,107)
C     WRITE(7,107)
  104 FORMAT(2X,'SIMULATION OF OVERLAND FLOW IN THE CASCADE OF PLANES'/
     *2X,'(EXPLICIT FINITE DIFFERENCE SCHEME OF LAX-WENDROFF):'/)
C
C     PRINTING DETAILED SUBHEADINGS
  106 FORMAT(2X,'INTERIM RESULTS OF SIMULATION:'/2X,30(1H-))
C     WRITE(6,104)
C     WRITE(7,104)
  108 FORMAT(2X,'NUMBER OF PLANES IN A CASCADE:',3X,I2)
  109 FORMAT(2X,'DESCRIPTION OF PLANES:'/2X,22(1H-)/2X,'SLOPES(-)',5X,
     *'LENGTHS(M)',5X,'WIDTHS(M)',5X,'MAN.ROUGHNESS',5X,'M-FLOWTYPE')
  110 FORMAT(4X,F6.3,5X,F9.3,6X,F9.3,8X,F7.4,10X,F7.4)
  111 FORMAT(2X,70(1H-))
C     WRITE(6,108)PP
C     WRITE(7,108)PP
      WRITE(6,109)
      WRITE(7,109)
      DO 16 J=1,PP
      WRITE(6,110)SO(J),DLN(J),WID(J),FRNM(J),AM(J)
      WRITE(7,110)SO(J),DLN(J),WID(J),FRNM(J),AM(J)
   16 CONTINUE
      WRITE(6,111)
      WRITE(7,111)
C     WRITE(6,106)
C     WRITE(7,106)
C
C     START OF SIMULATION
      DO 1 TD=DT,TTM,DT
      DO 7 I=1,N
           IF(TD.LE.T(1))THEN
             EFR=RER(1)
           ELSE IF(TD.GT.T(N))THEN
             EFR=0.0
           ELSE IF(TD.LE.T(I).AND.TD.GT.T(I-1))THEN
             EFR=RER(I)
             GO TO 7
           ENDIF
    7 CONTINUE
```

```
C     PREPARATION OF EFFECTIVE RAIN (PEFF) FOR EACH SLOPE
      DO 8 J=1,PP
      PEFF=EFR*WI(J)
      DDT=TD-DT
      IF(DDT.LT.0.001) PR(J)=PEFF
      PBAC(J)=PR(J)
      PR(J)=PEFF
    8 CONTINUE
C     START OF SPACE SIMULATION FOR DIFFERENT SLOPES
      DO 10 J=1,PP
C     RESET VALUES FOR THE NEXT TIME STEP
      I=2
      HBAC=H(J,I-1)
      HCEN=H(J,I)
      HFOR=H(J,I+1)
C     COMPUTATION PROCEDURE
      DO 9 I=2,IL
      FLAG=0
      IF(I.EQ.IL)FLAG=1
      DX1=DX(J)
      AK11=AK1(J)
      ALP=ALPHA(J)
      AM1=AM(J)
      PBAC1=PBAC(J)
      PR1=PR(J)
      H(J,I)=HKIN(HBAC,HCEN,HFOR,DT,DX1,ALP,AK11,AM1,PBAC1,PR1,FLAG)
      IF(H(J,I).LT.0.0)THEN
      H(J,I)=0.
      IF(H(PP,IL).LE.0.)THEN
      IT=INT(TD+0.01)
      CALL WRTR
      PRINT *,'------>COMPUTATIONAL BLOCK  '
      GOTO 20
      ENDIF
      CONTINUE
      ENDIF
C     RESET VALUES FOR CALCULATION ON THE NEXT GRIDPOINT
      IF(I.GT.1.AND.I.LT.IL-1)THEN
      HBAC=HCEN
      HCEN=HFOR
      HFOR=H(J,I+2)
      ENDIF
    9 CONTINUE
C     TEST OF STABILITY
      IF(H(J,IL).GT.0.)DEL=DX(J)/(AM(J)*ALPHA(J)*H(J,IL)**(AM(J)-1.0))
      IF(DT.GT.DEL.AND.DEL.GT.0.0)THEN
      PRINT *,'STABILITY CONDITIONS ARE NOT FULFILLED AT T  '
      ENDIF
      IF(J.NE.PP)THEN
      QL=ALPHA(J)*H(J,IL)**AM(J)*WI(J)*WID(J)
      H(J+1,1)=(QL/WI(J+1)/WID(J+1)/ALPHA(J+1))**(1.0/AM(J+1))
      ENDIF
   10 CONTINUE
C     TEST WHETHER TO WRITE INTERIM RESULTS
C
   11 IT=INT(TD+0.01)
      IF(ABS(IT-TIME).LT.0.01)THEN
        IF(H(PP,IL).GT.2E-6)THEN
        CALL WRTR
        ENDIF
      TIME=TIME+TIM
      ENDIF
C     END OF THE TIME CYCLE
    1 CONTINUE
      WRITE(6,'(A)')'   '
      WRITE(7,'(A)')'   '
   13 FORMAT(2X,'THE OVERLAND FLOW HYDROGRAPH: '/
     *       2X,'ORDIN.NO.',6X,'TIME(HOURS)',2X,'DISCHARGE Q (L/S)'//)
      WRITE(6,13)
      WRITE(7,13)
      DO 15 JK=1,IK
      QAB(JK)=QAA(JK)*1000.0
      WRITE(6,14)JK,TIND(JK),QAB(JK)
      WRITE(7,14)JK,TIND(JK),QAB(JK)
   14 FORMAT(2X,I3,8X,F10.3,8X,F10.3)
   15 CONTINUE
C     MASS BALANCE CHECK
```

```
      DELE=DELT/3600.
      SQ=0.
      DO 19 I=1,IK
   19 SQ=SQ+QAA(I)
      SQ=SQ*DELE*3.6/PAR
  114 FORMAT(2X,'THE MASS BALANCE CHECK:'/2X,25(1H-),/
     *2X,'TOTAL INFLOW DEPTH:',4X,F8.2,2X,'MM',/,
     *2X,'TOTAL OUTFLOW DEPTH:',3X,F8.2,2X,'MM',/////)
      WRITE(6,114)SI,SQ
      WRITE(7,114)SI,SQ
      GOTO 120
C     SUMMATION OF OUTFLOW DEPTH FROM ALL AREAS (PLANES +SEGMENTS)
      K=K+1
      DO 25 I=1,IK
      IF(K.EQ.1)THEN
      QAW(K,I)=QAA(I)
      ELSE
      QAW(K,I)=QAW(K-1,I)+QAA(I)
      ENDIF
   25 CONTINUE
      IF(NEXT.NE.0) GO TO 100
      WRITE(6,116)
      WRITE(7,116)
      WRITE(6,117)K,NPL
      WRITE(7,117)K,NPL
      DO 26 I=1,IK
      QAW(K,I)=Q0+QAW(K,I)
      WRITE(6,115)I,TIND(I),QAW(K,I)
      WRITE(7,115)I,TIND(I),QAW(K,I)
      QC(I)=QAW(K,I)
   26 CONTINUE
c     qaw(k,i) values are up to now summative, which means that
c     qaw(npl,i) values are equal to total values qc(i).
c     next procedure computes outflow qcw from partial ares k=1,npl.
      do 27 k=1,npl
      do 27 i=1,ik
      if(k.eq.1)then
      qcw(k,i)=qaw(1,i)
      else
      qcw(k,i)=qaw(k,i)-qaw(k-1,i)
      endif
   27 continue
  116 FORMAT(2X,'SUM OF THE HYDROGRAPH ORDINATES FROM ALL PARTIAL AREAS'
     */2X,60(1H*))
  115 FORMAT(2X,I3,6X,F10.3,6X,F10.3)
  117 FORMAT(2X,'K= ',I3,5X,'NPL= ',I3)
   20 CONTINUE
C     SCREEN CONTROL
      READ '(A12)',NDFILE
C     Data for graphs preparation
      write(*,'(A\)')' NAME OF EVENT DATA:'
      read(*,'(A)')fname$
      open(unit=8,file=fname$)
      write(8,'(2x,i3,2f8.3)')(i,tind(i),qaa(i),i=1,ik)
c     output data for all planes
c     READ '(A12)',NDFILE
c     write(*,'(A\)')' NAME OF DETAILED EVENT DATA:'
c     read(*,'(A)')fname$
c     Pozor na formatovani tabulky- max.pocet sloupcu =12(12f6.3)
c     write(8,'(1x,6f10.3)')((qcw(k,i),k=1,npl),i=1,ik)   !
  120 CONTINUE
      CLOSE (8)
      PRINT *,    '******END OF CASCADE OF PLANES******'
      if(next.eq.1)goto 100
   12 STOP
      END
      FUNCTION HKIN(HBAC,HCEN,HFOR,DT,DX1,ALP,AK11,AM1,PBAC1,PR1,FLAG)
      INTEGER FLAG
      REAL HN1,HN2,HN3,HN4,DT,DX1,ALP,AK11,AM1,PBAC1,PR1
      IF(FLAG.EQ.0)THEN
      HN1=DT/2.0/DX1*(ALP*HFOR**AM1-ALP*HBAC**AM1-2.0*DX1*PBAC1)
      HN2=(DT/DX1)**2.0/2.0*(ALP*AM1*HFOR**(AM1-1.)+ALP*AM1*HCEN**
     *(AM1-1.))*(ALP*HFOR**AM1-ALP*HCEN**AM1-DX1*PBAC1)
      HN3=(DT/DX1)**2.0/2.0*(ALP*AM1*HCEN**(AM1-1.)+ALP*AM1*HBAC**
     *(AM1-1.))*(ALP*HCEN**AM1-ALP*HBAC**AM1-DX1*PBAC1)
      HN4=DT/2.0*(PR1-PBAC1)
      HKIN=HCEN-HN1+HN2-HN3+HN4
```

```
      ELSE IF(FLAG.EQ.1)THEN
      HN1=DT/DX1*(ALP*HFOR**AM1-ALP*HCEN**AM1-DX1*PBAC1)
      HN2=(DT/DX1)**2.0/2.0*(ALP*AM1*HFOR**(AM1-1.)+ALP*AM1*HCEN**
     *(AM1-1.))*(ALP*HFOR**AM1-ALP*HCEN**AM1-DX1*PBAC1)
      HN3=(DT/DX1)**2.0/2.0*(ALP*AM1*HCEN**(AM1-1.)+ALP*AM1*HBAC**
     *(AM1-1.))*(ALP*HCEN**AM1-ALP*HBAC**AM1-DX1*PBAC1)
      HN4=DT/2.0*(PR1-PBAC1)
      HKIN=HFOR-HN1+HN2-HN3+HN4
      ENDIF
      AK11=ALP
      RETURN
      END
      SUBROUTINE WRTR
      COMMON/A1/DELT,PP,N,P,TD,CN,II,IT,IL,IK,TTM,TDR
      COMMON/A4/T(150),R(150),RE(150),RER(150),QAB(150)
      COMMON/A5/SO(10),DLN(10),WI(10),WID(10),AK1(10),ALPHA(10),AM(10)
      COMMON/A8/AREA,L,Y,NN,NF,TIM,DT
      COMMON/A9/FRNM(10),FRIC(10),DX(10),PBAC(10),AR(10),AL(10),PR(10)
      COMMON/A10/DELP(150),QA(150),TM(150),QM(150),TIND(150),QAA(150)
      COMMON/A12/H(11,150),VE(11,150),TAU(11,150),VSTR(11,150)
      INTEGER J,I,IL,IT,PM,PP
      REAL X1,TK1,TK2,TK3,TK4
      PARAMETER(PM=150)
C     PARAMETER(PP=3)
C     CHARACTER*12 NDFILE
C
C     WRITING RESULTS
C
C     LIMITS FOR CRITICAL SHEAR STRESS TK
      TK1=0.0076
      TK2=0.038
      TK3=0.19
      TK4=1.67
      IT=IT/60
   32 FORMAT(2X,'LIMITS FOR CRITICAL SHEAR STRESS TAUcr(Pa):  '/
     *2X,60(1H-))
      WRITE(6,32)
      WRITE(7,32)
   33 FORMAT(2X,F7.4,3X,F7.4,3X,F7.4,3X,F7.4/)
      WRITE(6,33)TK1,TK2,TK3,TK4
      WRITE(7,33)TK1,TK2,TK3,TK4
      DO 21 J=1,PP
      DO 22 I=1,IL
      VE(J,I)=ALPHA(J)*H(J,I)**(AM(J)-1.)
      TAU(J,I)=9810.0*SO(J)*H(J,I)
      X1=9.81*SO(J)*H(J,I)
      VSTR(J,I)=SQRT(X1)
   22 CONTINUE
      QA(J)=ALPHA(J)*H(J,IL)**AM(J)*WI(J)*WID(J)
C     PRINT INTERIM RESULTS
C     DISCHARGE AT L IN L/S
   23 FORMAT(2X,'J=' ,I2,'   T=  ',I5,/
     *2X,'   H(J,I): ',6F8.4,/
     *2X,'  VE(J,I): ',6F8.4)
   34 FORMAT(2X,' TAU(J,I): ',6F8.4/
     *2X,'VSTR(J,I): ',6F8.4/2X,60(1H-))
      WRITE(6,'(2X,A,I2,3X,A,I4)') 'PLANE  ',J,  'AT TIME  ',IT
      WRITE(7,'(2X,A,I2,3X,A,I4)') 'PLANE  ',J,  'AT TIME  ',IT
      WRITE(6,23)J,IT,(H(J,I),I=1,11,2),(VE(J,I),I=1,11,2)
      WRITE(7,23)J,IT,(H(J,I),I=1,11,2),(VE(J,I),I=1,11,2)
      WRITE(6,34)(TAU(J,I),I=1,11,2),(VSTR(J,I),I=1,11,2)
      WRITE(7,34)(TAU(J,I),I=1,11,2),(VSTR(J,I),I=1,11,2)
   21 CONTINUE
C     WRITE(6,'(I5,10F6.3)')IT,(QA(J),J=1,PP)
C     WRITE(7,'(A)')'        '
C     WRITE(7,'(I5,10F6.3)')IT,(QA(J),J=1,PP)
C     WRITE(6,27)
C     WRITE(7,27)
   27 FORMAT(2X,60(1H-))
C     WRITE(6,'(A)')'  '
C     WRITE(7,'(A)')'  '
   29 SIT=0.0
      DO 30 J=1,PP
      IF(J.EQ.PP) THEN
      JK=INT(J/PP)
      IK=IK+JK
      QAA(IK)=QA(J)
```

```
      SIT=SIT+IT
      TIND(IK)=SIT/60.0
      WRITE(6,24)IK,IT,TIND(IK),QAA(IK)
      WRITE(7,24)IK,IT,TIND(IK),QAA(IK)
   24 FORMAT(3X,I4,9X,I4,6X,F7.2,7X,F7.3)
      ENDIF
   30 CONTINUE
      WRITE(6,27)
      WRITE(7,27)
   31 RETURN
      END
```