Project 3

CHUNG,Chak Pong

20015116

Problem

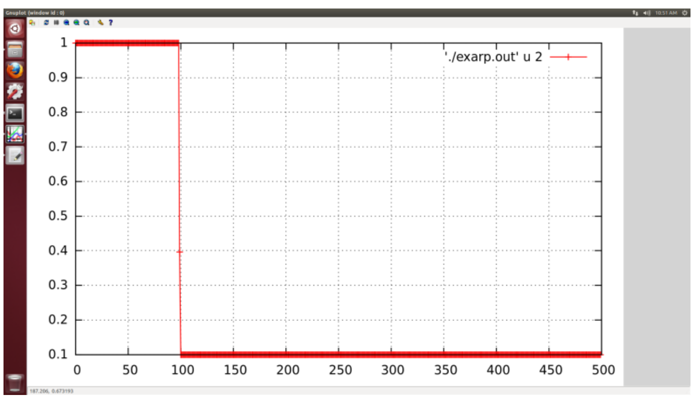
One dimentional shallow water equation with exact Riemann Solver

with Dam-Break Initial State UL=UR=0,HL=1,HR=0.1

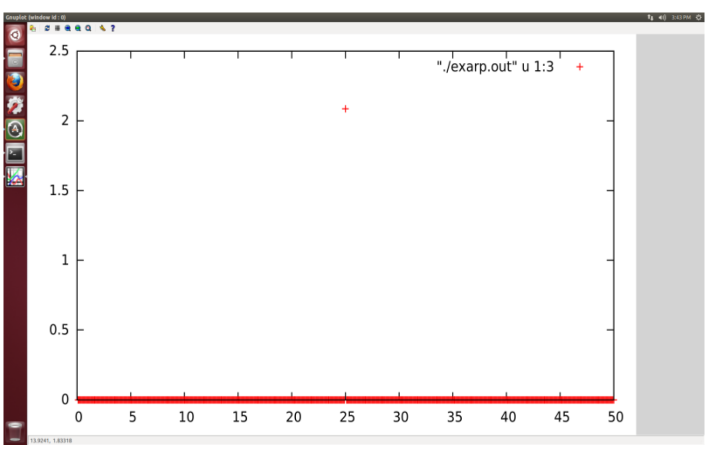
$$\begin{bmatrix} h \\ hu \end{bmatrix}_t + \begin{bmatrix} uh \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix}_x = 0.$$

Initial Condition

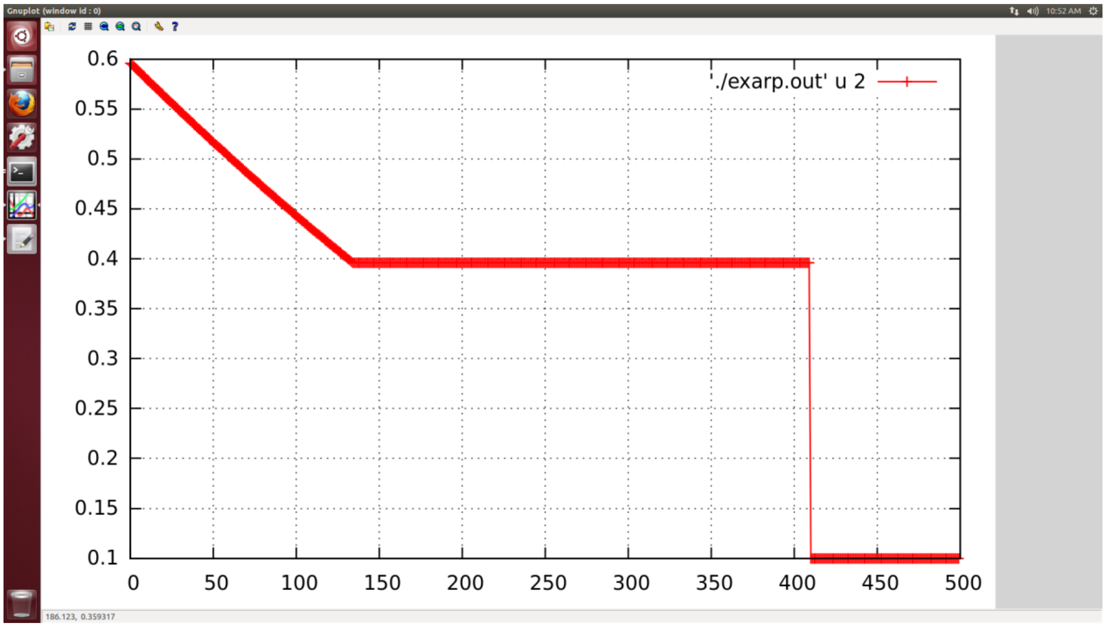
Initial H



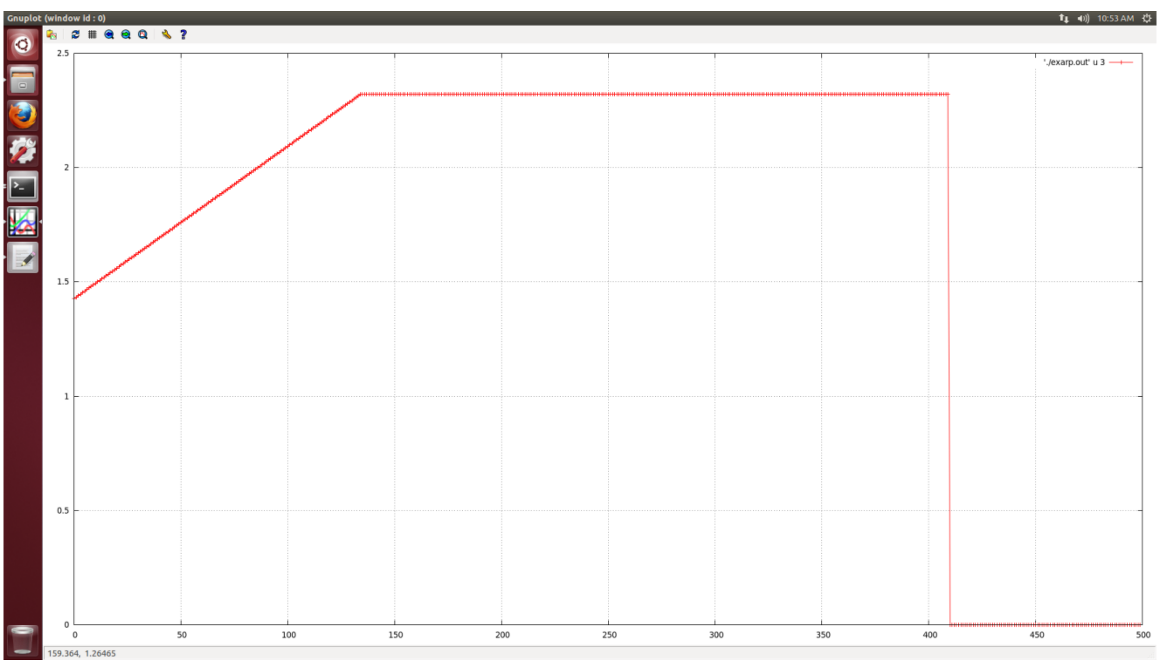
Initial HU



H at 10s



HU at 10s



Key steps in determing the Flux at star region

- Find $h_\star$ such that $\Phi(h_\star) = \Phi_r(h_\star) - \Phi_\ell(h_\star) = 0$, where

$$\Phi_\ell(h_\star) := \begin{cases} u_\ell - (h_\star - h_\ell)\sqrt{g\left(\frac{1}{2h_\star} + \frac{1}{2h_\ell}\right)} & \text{if} \quad h_\star > h_\ell \\ u_\ell + 2\left(\sqrt{gh_\ell} - \sqrt{gh_\star}\right) & \text{if} \quad h_\star \le h_\ell \end{cases}$$

$$\Phi_r(h_\star) := \begin{cases} u_r + (h_\star - h_r)\sqrt{g\left(\frac{1}{2h_\star} + \frac{1}{2h_r}\right)} & \text{if} \quad h_\star > h_r \\ u_r - 2\left(\sqrt{gh_r} - \sqrt{gh_\star}\right) & \text{if} \quad h_\star \le h_r \end{cases}$$

- Newton iteration: $h_\star^{k+1} = h_\star^k - \frac{\Phi(h_\star^k)}{\Phi'(h_\star^k)}$

u* is determined by

US = 0.5*(UL + UR) + 0.5*(FR - FL)

A more detailed derivation can be found in Toro 's book *Shock-Capturing Methods for Free-Surface Shallow Flows*, section 5.3.,from equation 5.5 to equation 5.12

The code used is attached.

```
*
      IMPLICIT NONE
*
C     Declaration of variables
*
      REAL    CHALEN,CL,CR,DCRIT,DL,DR,GATE,GRAVIT,TIMOUT,
     &        TOL,UL,UR
*
      INTEGER MCELLS, NITER
*
      COMMON /STATES/ CL, DL, UL, CR, DR, UR
      COMMON /ACCELE/ GRAVIT
      COMMON /TOLERA/ NITER, TOL
      COMMON /DOMAIN/ CHALEN, GATE, MCELLS, TIMOUT
*
C     Initial data and computational parameters are read in
*
      OPEN(UNIT=1,FILE='exarp.ini',STATUS='UNKNOWN')
*
      READ(1,*)CHALEN    ! length of channel
      READ(1,*)GATE      ! position of gate
      READ(1,*)GRAVIT    ! acceleration due to gravity
      READ(1,*)MCELLS    ! number of cells in profile
      READ(1,*)TOL       ! tolerance for convergence test
      READ(1,*)NITER     ! iterations in exact solver
      READ(1,*)TIMOUT    ! output time
      READ(1,*)DL        ! depth on left reservoir
      READ(1,*)UL        ! velocity in left reservoir
      READ(1,*)DR        ! depth in right reservoir
      READ(1,*)UR        ! velocity in right reservoir
*
      CLOSE(1)
*
C     Compute celerity on left and right states
*
      CL = SQRT(GRAVIT*DL)
      CR = SQRT(GRAVIT*DR)
*
C     Use the "depth positivity condition" to identify
C     type of data and thus of solution and to call
C     appropriate exact solver
*
      DCRIT = (UR-UL) - 2.0*(CL+CR)
*
      IF(DL.LE.0.0.OR.DR.LE.0.0.OR.DCRIT.GE.0.0)THEN
*
C        Dry bed cases
*
         CALL DRYBED
*
      ELSE
*
C        Wet bed case
*
         CALL WETBED
*
```

```
      ENDIF
*
C     Results are printed out
*
      CALL OUTPUT
*
      END
*
                                    *
*----------------------------------------------------------*
*
      SUBROUTINE OUTPUT
*
C     Purpose: to output exact solution at chosen
C              output time TIMOUT
*
      IMPLICIT NONE
*
C     Declaration of variables
*
      INTEGER  MX, I, MCELLS
*
      REAL     D, U,CHALEN,GATE,TIMOUT,XCOORD
*
      PARAMETER (MX = 3000)
*
      DIMENSION D(MX), U(MX)
*
      COMMON /SOLUTI/ D, U
      COMMON /DOMAIN/ CHALEN, GATE, MCELLS, TIMOUT
*
      OPEN(UNIT=1,FILE='exarp.out',STATUS='UNKNOWN')
*
      DO 10 I = 1, MCELLS
         XCOORD = REAL(I)*CHALEN/REAL(MCELLS)
         WRITE(1,20)XCOORD, D(I), U(I)
 10   CONTINUE
*
 20   FORMAT(3(F10.5,4X))
*
      CLOSE(1)
*
      END
*
                                    *
*----------------------------------------------------------*
*
      SUBROUTINE WETBED
*
C     Purpose: to solve the Riemann problem exactly for
C              the wet-bed case
*
      IMPLICIT NONE
*
C     Declaration of variables
*
      INTEGER  I,IT,MCELLS,MX,NITER
```

```
*
      REAL      CHA,CHALEN,CL,CR,CS,D,D0,DL,DR,DS,DSAM,FL,
     &          FLD,FR,FRD,GATE,GRAVIT,S,TIMOUT,TOL,U,UL,
     &          UR,US,USAM,XCOORD
*
      PARAMETER (MX = 3000)
*
      DIMENSION D(MX), U(MX)
*
      COMMON /SOLUTI/ D, U
      COMMON /STATES/ CL, DL, UL, CR, DR, UR
      COMMON /STARSO/ CS, DS, US
      COMMON /ACCELE/ GRAVIT
      COMMON /TOLERA/ NITER, TOL
      COMMON /DOMAIN/ CHALEN, GATE, MCELLS, TIMOUT
*
C     Find starting value for iteration
*
      WRITE(6,*)
      WRITE(6,*)'Exact Solution in Star Region'
      WRITE(6,*)'=============================='
      WRITE(6,*)
*
      CALL STARTE
*
C     Store starting value in D0
*
      D0 = DS
*
C     Start iteration
*
      WRITE(6,*)'   IT   ','    DS         ', '    CHA '
      WRITE(6,*)
      DO 10 IT = 1, NITER
*
         CALL GEOFUN(FL,FLD,DS,DL,CL)
         CALL GEOFUN(FR,FRD,DS,DR,CR)
         DS  = DS - (FL + FR + UR-UL)/(FLD + FRD)
         CHA = ABS(DS-D0)/(0.5*(DS+D0))
         WRITE(6,30)IT,DS,CHA
         IF(CHA.LE.TOL)GOTO 20
         IF(DS.LT.0.0)DS = TOL
         D0 = DS
*
 10   CONTINUE
*
      WRITE(6,*)'Number of NITER iterations exceeded,
     &           STOP'
*
      STOP
*
 20   CONTINUE
 30   FORMAT(I6,2X,2(F12.7,2X))
*
C     Converged solution for depth DS in Star Region.
C     Compute velocity US in Star Region
*
```

```
      US = 0.5*(UL + UR) + 0.5*(FR - FL)
*
      WRITE(6,*)
      WRITE(6,*)'Depth in Star Region    h* =',DS
      WRITE(6,*)'Velocity in Star Region u* =',US
      WRITE(6,*)
*
      CS = SQRT(GRAVIT*DS)
*
C     Evaluate exact solution at time TIMOUT
*
      DO 40 I = 1, MCELLS
*
         XCOORD = REAL(I)*CHALEN/REAL(MCELLS) - GATE
         S      = XCOORD/TIMOUT
*
C        Sample solution throughout wave structure at
C        time TIMOUT
*
         CALL SAMWET(DSAM,USAM,S)
*
C        Store solution
*
         D(I) = DSAM
         U(I) = USAM
*
 40   CONTINUE
*
      END
*
                                    *
*----------------------------------------------------------*
*
      SUBROUTINE GEOFUN(F,FD,D,DK,CK)
*
C     Purpose: to evaluate functions FL, FR and their
C              derivatives in iterative Riemann solver,
C              for wet-bed case.
*
      IMPLICIT NONE
*
C     Declaration of variables
*
      REAL   C,CK,D,DK,F,FD,GES,GRAVIT
*
      COMMON /ACCELE/ GRAVIT
*
      IF(D.LE.DK)THEN
*
C        Wave is rarefaction wave (or depression)
*
         C  = SQRT(GRAVIT*D)
         F  = 2.0*(C-CK)
         FD = GRAVIT/C
      ELSE
*
C        Wave is shock wave (or bore)
```

```
*
         GES = SQRT(0.5*GRAVIT*(D+DK)/(D*DK))
         F   = (D-DK)*GES
         FD  = GES - 0.25*GRAVIT*(D-DK)/(GES*D*D)
      ENDIF
*
      END
*
                                   *
*----------------------------------------------------------*
*
      SUBROUTINE STARTE
*
C     Purpose: to provide starting value for Newton-Raphson
C              iteration. The Two-Rarefaction Riemann
C              Solver (TRRS) and Two-Shock Riemann Solver
C              (TSRS) are used adaptively
*
      IMPLICIT NONE
*
C     Declaration of variables
*
      REAL     CL,CR,CS,DL,DMIN,DR,DS,GEL,GER, GRAVIT,
     &         UL,UR,US
*
      COMMON /STATES/ CL, DL, UL, CR, DR, UR
      COMMON /STARSO/ CS, DS, US
      COMMON /ACCELE/ GRAVIT
*
      DMIN = MIN(DL,DR)
*
C     Use Two-Rarefaction (TRRS) solution as starting value
*
      DS = (1.0/GRAVIT)*(0.5*(CL+CR)-0.25*(UR-UL))**2
*
      IF(DS.LE.DMIN)THEN
*
C        Use Two-Rarefaction (TSRS) approximation as
C        starting value
*
         WRITE(6,*)'TR approximation, h* =',DS
      ELSE
*
C        Use two-shock (TSRS) solution as starting value
C        with DS as computed from TRRS as estimate
*
         WRITE(6,*)'TS approximation, h* =',DS
*
         GEL = SQRT(0.5*GRAVIT*(DS+DL)/(DS*DL))
         GER = SQRT(0.5*GRAVIT*(DS+DR)/(DS*DR))
         DS  = (GEL*DL + GER*DR - (UR-UL))/(GEL + GER)
*
      ENDIF
      WRITE(6,*)
*
      END
*
```

```
                              *
*----------------------------------------------------------*
*
      SUBROUTINE SAMWET(D,U,S)
*
C     Purpose: to sample solution through wave structure at
C              TIMOUT for wet-bed case
*
      IMPLICIT NONE
*
C     Declaration of variables
*
      REAL    C,CL,CR,CS,D,DL,DR,DS,GRAVIT,QL,QR,S,SHL,
     &        SHR,SL,SR,STL,STR,U,UL,UR,US
*
      COMMON /STATES/ CL, DL, UL, CR, DR, UR
      COMMON /STARSO/ CS, DS, US
      COMMON /ACCELE/ GRAVIT
*
      IF(S.LE.US)THEN
********************************************
C        Sample left wave
********************************************
         IF(DS.GE.DL)THEN
*
C           Left shock
*
            QL = SQRT((DS + DL)*DS/(2.0*DL*DL))
            SL = UL - CL*QL
*
            IF(S.LE.SL)THEN
*
C              Sample point lies to the left of the shock
*
               D = DL
               U = UL
            ELSE
*
C              Sample point lies to the right of the shock
*
               D = DS
               U = US
            ENDIF
         ELSE
*
C           Left rarefaction
*
            SHL = UL - CL
*
            IF(S.LE.SHL)THEN
*
C              Sample point lies to the right of the
C              rarefaction
*
               D = DL
               U = UL
            ELSE
```

```
*
               STL = US - CS
*
               IF(S.LE.STL)THEN
*
C                 Sample point lies inside the rarefaction
*
                  U = (UL + 2.0*CL + 2.0*S)/3.0
                  C = (UL + 2.0*CL - S)/3.0
                  D = C*C/GRAVIT
               ELSE
*
C                 Sample point lies in the STAR region
*
                  D = DS
                  U = US
               ENDIF
            ENDIF
         ENDIF
*
      ELSE
*********************************************
C        Sample right wave
*********************************************
*
         IF(DS.GE.DR)THEN
*
C           Right shock
*
            QR = SQRT((DS + DR)*DS/(2.0*DR*DR))
            SR = UR + CR*QR
*
            IF(S.GE.SR)THEN
*
C              Sample point lies to the right of the shock
*
               D = DR
               U = UR
            ELSE
*
C              Sample point lies to the left of the shock
*
               D = DS
               U = US
            ENDIF
*
         ELSE
*
C           Right rarefaction
*
            SHR = UR + CR
*
            IF(S.GE.SHR)THEN
*
C              Sample point lies to the right of the
C              rarefaction
*
```

```
                D = DR
                U = UR
             ELSE
*
                STR = US + CS
*
                IF(S.GE.STR)THEN
*
C                  Sample point lies inside the rarefaction
*
                   U = (UR  - 2.0*CR + 2.0*S)/3.0
                   C = (-UR + 2.0*CR + S)/3.0
                   D = C*C/GRAVIT
                ELSE
*
C                  Sample point lies in the STAR region
*
                   D = DS
                   U = US
                ENDIF
             ENDIF
          ENDIF
       ENDIF
*
      END
*
                                  *
*----------------------------------------------------------*
*
      SUBROUTINE DRYBED
*
C     Pupose: to compute the exact solution in the case
C             in which a portion of dry bed is present
*
      IMPLICIT NONE
*
C     Declaration of variables
*
      INTEGER I,MCELLS,MX
*
      REAL    CHALEN,CL,CR,D,DL,DR,DSAM,GATE,S,TIMOUT,
     &        U,UL,UR,USAM,XCOORD
*
      PARAMETER (MX = 3000)
*
      DIMENSION D(MX), U(MX)
*
      COMMON /SOLUTI/ D, U
      COMMON /STATES/ CL, DL, UL, CR, DR, UR
      COMMON /DOMAIN/ CHALEN, GATE, MCELLS, TIMOUT
*
      DO 10 I = 1, MCELLS
*
         XCOORD = REAL(I)*CHALEN/REAL(MCELLS) - GATE
         S      = XCOORD/TIMOUT
C
         IF(DL.LE.0.0)THEN
```

```
*
C           Left state is dry
*
            CALL SAMLEF(DSAM,USAM,S)
         ELSE
            IF(DR.LE.0.0)THEN
*
C              Right state is dry
*
               CALL SAMRIG(DSAM,USAM,S)
            ELSE
*
C              Middle state is dry
*
               CALL SAMMID(DSAM,USAM,S)
            ENDIF
         ENDIF
*
         D(I) = DSAM
         U(I) = USAM
*
 10   CONTINUE
*
      END
*
                                   *
*----------------------------------------------------------*
*
      SUBROUTINE SAMLEF(D,U,S)
*
C     Purpose: to sample the solution through the wave
C              structure at time TIMOUT, for the case in
C              which the left state is dry. Solution
C              consists of single right rarefaction
*
      IMPLICIT NONE
*
C     Declaration of variables
*
      REAL   C,CL,CR,D,DL,DR,GRAVIT,S,SHR,STR,U,UL,UR
*
      COMMON /STATES/ CL, DL, UL, CR, DR, UR
      COMMON /ACCELE/ GRAVIT
*
      SHR = UR + CR
*
      IF(S.GE.SHR)THEN
*
C        Sampling point lies to the right of the
C        rarefaction
*
         D = DR
         U = UR
      ELSE
*
         STR = UR-2.0*CR
*
```

```fortran
         IF(S.GE.STR)THEN
*
C           Sampling point lies inside the rarefaction
*
            U = ( UR - 2.0*CR + 2.0*S)/3.0
            C = (-UR + 2.0*CR + S)/3.0
            D = C*C/GRAVIT
         ELSE
*
C           Sampling point lies in dry-bed state
*
            D = DL
            U = UL
         ENDIF
      ENDIF
*
      END
*
                                   *
*----------------------------------------------------------*
*
      SUBROUTINE SAMMID(D,U,S)
*
C     Purpose: to sample the solution through the wave
C              structure at time TIMOUT, for the case in
C              which the middle state is dry. Solution
C              consists of a left and a right rarefaction
C              with a dry portion in the the middle
*
      IMPLICIT NONE
*
C     Declaration of variables
*
      REAL    C,CL,CR,D,DL,DR,GRAVIT,S,SHL,SHR,SSL,SSR,
     &        U,UL,UR
*
      COMMON /STATES/ CL, DL, UL, CR, DR, UR
      COMMON /ACCELE/ GRAVIT
*
C     Compute wave speeds
*
      SHL = UL - CL
      SSL = UL + 2.0*CL
      SSR = UR - 2.0*CR
      SHR = UR + CR
*
      IF(S.LE.SHL)THEN
*
C        Sampling point lies to the left of the left
C        rarefaction
*
         D = DL
         U = UL
      ENDIF
*
      IF(S.GT.SHL.AND.S.LE.SSL)THEN
*
```

```
C        Sampling point lies inside the left rarefaction
*
         U = (UL + 2.0*CL + 2.0*S)/3.0
         C = (UL + 2.0*CL - S)/3.0
         D = C*C/GRAVIT
      ENDIF
*
      IF(S.GT.SSL.AND.S.LE.SSR)THEN
*
C        Sampling point lies inside the middle dry bed region
*
         D = 0.0
         U = 0.0
      ENDIF
*
      IF(S.GT.SSR.AND.S.LE.SHR)THEN
*
C        Sampling point lies inside the right rarefaction
*
         U = ( UR - 2.0*CR + 2.0*S)/3.0
         C = (-UR + 2.0*CR + S)/3.0
         D = C*C/GRAVIT
      ENDIF
*
      IF(S.GT.SHR)THEN
*
C        Sampling point lies to the right of the right
C        rarefaction
*
         D = DR
         U = UR
      ENDIF
*
      END
*
                                   *
*----------------------------------------------------------*
*
      SUBROUTINE SAMRIG(D,U,S)
*
C     Purpose: to sample the solution through the wave
C              structure at time TIMOUT, for the case in
C              which the right state is dry. Solution
C              consists of single left rarefaction
*
      IMPLICIT NONE
*
C     Declaration of variables
*
      REAL   C,CL,CR,D,DL,DR,GRAVIT,S,SHL,STL,U,UL,UR
*
      COMMON /STATES/ CL, DL, UL, CR, DR, UR
      COMMON /ACCELE/ GRAVIT
*
      SHL = UL - CL
*
      IF(S.LE.SHL)THEN
```

```
*
C         Sampling point lies to the left of the rarefaction
*
          D = DL
          U = UL
       ELSE
*
          STL = UL + 2.0*CL
*
          IF(S.LE.STL)THEN
*
C            Sampling point lies inside the rarefaction
*
             U = (UL + 2.0*CL + 2.0*S)/3.0
             C = (UL + 2.0*CL - S)/3.0
             D = C*C/GRAVIT
          ELSE
*
C            Sampling point lies in right dry-bed state
*
             D = DR
             U = UR
          ENDIF
       ENDIF
*
       END
*
                                     *
*----------------------------------------------------------*
*

SUB
```