

```
SUBROUTINE ASSIGN_A_JR_JC_CCD(N , Dx , C_X , U , Nz , AA , JR , JC , Src)
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!Routine assign CCD coeff.
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
IMPLICIT NONE
```

```
INTEGER,PARAMETER :: DP = SELECTED_REAL_KIND(p=15)
```

```
INTEGER,INTENT(IN) :: N
```

```
INTEGER,INTENT(OUT) :: Nz , JR(:) , JC(:)
```

```
REAL(KIND=DP),INTENT(IN) :: Dx , C_X(:) , U(:)
```

```
REAL(KIND=DP),INTENT(OUT) :: AA(:) , Src(:)
```

```
INTEGER :: I
```

```
REAL(KIND=DP) :: A1 , B1 , B2 , B3 , C1 , C2 , C3
```

```
REAL(KIND=DP) :: AA1 , AA3 , BB1 , BB3 , CC1 , CC2 , CC3
```

```
REAL(KIND=DP) :: A3D , B1D , B2D , B3D , C1D , C2D , C3D
```

```
! !For ux eq., a>0
```

```
! A1 = 0.875_DP
```

```
! B1 = 0.12512314207240799718_DP
```

```
! B2 = -0.24876857927592002816_DP
```

```
! B3 = 0.00012314207240799718364_DP
```

```
! C1 = -1.9360222951311040338_DP
```

```
! C2 = 1.9970445902622080676_DP
```

```
! C3 = -0.061022295131104033796_DP
```

```
! !For ux eq., a<0
```

```
! A3D = 0.875_DP
```

```
! B1D = -0.00012314207240799718364_DP
```

```
! B2D = 0.24876857927592002816_DP
```

```
! B3D = -0.12512314207240799718_DP
```

```
! C1D = 0.061022295131104033796_DP
```

```
! C2D = -1.9970445902622080676_DP
```

```
! C3D = 1.9360222951311040338_DP
```

```
!For ux eq., a>0
```

```
A1 = 0.875_DP
```

```
B1 = 0.12547425010813624084_DP
```

```
B2 = -0.24525749891863759159_DP
```

```
B3 = 0.00047425010813624083774_DP
```

```
C1 = -1.9318089987023651093_DP
```

```
C2 = 1.9886179974047302197_DP
```

```
C3 = -0.056808998702365109925_DP
```

```
!For ux eq., a<0
```

```
A3D = 0.875_DP
```

```
B1D = -0.00047425010813624083774_DP
```

```
B2D = 0.24525749891863759159_DP
```

```
B3D = -0.12547425010813624084_DP
```

```
C1D = 0.056808998702365109925_DP
```

```
C2D = -1.9886179974047302197_DP
```

```
C3D = 1.9318089987023651093_DP
```

```
!For uxx eq.
```

```
AA1 = -9.0_DP / 8.0_DP
```

```
AA3 = 9.0_DP / 8.0_DP
```

```
BB1 = -1.0_DP / 8.0_DP
```

```
BB3 = -1.0_DP / 8.0_DP
```

```
CC1 = 3.0_DP
```

```

CC2 = -6.0_DP
CC3 = 3.0_DP

```

```

Nz = 0

```

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

```

!For U_X EQ.

```

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

```

!For I = 1

```

```

!U_X(I)

```

```

Nz = Nz+1

```

```

AA(Nz) = 1.0_DP

```

```

JR(Nz) = 1

```

```

JC(Nz) = 1

```

```

!U_X(I+1)

```

```

Nz = Nz+1

```

```

AA(Nz) = 2.0_DP

```

```

JR(Nz) = 1

```

```

JC(Nz) = 2

```

```

!U_XX(I+1)

```

```

Nz = Nz+1

```

```

AA(Nz) = -Dx

```

```

JR(Nz) = 1

```

```

JC(Nz) = N + 2

```

```

Src(1) = (-3.5_DP * U(1) + 4.0_DP * U(2) - 0.5_DP * U(3)) / Dx

```

```

!For I = 2 , Nx-1,

```

```

DO I = 2 , N-1

```

```

* IF(C_X(I) >= 0.0_DP) THEN

```

```

!U_X(I-1)

```

```

Nz = Nz + 1

```

```

AA(Nz) = A1

```

```

JR(Nz) = I

```

```

JC(Nz) = I-1

```

```

!U_X(I)

```

```

Nz = Nz + 1

```

```

AA(Nz) = 1.0_DP

```

```

JR(Nz) = I

```

```

JC(Nz) = I

```

```

!U_XX(I-1)

```

```

Nz = Nz + 1

```

```

AA(Nz) = B1*Dx

```

```

JR(Nz) = I

```

```

JC(Nz) = N + I -1

```

```

!U_XX(I)

```

```

Nz = Nz + 1

```

```

AA(Nz) = B2*Dx

```

```

JR(Nz) = I

```

```

JC(Nz) = N + I

```

```

!U_XX(I+1)

```

```

Nz = Nz + 1

```

```

AA(Nz) = B3*Dx

```

```

JR(Nz) = I

```

```

JC(Nz) = N + I + 1

```

```

Src(I) = (C1*U(I-1) + C2*U(I) + C3*U(I+1)) / Dx

```

```

ELSE

```

```

!U_X(I)

```

```

Nz = Nz + 1

```

```
AA(Nz) = 1.0_DP
```

```
JR(Nz) = I
```

```
JC(Nz) = I
```

```
!U_X(I+1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = A3D
```

```
JR(Nz) = I
```

```
JC(Nz) = I+1
```

```
!U_XX(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = B1D*Dx
```

```
JR(Nz) = I
```

```
JC(Nz) = N + I - 1
```

```
!U_XX(I)
```

```
Nz = Nz + 1
```

```
AA(Nz) = B2D*Dx
```

```
JR(Nz) = I
```

```
JC(Nz) = N + I
```

```
!U_XX(I+1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = B3D*Dx
```

```
JR(Nz) = I
```

```
JC(Nz) = N + I + 1
```

```
Src(I) = (C1D*U(I-1) + C2D*U(I) + C3D*U(I+1)) / Dx
```

```
END IF
```

```
END DO
```

```
!For I = N
```

```
!U_X(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = 2.0_DP
```

```
JR(Nz) = N
```

```
JC(Nz) = N-1
```

```
!U_X(I)
```

```
Nz = Nz + 1
```

```
AA(Nz) = 1.0_DP
```

```
JR(Nz) = N
```

```
JC(Nz) = N
```

```
!U_XX(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = Dx
```

```
JR(Nz) = N
```

```
JC(Nz) = N + N-1
```

```
Src(N) = -(-3.5_DP * U(N) + 4.0_DP * U(N-1) - 0.5_DP * U(N-2)) / Dx
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!For U_XX EQ.
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!For I = 1
```

```
!3rd order
```

```
! !U_X(I+1)
```

```
! Nz = Nz+1
```

```
! AA(Nz) = -6.0_DP / Dx
```

```
! JR(Nz) = N + 1
```

```
! JC(Nz) = 2
```

```

!
!
! !U_XX(I)
! Nz = Nz+1
! AA(Nz) = 1.0_DP
! JR(Nz) = N + 1
! JC(Nz) = N + 1
!
! !U_XX(I+1)
! Nz = Nz+1
! AA(Nz) = 5.0_DP
! JR(Nz) = N + 1
! JC(Nz) = N + 2
!
! Src(N+1) = (9.0_DP * U(1) - 12.0_DP * U(2) + 3.0_DP * U(3)) / (Dx**2)

```

```

!4th order
!U_X(I+1)
Nz = Nz+1
AA(Nz) = -2.5_DP / Dx
JR(Nz) = N + 1
JC(Nz) = 2

```

```

!U_XX(I)
Nz = Nz+1
AA(Nz) = 1.0_DP
JR(Nz) = N + 1
JC(Nz) = N + 1

```

```

!U_XX(I+1)
Nz = Nz+1
AA(Nz) = 8.5_DP
JR(Nz) = N + 1
JC(Nz) = N + 2

```

```

Src(N+1) = ( 34.0_DP / 3.0_DP * U(1) - 83.0_DP / 4.0_DP * U(2) &
& +10.0_DP * U(3) -7.0_DP/12.0_DP * U(4) ) / (Dx**2)

```

```

!For I = 2 , N-1
DO I = 2 , N-1
!U_X(I-1)
Nz = Nz + 1
AA(Nz) = AA1 / Dx
JR(Nz) = N + I
JC(Nz) = I-1

```

```

!U_X(I+1)
Nz = Nz + 1
AA(Nz) = AA3 / Dx
JR(Nz) = N + I
JC(Nz) = I+1

```

```

!U_XX(I-1)
Nz = Nz + 1
AA(Nz) = BB1
JR(Nz) = N + I
JC(Nz) = N + I -1

```

```

!U_XX(I)
Nz = Nz + 1
AA(Nz) = 1.0_DP
JR(Nz) = N + I
JC(Nz) = N + I

```

```

      !U_XX(I+1)
      Nz = Nz + 1
      AA(Nz) = BB3
      JR(Nz) = N + I
      JC(Nz) = N + I + 1

      Src(N+I) = (CC1*U(I-1) + CC2*U(I) + CC3*U(I+1)) / (Dx**2)

END DO

!For I = N

      !3rd order
      ! !U_X(I-1)
      ! Nz = Nz + 1
      ! AA(Nz) = 6.0_DP / Dx
      ! JR(Nz) = N + N
      ! JC(Nz) = N-1
      !
      ! !U_XX(I-1)
      ! Nz = Nz + 1
      ! AA(Nz) = 5.0_DP
      ! JR(Nz) = N + N
      ! JC(Nz) = N + N-1
      !
      ! !U_XX(I)
      ! Nz = Nz + 1
      ! AA(Nz) = 1.0_DP
      ! JR(Nz) = N + N
      ! JC(Nz) = N + N
      !
      Src(N+N) = (9.0_DP * U(N) - 12.0_DP * U(N-1) + 3.0_DP * U(N-2)) / (Dx**2)

      !4th order
      !U_X(I-1)
      Nz = Nz + 1
      AA(Nz) = 2.5_DP / Dx
      JR(Nz) = N + N
      JC(Nz) = N-1

      !U_XX(I-1)
      Nz = Nz + 1
      AA(Nz) = 8.5_DP
      JR(Nz) = N + N
      JC(Nz) = N + N-1

      !U_XX(I)
      Nz = Nz + 1
      AA(Nz) = 1.0_DP
      JR(Nz) = N + N
      JC(Nz) = N + N

      Src(N+N) = ( 34.0_DP / 3.0_DP * U(N) - 83.0_DP / 4.0_DP * U(N-1) &
      & +10.0_DP * U(N-2) -7.0_DP/12.0_DP * U(N-3) ) / (Dx**2)

END SUBROUTINE ASSIGN_A_JR_JC_CCD

SUBROUTINE ASSIGN_A_JR_JC_CCD_SRC(N , Dx , C_X , U , Src)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Routine assign CCD coeff.
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

```

IMPLICIT NONE
INTEGER,PARAMETER :: DP = SELECTED_REAL_KIND(p=15)

INTEGER,INTENT(IN) :: N
REAL(KIND=DP),INTENT(IN) :: Dx , C_X(:) , U(:)
REAL(KIND=DP),INTENT(OUT) :: Src(:)

INTEGER :: I

REAL(KIND=DP) :: C1 , C2 , C3
REAL(KIND=DP) :: CC1 , CC2 , CC3
REAL(KIND=DP) :: C1D , C2D , C3D

! !For ux eq., a>0
! C1 = -1.9360222951311040338_DP
! C2 = 1.9970445902622080676_DP
! C3 = -0.061022295131104033796_DP
!
! !For ux eq., a<0
! C1D = 0.061022295131104033796_DP
! C2D = -1.9970445902622080676_DP
! C3D = 1.9360222951311040338_DP

!For ux eq., a>0
C1 = -1.9318089987023651093_DP
C2 = 1.9886179974047302197_DP
C3 = -0.056808998702365109925_DP

!For ux eq., a<0
C1D = 0.056808998702365109925_DP
C2D = -1.9886179974047302197_DP
C3D = 1.9318089987023651093_DP

!For uxx eq.
CC1 = 3.0_DP
CC2 = -6.0_DP
CC3 = 3.0_DP

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_X EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1

Src(1) = (-3.5_DP * U(1) + 4.0_DP * U(2) - 0.5_DP * U(3)) / Dx

!For I = 2 , Nx-1
DO I = 2 , N-1
  IF(C_X(I) >= 0.0_DP) THEN

    Src(I) = (C1*U(I-1) + C2*U(I) + C3*U(I+1)) / Dx

  ELSE

    Src(I) = (C1D*U(I-1) + C2D*U(I) + C3D*U(I+1)) / Dx

  END IF
END DO

!For I = N

Src(N) = -(-3.5_DP * U(N) + 4.0_DP * U(N-1) - 0.5_DP * U(N-2)) / Dx

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_XX EQ.

```

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1

* !3rd order
!Src(N+1) = (9.0_DP * U(1) - 12.0_DP * U(2) + 3.0_DP * U(3)) / (Dx**2)

* !4th order
Src(N+1) = ( 34.0_DP / 3.0_DP * U(1) - 83.0_DP / 4.0_DP * U(2) &
& +10.0_DP * U(3) -7.0_DP/12.0_DP * U(4) ) / (Dx**2)

!For I = 2 , N-1
DO I = 2 , N-1

    Src(N+I) = (CC1*U(I-1) + CC2*U(I) + CC3*U(I+1)) / (Dx**2)

END DO

!3rd order
!Src(N+N) = (9.0_DP * U(N) - 12.0_DP * U(N-1) + 3.0_DP * U(N-2)) / (Dx**2)

!4th order
Src(N+N) = ( 34.0_DP / 3.0_DP * U(N) - 83.0_DP / 4.0_DP * U(N-1) &
& +10.0_DP * U(N-2) -7.0_DP/12.0_DP * U(N-3) ) / (Dx**2)

END SUBROUTINE ASSIGN_A_JR_JC_CCD_SRC

SUBROUTINE ASSIGN_A_JR_JC_CCD_2(N , Dx , C_X , U , Nz , AA , JR , JC , Src)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Routine assign CCD coeff.
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
IMPLICIT NONE
INTEGER,PARAMETER :: DP = SELECTED_REAL_KIND(p=15)

INTEGER,INTENT(IN) :: N
INTEGER,INTENT(OUT) :: Nz , JR(:) , JC(:)
REAL(KIND=DP),INTENT(IN) :: Dx , C_X(:) , U(:)
REAL(KIND=DP),INTENT(OUT) :: AA(:) , Src(:)

INTEGER :: I

REAL(KIND=DP) :: A1 , A3 , B1 , B3 , C1 , C2 , C3
REAL(KIND=DP) :: AA1 , AA3 , BB1 , BB3 , CC1 , CC2 , CC3
REAL(KIND=DP) :: A1D , A3D , B1D , B3D , C1D , C2D , C3D

!For ux eq., a>0
A1 = 0.59908531371653270938_DP
A3 = 0.27591468628346729054_DP
B1 = 0.094817062743306541879_DP
B3 = -0.030182937256693458113_DP
C1 = -1.1960365019464523346_DP
C2 = 0.51707300389290467012_DP
C3 = 0.67896349805354766484_DP

!For ux eq., a<0
A1D = 0.27591468628346729054_DP
A3D = 0.59908531371653270938_DP
B1D = 0.030182937256693458113_DP
B3D = -0.094817062743306541879_DP

```

```
C1D = -0.67896349805354766484_DP
C2D = -0.51707300389290467012_DP
C3D = 1.1960365019464523346_DP
```

```
!For uxx eq.
```

```
AA1 = -9.0_DP / 8.0_DP
AA3 = 9.0_DP / 8.0_DP
BB1 = -1.0_DP / 8.0_DP
BB3 = -1.0_DP / 8.0_DP
CC1 = 3.0_DP
CC2 = -6.0_DP
CC3 = 3.0_DP
```

```
Nz = 0
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!For U_X EQ.
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!For I = 1
```

```
!U_X(I)
```

```
Nz = Nz+1
```

```
AA(Nz) = 1.0_DP
```

```
JR(Nz) = 1
```

```
JC(Nz) = 1
```

```
!U_X(I+1)
```

```
Nz = Nz+1
```

```
AA(Nz) = 2.0_DP
```

```
JR(Nz) = 1
```

```
JC(Nz) = 2
```

```
!U_XX(I+1)
```

```
Nz = Nz+1
```

```
AA(Nz) = -Dx
```

```
JR(Nz) = 1
```

```
JC(Nz) = N + 2
```

```
Src(1) = (-3.5_DP * U(1) + 4.0_DP * U(2) - 0.5_DP * U(3)) / Dx
```

```
!For I = 2 , Nx-1
```

```
DO I = 2 , N-1
```

```
IF(C_X(I) >= 0.0_DP) THEN
```

```
!U_X(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = A1
```

```
JR(Nz) = I
```

```
JC(Nz) = I-1
```

```
!U_X(I)
```

```
Nz = Nz + 1
```

```
AA(Nz) = 1.0_DP
```

```
JR(Nz) = I
```

```
JC(Nz) = I
```

```
!U_X(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = A3
```

```
JR(Nz) = I
```

```
JC(Nz) = I+1
```

```
!U_XX(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = B1*Dx
```

```
JR(Nz) = I
```

```
JC(Nz) = N + I -1
```

```

      !U_XX(I+1)
      Nz = Nz + 1
      AA(Nz) = B3*Dx
      JR(Nz) = I
      JC(Nz) = N + I + 1

      Src(I) = (C1*U(I-1) + C2*U(I) + C3*U(I+1)) / Dx

ELSE

      !U_X(I+1)
      Nz = Nz + 1
      AA(Nz) = A1D
      JR(Nz) = I
      JC(Nz) = I-1

      !U_X(I)
      Nz = Nz + 1
      AA(Nz) = 1.0_DP
      JR(Nz) = I
      JC(Nz) = I

      !U_X(I+1)
      Nz = Nz + 1
      AA(Nz) = A3D
      JR(Nz) = I
      JC(Nz) = I+1

      !U_XX(I-1)
      Nz = Nz + 1
      AA(Nz) = B1D*Dx
      JR(Nz) = I
      JC(Nz) = N + I -1

      !U_XX(I+1)
      Nz = Nz + 1
      AA(Nz) = B3D*Dx
      JR(Nz) = I
      JC(Nz) = N + I + 1

      Src(I) = (C1D*U(I-1) + C2D*U(I) + C3D*U(I+1)) / Dx

END IF
END DO

!For I = N
!U_X(I-1)
Nz = Nz + 1
AA(Nz) = 2.0_DP
JR(Nz) = N
JC(Nz) = N-1

!U_X(I)
Nz = Nz + 1
AA(Nz) = 1.0_DP
JR(Nz) = N
JC(Nz) = N

!U_XX(I-1)
Nz = Nz + 1
AA(Nz) = Dx
JR(Nz) = N
JC(Nz) = N + N-1

Src(N) = -(-3.5_DP * U(N) + 4.0_DP * U(N-1) - 0.5_DP * U(N-2)) / Dx

```

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_XX EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1

!3rd order
! !U_X(I+1)
! Nz = Nz+1
! AA(Nz) = -6.0_DP / Dx
! JR(Nz) = N + 1
! JC(Nz) = 2
!
!
! !U_XX(I)
! Nz = Nz+1
! AA(Nz) = 1.0_DP
! JR(Nz) = N + 1
! JC(Nz) = N + 1
!
! !U_XX(I+1)
! Nz = Nz+1
! AA(Nz) = 5.0_DP
! JR(Nz) = N + 1
! JC(Nz) = N + 2
!
! Src(N+1) = (9.0_DP * U(1) - 12.0_DP * U(2) + 3.0_DP * U(3)) / (Dx**2)

!4th order
!U_X(I+1)
Nz = Nz+1
AA(Nz) = -2.5_DP / Dx
JR(Nz) = N + 1
JC(Nz) = 2

!U_XX(I)
Nz = Nz+1
AA(Nz) = 1.0_DP
JR(Nz) = N + 1
JC(Nz) = N + 1

!U_XX(I+1)
Nz = Nz+1
AA(Nz) = 8.5_DP
JR(Nz) = N + 1
JC(Nz) = N + 2

Src(N+1) = ( 34.0_DP / 3.0_DP * U(1) - 83.0_DP / 4.0_DP * U(2) &
& +10.0_DP * U(3) -7.0_DP/12.0_DP * U(4) ) / (Dx**2)

!For I = 2 , N-1
DO I = 2 , N-1
!U_X(I-1)
Nz = Nz + 1
AA(Nz) = AA1 / Dx
JR(Nz) = N + I
JC(Nz) = I-1

!U_X(I+1)
Nz = Nz + 1
AA(Nz) = AA3 / Dx
JR(Nz) = N + I
JC(Nz) = I+1

```

```

      !U_XX(I-1)
      Nz = Nz + 1
      AA(Nz) = BB1
      JR(Nz) = N + I
      JC(Nz) = N + I -1

      !U_XX(I)
      Nz = Nz + 1
      AA(Nz) = 1.0_DP
      JR(Nz) = N + I
      JC(Nz) = N + I

      !U_XX(I+1)
      Nz = Nz + 1
      AA(Nz) = BB3
      JR(Nz) = N + I
      JC(Nz) = N + I + 1

      Src(N+I) = (CC1*U(I-1) + CC2*U(I) + CC3*U(I+1)) / (Dx**2)

END DO

!For I = N

      !3rd order
      ! !U_X(I-1)
      ! Nz = Nz + 1
      ! AA(Nz) = 6.0_DP / Dx
      ! JR(Nz) = N + N
      ! JC(Nz) = N-1
      !
      ! !U_XX(I-1)
      ! Nz = Nz + 1
      ! AA(Nz) = 5.0_DP
      ! JR(Nz) = N + N
      ! JC(Nz) = N + N-1
      !
      ! !U_XX(I)
      ! Nz = Nz + 1
      ! AA(Nz) = 1.0_DP
      ! JR(Nz) = N + N
      ! JC(Nz) = N + N
      !
      ! Src(N+N) = (9.0_DP * U(N) - 12.0_DP * U(N-1) + 3.0_DP * U(N-2)) / (Dx**2)

      !4th order
      !U_X(I-1)
      Nz = Nz + 1
      AA(Nz) = 2.5_DP / Dx
      JR(Nz) = N + N
      JC(Nz) = N-1

      !U_XX(I-1)
      Nz = Nz + 1
      AA(Nz) = 8.5_DP
      JR(Nz) = N + N
      JC(Nz) = N + N-1

      !U_XX(I)
      Nz = Nz + 1
      AA(Nz) = 1.0_DP
      JR(Nz) = N + N
      JC(Nz) = N + N

      Src(N+N) = ( 34.0_DP / 3.0_DP * U(N) - 83.0_DP / 4.0_DP * U(N-1) &

```

$$\& +10.0_DP * U(N-2) - 7.0_DP/12.0_DP * U(N-3)) / (Dx**2)$$

END SUBROUTINE ASSIGN_A_JR_JC_CCD_2

SUBROUTINE ASSIGN_A_JR_JC_CCD_SRC_2(N , Dx , C_X , U , Src)

!!

!Routine assign CCD coeff.

!

!

!

!!

IMPLICIT NONE

INTEGER,PARAMETER :: DP = SELECTED_REAL_KIND(p=15)

INTEGER,INTENT(IN) :: N

REAL(KIND=DP),INTENT(IN) :: Dx , C_X(:) , U(:)

REAL(KIND=DP),INTENT(OUT) :: Src(:)

INTEGER :: I

REAL(KIND=DP) :: C1 , C2 , C3

REAL(KIND=DP) :: CC1 , CC2 , CC3

REAL(KIND=DP) :: C1D , C2D , C3D

!For ux eq., a>0

C1 = -1.1960365019464523346_DP

C2 = 0.51707300389290467012_DP

C3 = 0.67896349805354766484_DP

!For ux eq., a<0

C1D = -0.67896349805354766484_DP

C2D = -0.51707300389290467012_DP

C3D = 1.1960365019464523346_DP

!For uxx eq.

CC1 = 3.0_DP

CC2 = -6.0_DP

CC3 = 3.0_DP

!!

!For U_X EQ.

!!

!For I = 1

Src(1) = (-3.5_DP * U(1) + 4.0_DP * U(2) - 0.5_DP * U(3)) / Dx

!For I = 2 , Nx-1

DO I = 2 , N-1

IF(C_X(I) >= 0.0_DP) THEN

Src(I) = (C1*U(I-1) + C2*U(I) + C3*U(I+1)) / Dx

ELSE

Src(I) = (C1D*U(I-1) + C2D*U(I) + C3D*U(I+1)) / Dx

END IF

END DO

!For I = N

Src(N) = -(-3.5_DP * U(N) + 4.0_DP * U(N-1) - 0.5_DP * U(N-2)) / Dx

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_XX EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*!For I = 1

!3rd order
*!Src(N+1) = (9.0_DP * U(1) - 12.0_DP * U(2) + 3.0_DP * U(3)) / (Dx**2)

!4th order
Src(N+1) = ( 34.0_DP / 3.0_DP * U(1) - 83.0_DP / 4.0_DP * U(2) &
& +10.0_DP * U(3) -7.0_DP/12.0_DP * U(4) ) / (Dx**2)

!For I = 2 , N-1
DO I = 2 , N-1

    Src(N+I) = (CC1*U(I-1) + CC2*U(I) + CC3*U(I+1)) / (Dx**2)

END DO

!3rd order
!Src(N+N) = (9.0_DP * U(N) - 12.0_DP * U(N-1) + 3.0_DP * U(N-2)) / (Dx**2)

!4th order
Src(N+N) = ( 34.0_DP / 3.0_DP * U(N) - 83.0_DP / 4.0_DP * U(N-1) &
& +10.0_DP * U(N-2) -7.0_DP/12.0_DP * U(N-3) ) / (Dx**2)

END SUBROUTINE ASSIGN_A_JR_JC_CCD_SRC_2

~
SUBROUTINE ASSIGN_A_JR_JC_CCD_LS(N , Dx , C_X , U , Nz , AA , JR , JC , Src)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Routine assign CCD coeff.
!
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
IMPLICIT NONE
INTEGER,PARAMETER :: DP = SELECTED_REAL_KIND(p=15)

INTEGER,INTENT(IN) :: N
INTEGER,INTENT(OUT) :: Nz , JR(:) , JC(:)
REAL(KIND=DP),INTENT(IN) :: Dx , C_X(:) , U(:)
REAL(KIND=DP),INTENT(OUT) :: AA(:) , Src(:)

INTEGER :: I

REAL(KIND=DP) :: A1 , B1 , B2 , B3 , C1 , C2 , C3
REAL(KIND=DP) :: AA1 , AA3 , BB1 , BB3 , CC1 , CC2 , CC3
REAL(KIND=DP) :: A3D , B1D , B2D , B3D , C1D , C2D , C3D

! !For ux eq., a>0
! A1 = 0.875_DP
! B1 = 0.11657006997771659445_DP
! B2 = -0.33429930022283405532_DP
! B3 = -0.0084299300222834055385_DP
! C1 = -2.0386591602674008665_DP
! C2 = 2.2023183205348017329_DP
! C3 = -0.16365916026740086650_DP
!
! !For ux eq., a<0
! A3D = 0.875_DP
! B1D = 0.0084299300222834055385_DP
! B2D = 0.33429930022283405532_DP

```

```
! B3D = -0.11657006997771659445_DP
! C1D = 0.16365916026740086650_DP
! C2D = -2.2023183205348017329_DP
! C3D = 2.0386591602674008665_DP
```

```
!For ux eq., a>0
A1 = 0.875_DP
B1 = 0.12547425010813624084_DP
B2 = -0.24525749891863759159_DP
B3 = 0.00047425010813624083774_DP
C1 = -1.9318089987023651093_DP
C2 = 1.9886179974047302197_DP
C3 = -0.056808998702365109925_DP
```

```
!For ux eq., a<0
A3D = 0.875_DP
B1D = -0.00047425010813624083774_DP
B2D = 0.24525749891863759159_DP
B3D = -0.12547425010813624084_DP
C1D = 0.056808998702365109925_DP
C2D = -1.9886179974047302197_DP
C3D = 1.9318089987023651093_DP
```

```
!For uxx eq.
AA1 = -9.0_DP / 8.0_DP
AA3 = 9.0_DP / 8.0_DP
BB1 = -1.0_DP / 8.0_DP
BB3 = -1.0_DP / 8.0_DP
CC1 = 3.0_DP
CC2 = -6.0_DP
CC3 = 3.0_DP
```

```
Nz = 0
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!For U_X EQ.
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!For I = 1
```

```
!U_X(I)
```

```
Nz = Nz+1
```

```
AA(Nz) = 1.0_DP
```

```
JR(Nz) = 1
```

```
JC(Nz) = 1
```

```
Src(1) = 0.0_DP
```

```
!For I = 2 , Nx-1
```

```
DO I = 2 , N-1
```

```
IF(C_X(I) >= 0.0_DP) THEN
```

```
!U_X(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = A1
```

```
JR(Nz) = I
```

```
JC(Nz) = I-1
```

```
!U_X(I)
```

```
Nz = Nz + 1
```

```
AA(Nz) = 1.0_DP
```

```
JR(Nz) = I
```

```
JC(Nz) = I
```

```
!U_XX(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = B1*Dx
```

```
JR(Nz) = I
```

```
JC(Nz) = N + I -1
```

```

      !U_XX(I)
      Nz = Nz + 1
      AA(Nz) = B2*Dx
      JR(Nz) = I
      JC(Nz) = N + I

      !U_XX(I+1)
      Nz = Nz + 1
      AA(Nz) = B3*Dx
      JR(Nz) = I
      JC(Nz) = N + I + 1

      Src(I) = (C1*U(I-1) + C2*U(I) + C3*U(I+1)) / Dx

ELSE
      !U_X(I)
      Nz = Nz + 1
      AA(Nz) = 1.0_DP
      JR(Nz) = I
      JC(Nz) = I

      !U_X(I+1)
      Nz = Nz + 1
      AA(Nz) = A3D
      JR(Nz) = I
      JC(Nz) = I+1

      !U_XX(I-1)
      Nz = Nz + 1
      AA(Nz) = B1D*Dx
      JR(Nz) = I
      JC(Nz) = N + I -1

      !U_XX(I)
      Nz = Nz + 1
      AA(Nz) = B2D*Dx
      JR(Nz) = I
      JC(Nz) = N + I

      !U_XX(I+1)
      Nz = Nz + 1
      AA(Nz) = B3D*Dx
      JR(Nz) = I
      JC(Nz) = N + I + 1

      Src(I) = (C1D*U(I-1) + C2D*U(I) + C3D*U(I+1)) / Dx

END IF
END DO

!For I = N
!U_X(I)
Nz = Nz + 1
AA(Nz) = 1.0_DP
JR(Nz) = N
JC(Nz) = N

Src(N) = 0.0_DP

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_XX EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1

!U_X(I+1)

```

```

Nz = Nz+1
AA(Nz) = -6.0_DP / Dx
JR(Nz) = N + 1
JC(Nz) = 2

!U_XX(I)
Nz = Nz+1
AA(Nz) = 1.0_DP
JR(Nz) = N + 1
JC(Nz) = N + 1

!U_XX(I+1)
Nz = Nz+1
AA(Nz) = 5.0_DP
JR(Nz) = N + 1
JC(Nz) = N + 2

Src(N+1) = (9.0_DP * U(1) - 12.0_DP * U(2) + 3.0_DP * U(3)) / (Dx**2)

!For I = 2 , N-1
DO I = 2 , N-1
    !U_X(I-1)
    Nz = Nz + 1
    AA(Nz) = AA1 / Dx
    JR(Nz) = N + I
    JC(Nz) = I-1

    !U_X(I+1)
    Nz = Nz + 1
    AA(Nz) = AA3 / Dx
    JR(Nz) = N + I
    JC(Nz) = I+1

    !U_XX(I-1)
    Nz = Nz + 1
    AA(Nz) = BB1
    JR(Nz) = N + I
    JC(Nz) = N + I -1

    !U_XX(I)
    Nz = Nz + 1
    AA(Nz) = 1.0_DP
    JR(Nz) = N + I
    JC(Nz) = N + I

    !U_XX(I+1)
    Nz = Nz + 1
    AA(Nz) = BB3
    JR(Nz) = N + I
    JC(Nz) = N + I + 1

    Src(N+I) = (CC1*U(I-1) + CC2*U(I) + CC3*U(I+1)) / (Dx**2)

END DO

!For I = N
!U_X(I-1)
Nz = Nz + 1
AA(Nz) = 6.0_DP / Dx
JR(Nz) = N + N
JC(Nz) = N-1

!U_XX(I-1)
Nz = Nz + 1
AA(Nz) = 5.0_DP

```

```

JR(Nz) = N + N
JC(Nz) = N + N-1

!U_XX(I)
Nz = Nz + 1
AA(Nz) = 1.0_DP
*JR(Nz) = N + N
JC(Nz) = N + N

Src(N+N) = (9.0_DP * U(N) - 12.0_DP * U(N-1) + 3.0_DP * U(N-2)) / (Dx**2)

END SUBROUTINE ASSIGN_A_JR_JC_CCD_LS

SUBROUTINE ASSIGN_A_JR_JC_CCD_FV(N , Dx , Uc , U , Nz , AA , JR , JC , Src)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Routine assign CCD coeff.
!
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
IMPLICIT NONE
INTEGER,PARAMETER :: DP = SELECTED_REAL_KIND(p=15)

INTEGER,INTENT(IN) :: N
INTEGER,INTENT(OUT) :: Nz , JR(:) , JC(:)
REAL(KIND=DP),INTENT(IN) :: Dx , Uc(:) , U(:)
REAL(KIND=DP),INTENT(OUT) :: AA(:) , Src(:)

INTEGER :: I , NN

REAL(KIND=DP) :: A1 , B1 , B2 , B3 , C1 , C3
REAL(KIND=DP) :: AA1 , AA3 , BB1 , BB3 , CC1 , CC3
REAL(KIND=DP) :: A3D , B1D , B2D , B3D , C1D , C3D
REAL(KIND=DP) :: U_old(N)

NN = N - 1

! !For ux eq., a>0
! A1 = 0.875_DP
! B1 = 0.11657006997771659445_DP
! B2 = -0.33429930022283405532_DP
! B3 = -0.0084299300222834055385_DP
! C1 = -2.0386591602674008665_DP
! C3 = -0.16365916026740086650_DP
!
! !For ux eq., a<0
! A3D = 0.875_DP
! B1D = 0.0084299300222834055385_DP
! B2D = 0.33429930022283405532_DP
! B3D = -0.11657006997771659445_DP
! C1D = 0.16365916026740086650_DP
! C3D = 2.0386591602674008665_DP

!For ux eq., a>0
A1 = 0.875_DP
B1 = 0.12547425010813624084_DP
B2 = -0.24525749891863759159_DP
B3 = 0.00047425010813624083774_DP
C1 = -1.9318089987023651093_DP
C3 = -0.056808998702365109925_DP

!For ux eq., a<0
A3D = 0.875_DP
B1D = -0.00047425010813624083774_DP
B2D = 0.24525749891863759159_DP

```

```

B3D = -0.12547425010813624084_DP
C1D = 0.056808998702365109925_DP
C3D = 1.9318089987023651093_DP

```

```
!For uxx eq.
```

```

AA1 = -9.0_DP / 8.0_DP
AA3 = 9.0_DP / 8.0_DP
BB1 = -1.0_DP / 8.0_DP
BB3 = -1.0_DP / 8.0_DP
CC1 = 3.0_DP
CC3 = 3.0_DP

```

```

DO I = 1 , N
  U_old(I) = U(I)
END DO

```

```
!Extrapolate value to ghost cell.
```

```

U_old(1) = ( 8.0_DP * U(1) - 6.0_DP * U(2) + U(3) ) / 3.0_DP
U_old(N) = ( 8.0_DP * U(N) - 6.0_DP * U(N-1) + U(N-2) ) / 3.0_DP
!U_old(1) = 2.0_DP * U(1) - U(2)
!U_old(N) = 2.0_DP * U(N) - U(N-1)

```

```
Nz = 0
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!For U_X EQ.
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!For I = 1
```

```

!U_X(I)
Nz = Nz+1
AA(Nz) = 1.0_DP
JR(Nz) = 1
JC(Nz) = 1

```

```
Src(1) = U(1)
```

```
!For I = 2 , Nx-1
```

```
DO I = 2 , N-2
```

```
IF(Uc(I) >= 0.0_DP) THEN
```

```
!U_X(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = A1
```

```
JR(Nz) = I
```

```
JC(Nz) = I-1
```

```
!U_X(I)
```

```
Nz = Nz + 1
```

```
AA(Nz) = 1.0_DP
```

```
JR(Nz) = I
```

```
JC(Nz) = I
```

```
!U_XX(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = B1*Dx
```

```
JR(Nz) = I
```

```
JC(Nz) = NN + I -1
```

```
!U_XX(I)
```

```
Nz = Nz + 1
```

```
AA(Nz) = B2*Dx
```

```
JR(Nz) = I
```

```

JC(Nz) = NN + I

!U_XX(I+1)
Nz = Nz + 1
AA(Nz) = B3*Dx
JR(Nz) = I
JC(Nz) = NN + I + 1

Src(I) = (C3*U(I+1) - C1*U(I) )

ELSE
!U_X(I)
Nz = Nz + 1
AA(Nz) = 1.0_DP
JR(Nz) = I
JC(Nz) = I

!U_X(I+1)
Nz = Nz + 1
AA(Nz) = A3D
JR(Nz) = I
JC(Nz) = I+1

!U_XX(I-1)
Nz = Nz + 1
AA(Nz) = B1D*Dx
JR(Nz) = I
JC(Nz) = NN + I -1

!U_XX(I)
Nz = Nz + 1
AA(Nz) = B2D*Dx
JR(Nz) = I
JC(Nz) = NN + I

!U_XX(I+1)
Nz = Nz + 1
AA(Nz) = B3D*Dx
JR(Nz) = I
JC(Nz) = NN + I + 1

Src(I) = (C3D*U(I+1) - C1D*U(I) )

END IF
END DO

!For I = NN
!U_X(I)
Nz = Nz + 1
AA(Nz) = 1.0_DP
JR(Nz) = NN
JC(Nz) = NN

Src(NN) = U(N)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_XX EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1

!U_X(I+1)
Nz = Nz+1
AA(Nz) = -2.5_DP / Dx
JR(Nz) = NN + 1
JC(Nz) = 2

```

```
!U_XX(I)
```

```
Nz = Nz+1
```

```
AA(Nz) = 1.0_DP
```

```
JR(Nz) = NN + 1
```

```
JC(Nz) = NN + 1
```

```
!U_XX(I+1)
```

```
Nz = Nz+1
```

```
AA(Nz) = 2.75_DP
```

```
JR(Nz) = NN + 1
```

```
JC(Nz) = NN + 2
```

```
Src(NN+1) = (-23.0_DP * U_old(1) - 199.0_DP * U_old(2) + 107.0_DP * U_old(3) - 5.0_DP * U_old(4) )  
/ Dx / 48.0_DP
```

```
!For I = 2 , N-2
```

```
DO I = 2 , N-2
```

```
!U_X(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = AA1 / Dx
```

```
JR(Nz) = NN + I
```

```
JC(Nz) = I-1
```

```
!U_X(I+1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = AA3 / Dx
```

```
JR(Nz) = NN + I
```

```
JC(Nz) = I+1
```

```
!U_XX(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = BB1
```

```
JR(Nz) = NN + I
```

```
JC(Nz) = NN + I -1
```

```
!U_XX(I)
```

```
Nz = Nz + 1
```

```
AA(Nz) = 1.0_DP
```

```
JR(Nz) = NN + I
```

```
JC(Nz) = NN + I
```

```
!U_XX(I+1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = BB3
```

```
JR(Nz) = NN + I
```

```
JC(Nz) = NN + I + 1
```

```
Src(NN+I) = (CC3*U(I+1) - CC1*U(I) ) / Dx
```

```
END DO
```

```
!For I = N
```

```
!U_X(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = 2.5_DP / Dx
```

```
JR(Nz) = NN + NN
```

```
JC(Nz) = NN-1
```

```
!U_XX(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = 2.75_DP
```

```
JR(Nz) = NN + NN
```

```
JC(Nz) = NN + NN-1
```

```
!U_XX(I)
```

```

Nz = Nz + 1
AA(Nz) = 1.0_DP
JR(Nz) = NN + NN
JC(Nz) = NN + NN

```

```

Src(NN+NN) = (23.0_DP * U_old(N) + 199.0_DP * U_old(N-1) - 107.0_DP * U_old(N-2) + 5.0_DP *
U_old(N-3) ) / Dx / 48.0_DP

```

```

END SUBROUTINE ASSIGN_A_JR_JC_CCD_FV

```

```

SUBROUTINE ASSIGN_A_JR_JC_CCD_FV_CLS(N , Dx , Uc , U , Nz , AA , JR , JC , Src)

```

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

```

!Routine assign CCD coeff.

```

```

!

```

```

!

```

```

!

```

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

```

IMPLICIT NONE

```

```

INTEGER,PARAMETER :: DP = SELECTED_REAL_KIND(p=15)

```

```

INTEGER,INTENT(IN) :: N

```

```

INTEGER,INTENT(OUT) :: Nz , JR(:) , JC(:)

```

```

REAL(KIND=DP),INTENT(IN) :: Dx , Uc(:) , U(:)

```

```

REAL(KIND=DP),INTENT(OUT) :: AA(:) , Src(:)

```

```

INTEGER :: I , NN

```

```

REAL(KIND=DP) :: A1 , B1 , B2 , B3 , C1 , C3

```

```

REAL(KIND=DP) :: AA1 , AA3 , BB1 , BB3 , CC1 , CC3

```

```

REAL(KIND=DP) :: A3D , B1D , B2D , B3D , C1D , C3D

```

```

NN = N - 1

```

```

! !For ux eq., a>0

```

```

! A1 = 0.875_DP

```

```

! B1 = 0.11657006997771659445_DP

```

```

! B2 = -0.33429930022283405532_DP

```

```

! B3 = -0.0084299300222834055385_DP

```

```

! C1 = -2.0386591602674008665_DP

```

```

! C3 = -0.16365916026740086650_DP

```

```

!

```

```

! !For ux eq., a<0

```

```

! A3D = 0.875_DP

```

```

! B1D = 0.0084299300222834055385_DP

```

```

! B2D = 0.33429930022283405532_DP

```

```

! B3D = -0.11657006997771659445_DP

```

```

! C1D = 0.16365916026740086650_DP

```

```

! C3D = 2.0386591602674008665_DP

```

```

!For ux eq., a>0

```

```

A1 = 0.875_DP

```

```

B1 = 0.12547425010813624084_DP

```

```

B2 = -0.24525749891863759159_DP

```

```

B3 = 0.00047425010813624083774_DP

```

```

C1 = -1.9318089987023651093_DP

```

```

C3 = -0.056808998702365109925_DP

```

```

!For ux eq., a<0

```

```

A3D = 0.875_DP

```

```

B1D = -0.00047425010813624083774_DP

```

```

B2D = 0.24525749891863759159_DP

```

```

B3D = -0.12547425010813624084_DP

```

```

C1D = 0.056808998702365109925_DP

```

```

C3D = 1.9318089987023651093_DP

```

```

!For uxx eq.
AA1 = -9.0_DP / 8.0_DP
AA3 = 9.0_DP / 8.0_DP
BB1 = -1.0_DP / 8.0_DP
BB3 = -1.0_DP / 8.0_DP
CC1 = 3.0_DP
CC3 = 3.0_DP

Nz = 0

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_X EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1
!U_X(I)
Nz = Nz+1
AA(Nz) = 1.0_DP
JR(Nz) = 1
JC(Nz) = 1

Src(1) = U(1)

!For I = 2 , Nx-1
DO I = 2 , N-2
  IF(Uc(I) >= 0.0_DP) THEN
    !U_X(I-1)
    Nz = Nz + 1
    AA(Nz) = A1
    JR(Nz) = I
    JC(Nz) = I-1

    !U_X(I)
    Nz = Nz + 1
    AA(Nz) = 1.0_DP
    JR(Nz) = I
    JC(Nz) = I

    !U_XX(I-1)
    Nz = Nz + 1
    AA(Nz) = B1*Dx
    JR(Nz) = I
    JC(Nz) = NN + I -1

    !U_XX(I)
    Nz = Nz + 1
    AA(Nz) = B2*Dx
    JR(Nz) = I
    JC(Nz) = NN + I

    !U_XX(I+1)
    Nz = Nz + 1
    AA(Nz) = B3*Dx
    JR(Nz) = I
    JC(Nz) = NN + I + 1

    Src(I) = (C3*U(I+1) - C1*U(I) )

  ELSE
    !U_X(I)
    Nz = Nz + 1
    AA(Nz) = 1.0_DP
    JR(Nz) = I
    JC(Nz) = I

```

```

      !U_X(I+1)
      Nz = Nz + 1
      AA(Nz) = A3D
      JR(Nz) = I
      JC(Nz) = I+1

      !U_XX(I-1)
      Nz = Nz + 1
      AA(Nz) = B1D*Dx
      JR(Nz) = I
      JC(Nz) = NN + I -1

      !U_XX(I)
      Nz = Nz + 1
      AA(Nz) = B2D*Dx
      JR(Nz) = I
      JC(Nz) = NN + I

      !U_XX(I+1)
      Nz = Nz + 1
      AA(Nz) = B3D*Dx
      JR(Nz) = I
      JC(Nz) = NN + I + 1

      Src(I) = (C3D*U(I+1) - C1D*U(I) )

```

```

      END IF
    END DO

```

```

      !For I = NN
      !U_X(I)
      Nz = Nz + 1
      AA(Nz) = 1.0_DP
      JR(Nz) = NN
      JC(Nz) = NN

```

```

      Src(NN) = U(N)

```

```

      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      !For U_XX EQ.
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      !For I = 1

```

```

      !U_XX(I)
      Nz = Nz+1
      AA(Nz) = 1.0_DP
      JR(Nz) = NN + 1
      JC(Nz) = NN + 1

```

```

      Src(NN+1) = 0.0_DP

```

```

      !For I = 2 , N-2
      DO I = 2 , N-2
        !U_X(I-1)
        Nz = Nz + 1
        AA(Nz) = AA1 / Dx
        JR(Nz) = NN + I
        JC(Nz) = I-1

```

```

      !U_X(I+1)
      Nz = Nz + 1
      AA(Nz) = AA3 / Dx
      JR(Nz) = NN + I
      JC(Nz) = I+1

```

```

      !U_XX(I-1)

```

```

      Nz = Nz + 1
      AA(Nz) = BB1
      JR(Nz) = NN + I
      JC(Nz) = NN + I - 1

      !U_XX(I)
      Nz = Nz + 1
      AA(Nz) = 1.0_DP
      JR(Nz) = NN + I
      JC(Nz) = NN + I

      !U_XX(I+1)
      Nz = Nz + 1
      AA(Nz) = BB3
      JR(Nz) = NN + I
      JC(Nz) = NN + I + 1

      Src(NN+I) = (CC3*U(I+1) - CC1*U(I) ) / Dx

END DO

!For I = N
!U_XX(I)
Nz = Nz + 1
AA(Nz) = 1.0_DP
JR(Nz) = NN + NN
JC(Nz) = NN + NN

Src(NN+NN) = 0.0_DP

END SUBROUTINE ASSIGN_A_JR_JC_CCD_FV_CLS

SUBROUTINE ASSIGN_A_JR_JC_CCD_FV_CLS_SRC(N , Dx , Uc , U , Src)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Routine assign CCD coeff.
!
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
IMPLICIT NONE
INTEGER,PARAMETER :: DP = SELECTED_REAL_KIND(p=15)

INTEGER,INTENT(IN) :: N
REAL(KIND=DP),INTENT(IN) :: Dx , Uc(:) , U(:)
REAL(KIND=DP),INTENT(OUT) :: Src(:)

INTEGER :: I , NN

REAL(KIND=DP) :: C1 , C3
REAL(KIND=DP) :: CC1 , CC3
REAL(KIND=DP) :: C1D , C3D

NN = N - 1

! !For ux eq., a>0
! C1 = -2.0386591602674008665_DP
! C3 = -0.16365916026740086650_DP
!
! !For ux eq., a<0
! C1D = 0.16365916026740086650_DP
! C3D = 2.0386591602674008665_DP

!For ux eq., a>0
C1 = -1.9318089987023651093_DP
C3 = -0.056808998702365109925_DP

```

```

!For ux eq., a<0
C1D = 0.056808998702365109925_DP
C3D = 1.9318089987023651093_DP

!For uxx eq.
CC1 = 3.0_DP
CC3 = 3.0_DP

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_X EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1
Src(1) = U(1)

!For I = 2 , Nx-1
DO I = 2 , N-2
  IF(Uc(I) >= 0.0_DP) THEN

    Src(I) = (C3*U(I+1) - C1*U(I) )

  ELSE

    Src(I) = (C3D*U(I+1) - C1D*U(I) )

  END IF
END DO

!For I = NN
Src(NN) = U(N)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_XX EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1
Src(NN+1) = 0.0_DP

!For I = 2 , N-2
DO I = 2 , N-2

  Src(NN+I) = (CC3*U(I+1) - CC1*U(I) ) / Dx

END DO

!For I = N
Src(NN+NN) = 0.0_DP

END SUBROUTINE ASSIGN_A_JR_JC_CCD_FV_CLS_SRC

SUBROUTINE ASSIGN_A_JR_JC_CCD_P(N , Dx , U , Nz , AA , JR , JC , Src)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Routine assign CCD coeff.
!
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  IMPLICIT NONE
  INTEGER,PARAMETER :: DP = SELECTED_REAL_KIND(p=15)

  INTEGER,INTENT(IN) :: N
  INTEGER,INTENT(OUT) :: Nz , JR(:) , JC(:)
  REAL(KIND=DP),INTENT(IN) :: Dx , U(:)
  REAL(KIND=DP),INTENT(OUT) :: AA(:) , Src(:)

```

```
INTEGER :: I
```

```
REAL(KIND=DP) :: A1 , A3 , B1 , B3 , C1 , C3
```

```
REAL(KIND=DP) :: AA1 , AA3 , BB1 , BB3 , CC1 , CC2 , CC3
```

```
!For ux eq., a>0
```

```
A1 = 7.0_DP / 16.0_DP
```

```
A3 = 7.0_DP / 16.0_DP
```

```
B1 = 1.0_DP / 16.0_DP
```

```
B3 = -1.0_DP / 16.0_DP
```

```
C1 = -15.0_DP / 16.0_DP
```

```
C3 = 15.0_DP / 16.0_DP
```

```
!For uxx eq.
```

```
AA1 = -9.0_DP / 8.0_DP
```

```
AA3 = 9.0_DP / 8.0_DP
```

```
BB1 = -1.0_DP / 8.0_DP
```

```
BB3 = -1.0_DP / 8.0_DP
```

```
CC1 = 3.0_DP
```

```
CC2 = -6.0_DP
```

```
CC3 = 3.0_DP
```

```
Nz = 0
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!For U_X EQ.
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!For I = 1
```

```
!U_X(I)
```

```
Nz = Nz+1
```

```
AA(Nz) = 1.0_DP
```

```
JR(Nz) = 1
```

```
JC(Nz) = 1
```

```
!U_X(I+1)
```

```
Nz = Nz+1
```

```
AA(Nz) = 2.0_DP
```

```
JR(Nz) = 1
```

```
JC(Nz) = 2
```

```
!U_XX(I+1)
```

```
Nz = Nz+1
```

```
AA(Nz) = -Dx
```

```
JR(Nz) = 1
```

```
JC(Nz) = N + 2
```

```
Src(1) = (-3.5_DP * U(1) + 4.0_DP * U(2) - 0.5_DP * U(3)) / Dx
```

```
!For I = 2 , Nx-1
```

```
DO I = 2 , N-1
```

```
!U_X(I-1)
```

```
Nz = Nz + 1
```

```
AA(Nz) = A1
```

```
JR(Nz) = I
```

```
JC(Nz) = I-1
```

```
!U_X(I)
```

```
Nz = Nz + 1
```

```
AA(Nz) = 1.0_DP
```

```
JR(Nz) = I
```

```
JC(Nz) = I
```

```
!U_X(I+1)
```

```

Nz = Nz + 1
AA(Nz) = A3
JR(Nz) = I
JC(Nz) = I+1

```

```

!U_XX(I-1)
Nz = Nz + 1
AA(Nz) = B1*Dx
JR(Nz) = I
JC(Nz) = N + I -1

```

```

!U_XX(I+1)
Nz = Nz + 1
AA(Nz) = B3*Dx
JR(Nz) = I
JC(Nz) = N + I + 1

```

```

Src(I) = (C1*U(I-1) + C3*U(I+1)) / Dx

```

```

END DO

```

```

!For I = N
!U_X(I-1)
Nz = Nz + 1
AA(Nz) = 2.0_DP
JR(Nz) = N
JC(Nz) = N-1

```

```

!U_X(I)
Nz = Nz + 1
AA(Nz) = 1.0_DP
JR(Nz) = N
JC(Nz) = N

```

```

!U_XX(I-1)
Nz = Nz + 1
AA(Nz) = Dx
JR(Nz) = N
JC(Nz) = N + N-1

```

```

Src(N) = -(-3.5_DP * U(N) + 4.0_DP * U(N-1) - 0.5_DP * U(N-2)) / Dx

```

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_XX EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1

```

```

!3rd order
! !U_X(I+1)
! Nz = Nz+1
! AA(Nz) = -6.0_DP / Dx
! JR(Nz) = N + 1
! JC(Nz) = 2
!
!
! !U_XX(I)
! Nz = Nz+1
! AA(Nz) = 1.0_DP
! JR(Nz) = N + 1
! JC(Nz) = N + 1
!
! !U_XX(I+1)
! Nz = Nz+1
! AA(Nz) = 5.0_DP

```

```

! JR(Nz) = N + 1
! JC(Nz) = N + 2
!
! Src(N+1) = (9.0_DP * U(1) - 12.0_DP * U(2) + 3.0_DP * U(3)) / (Dx**2)

!4th order
!U_X(I+1)
Nz = Nz+1
AA(Nz) = -2.5_DP / Dx
JR(Nz) = N + 1
JC(Nz) = 2

!U_XX(I)
Nz = Nz+1
AA(Nz) = 1.0_DP
JR(Nz) = N + 1
JC(Nz) = N + 1

!U_XX(I+1)
Nz = Nz+1
AA(Nz) = 8.5_DP
JR(Nz) = N + 1
JC(Nz) = N + 2

Src(N+1) = ( 34.0_DP / 3.0_DP * U(1) - 83.0_DP / 4.0_DP * U(2) &
& +10.0_DP * U(3) -7.0_DP/12.0_DP * U(4) ) / (Dx**2)

!For I = 2 , N-1
DO I = 2 , N-1
!U_X(I-1)
Nz = Nz + 1
AA(Nz) = AA1 / Dx
JR(Nz) = N + I
JC(Nz) = I-1

!U_X(I+1)
Nz = Nz + 1
AA(Nz) = AA3 / Dx
JR(Nz) = N + I
JC(Nz) = I+1

!U_XX(I-1)
Nz = Nz + 1
AA(Nz) = BB1
JR(Nz) = N + I
JC(Nz) = N + I -1

!U_XX(I)
Nz = Nz + 1
AA(Nz) = 1.0_DP
JR(Nz) = N + I
JC(Nz) = N + I

!U_XX(I+1)
Nz = Nz + 1
AA(Nz) = BB3
JR(Nz) = N + I
JC(Nz) = N + I + 1

Src(N+I) = (CC1*U(I-1) + CC2*U(I) + CC3*U(I+1)) / (Dx**2)

```

END DO

```
!For I = N
```

```
!3rd order
```

```
! !U_X(I-1)
! Nz = Nz + 1
! AA(Nz) = 6.0_DP / Dx
! JR(Nz) = N + N
! JC(Nz) = N-1
!
! !U_XX(I-1)
! Nz = Nz + 1
! AA(Nz) = 5.0_DP
! JR(Nz) = N + N
! JC(Nz) = N + N-1
!
! !U_XX(I)
! Nz = Nz + 1
! AA(Nz) = 1.0_DP
! JR(Nz) = N + N
! JC(Nz) = N + N
!
! Src(N+N) = (9.0_DP * U(N) - 12.0_DP * U(N-1) + 3.0_DP * U(N-2)) / (Dx**2)
```

```
!4th order
```

```
!U_X(I-1)
Nz = Nz + 1
AA(Nz) = 2.5_DP / Dx
JR(Nz) = N + N
JC(Nz) = N-1
```

```
!U_XX(I-1)
Nz = Nz + 1
AA(Nz) = 8.5_DP
JR(Nz) = N + N
JC(Nz) = N + N-1
```

```
!U_XX(I)
Nz = Nz + 1
AA(Nz) = 1.0_DP
JR(Nz) = N + N
JC(Nz) = N + N
```

```
Src(N+N) = ( 34.0_DP / 3.0_DP * U(N) - 83.0_DP / 4.0_DP * U(N-1) &
& +10.0_DP * U(N-2) -7.0_DP/12.0_DP * U(N-3) ) / (Dx**2)
```

```
END SUBROUTINE ASSIGN_A_JR_JC_CCD_P
```

```
SUBROUTINE ASSIGN_A_JR_JC_CCD_P_SRC(N , Dx , U , Src)
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!Routine assign CCD coeff.
```

```
!
```

```
!
```

```
!
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
IMPLICIT NONE
```

```
INTEGER,PARAMETER :: DP = SELECTED_REAL_KIND(p=15)
```

```
INTEGER,INTENT(IN) :: N
```

```
REAL(KIND=DP),INTENT(IN) :: Dx , U(:)
```

```
REAL(KIND=DP),INTENT(OUT) :: Src(:)
```

```
*
```

```
INTEGER :: I
```

```
REAL(KIND=DP) :: C1 , C3
```

```
REAL(KIND=DP) :: CC1 , CC2 , CC3
```

```
!For ux eq., a>0
C1 = -15.0_DP / 16.0_DP
C3 = 15.0_DP / 16.0_DP
```

```
!For uxx eq.
CC1 = 3.0_DP
CC2 = -6.0_DP
CC3 = 3.0_DP
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_X EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1
```

```
Src(1) = (-3.5_DP * U(1) + 4.0_DP * U(2) - 0.5_DP * U(3)) / Dx
```

```
!For I = 2 , Nx-1
DO I = 2 , N-1
```

```
Src(I) = (C1*U(I-1) + C3*U(I+1)) / Dx
```

```
END DO
```

```
Src(N) = -(-3.5_DP * U(N) + 4.0_DP * U(N-1) - 0.5_DP * U(N-2)) / Dx
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_XX EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1
```

```
!3rd order
```

```
!Src(N+1) = (9.0_DP * U(1) - 12.0_DP * U(2) + 3.0_DP * U(3)) / (Dx**2)
```

```
!4th order
```

```
Src(N+1) = ( 34.0_DP / 3.0_DP * U(1) - 83.0_DP / 4.0_DP * U(2) &
& +10.0_DP * U(3) -7.0_DP/12.0_DP * U(4) ) / (Dx**2)
```

```
!For I = 2 , N-1
DO I = 2 , N-1
```

```
Src(N+I) = (CC1*U(I-1) + CC2*U(I) + CC3*U(I+1)) / (Dx**2)
```

```
END DO
```

```
!3rd order
```

```
!Src(N+N) = (9.0_DP * U(N) - 12.0_DP * U(N-1) + 3.0_DP * U(N-2)) / (Dx**2)
```

```
!4th order
```

```
Src(N+N) = ( 34.0_DP / 3.0_DP * U(N) - 83.0_DP / 4.0_DP * U(N-1) &
& +10.0_DP * U(N-2) -7.0_DP/12.0_DP * U(N-3) ) / (Dx**2)
```

```
END SUBROUTINE ASSIGN_A_JR_JC_CCD_P_SRC
```

```
SUBROUTINE ASSIGN_A_JR_JC_CCD_FV_U(N , Dx , U , Nz , AA , JR , JC , Src)
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!Routine assign CCD coeff.
```

```
!
```

```

!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*IMPLICIT NONE
INTEGER,PARAMETER :: DP = SELECTED_REAL_KIND(p=15)

INTEGER,INTENT(IN) :: N
INTEGER,INTENT(OUT) :: Nz , JR(:) , JC(:)
REAL(KIND=DP),INTENT(IN) :: Dx , U(:)
REAL(KIND=DP),INTENT(OUT) :: AA(:) , Src(:)

INTEGER :: I , NN

REAL(KIND=DP) :: A1 , A3 , B1 , B3 , C1 , C3
REAL(KIND=DP) :: AA1 , AA3 , BB1 , BB3 , CC1 , CC3
REAL(KIND=DP) :: A3D , B1D , B2D , B3D , C1D , C3D

NN = N - 1

!For ux
A1 = 7.0_DP / 16.0_DP
A3 = 7.0_DP / 16.0_DP
B1 = 1.0_DP / 16.0_DP
B3 = -1.0_DP / 16.0_DP
C1 = -15.0_DP / 16.0_DP
C3 = 15.0_DP / 16.0_DP

!For uxx eq.
AA1 = -9.0_DP / 8.0_DP
AA3 = 9.0_DP / 8.0_DP
BB1 = -1.0_DP / 8.0_DP
BB3 = -1.0_DP / 8.0_DP
CC1 = 3.0_DP
CC3 = 3.0_DP

Nz = 0

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For U_X EQ.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!For I = 1
!U_X(I)
Nz = Nz+1
AA(Nz) = 1.0_DP
JR(Nz) = 1
JC(Nz) = 1

!U_X(I+1)
Nz = Nz+1
AA(Nz) = 1.5_DP
JR(Nz) = 1
JC(Nz) = 2

!U_XX(I+1)
Nz = Nz+1
AA(Nz) = -0.75_DP*Dx
JR(Nz) = 1
JC(Nz) = NN + 2

Src(1) = U(1) / 16.0_DP + 139.0_DP / 48.0_DP * U(2) - 23.0_DP / 48.0_DP * U(3) + U(4) / 48.0_DP

!For I = 2 , Nx-1
DO I = 2 , N-2

```

頁： 32

```
JR(Nz) = NN + 1
JC(Nz) = 2
```

```
!U_XX(I)
Nz = Nz+1
AA(Nz) = 1.0_DP
JR(Nz) = NN + 1
JC(Nz) = NN + 1
```

```
!U_XX(I+1)
Nz = Nz+1
AA(Nz) = 2.75_DP
JR(Nz) = NN + 1
JC(Nz) = NN + 2
```

```
Src(NN+1) = (-23.0_DP * U(1) - 199.0_DP * U(2) + 107.0_DP * U(3) - 5.0_DP * U(4) ) / Dx / 48.0_DP
```

```
!For I = 2 , N-2
DO I = 2 , N-2
    !U_X(I-1)
    Nz = Nz + 1
    AA(Nz) = AA1 / Dx
    JR(Nz) = NN + I
    JC(Nz) = I-1
```

```
    !U_X(I+1)
    Nz = Nz + 1
    AA(Nz) = AA3 / Dx
    JR(Nz) = NN + I
    JC(Nz) = I+1
```

```
    !U_XX(I-1)
    Nz = Nz + 1
    AA(Nz) = BB1
    JR(Nz) = NN + I
    JC(Nz) = NN + I -1
```

```
    !U_XX(I)
    Nz = Nz + 1
    AA(Nz) = 1.0_DP
    JR(Nz) = NN + I
    JC(Nz) = NN + I
```

```
    !U_XX(I+1)
    Nz = Nz + 1
    AA(Nz) = BB3
    JR(Nz) = NN + I
    JC(Nz) = NN + I + 1
```

```
    Src(NN+I) = (CC3*U(I+1) - CC1*U(I) ) / Dx
```

```
END DO
```

```
!For I = N
!U_X(I-1)
Nz = Nz + 1
AA(Nz) = 2.5_DP / Dx
JR(Nz) = NN + NN
JC(Nz) = NN-1
```

```
!U_XX(I-1)
Nz = Nz + 1
AA(Nz) = 2.75_DP
JR(Nz) = NN + NN
```

```
JC(Nz) = NN + NN-1

!U_XX(I)
Nz = Nz + 1
AA(Nz) = 1.0_DP
JR(Nz) = NN + NN
JC(Nz) = NN + NN

Src(NN+NN) = (23.0_DP * U(N) + 199.0_DP * U(N-1) - 107.0_DP * U(N-2) + 5.0_DP * U(N-3) ) / Dx /
48.0_DP

END SUBROUTINE ASSIGN_A_JR_JC_CCD_FV_U
```