BUG REPORT

Talat Odman
Georgia Tech

September 30, 2009

Synopsis: In CMAQ, the advection process may be incomplete when it is synchronized with other processes. This problem may occur when advection time step is shorter than the synchronization time step, more likely in upper layers but also in the boundary layer.

Description: The main purpose of SUBROUTINE ADVSTEP is to determine the frequency of calls to process modules, a.k.a. the synchronization time step (SYNC). The characteristic time for advection, which is equal to the grid spacing divided by the maximum wind speed, plays an important role in determining SYNC. The Courant number, an important dimensionless parameter, is equal to the wind speed multiplied by the time step and divided by the grid spacing. In CMAQ, the advection time step (ADV) is selected such that the Courant number is less than 0.75. SYNC is set equal to ADV as long as the following conditions are satisfied.

1) SYNC must divide the output time step, STEP, evenly.
2) SYNC must be smaller than MAXSYNC (default value is 720 seconds if not user defined)
3) SYNC must be larger than MINSECS (default value is 60 seconds if MINSYNC is not user defined)

But, if ADV < MINSECS, then SYNC is set equal to MINSECS and advection is sub-cycled, i.e., applied several times before other processes are applied. The following piece of code is intended to do this.

```
DO REP = 1, STEP
   IF ( MOD( STEP, REP ) .EQ. 0 ) THEN  ! make TSTEP(2) divide TSTEP(1)
      SYNC = STEP / REP
      IF ( SYNC .LE. MAXSYNC ) THEN        ! force max TSTEP(2)
         ADV = SYNC
         IF ( MXUOVDX * FLOAT( ADV ) .LT. CC ) THEN
            IF ( SYNC .GE. MINSECS ) THEN  ! force min TSTEP(2)
               NREPS = REP
               ELSE   ! make ADV divide TSTEP(2) evenly
               SYNC = MINSECS
               NREPS = STEP / MINSECS
               IF ( MOD ( MINSECS,ADV ) .EQ. 0 ) THEN ! make ADV
                  K = MINSECS / ADV            ! divide TSTEP(2)
                  ELSE
                  K = MINSECS / ADV + 1
                  END IF
               ADV = MINSECS / K
               ADJFLG = .TRUE.
               END IF
            GO TO 301
            END IF      ! if Courant condition satisfied
         END IF      ! if SYNC .le. MAXSYNC
      END IF      ! if REP divides STEP evenly
   END DO      ! step loop
```

However, there is a problem when SYNC is forced to equal MINSECS. Suppose MXUOVDX is 0.08. ADV that satisfies the Courant condition is 9 seconds. Note that 9 divides 3600 evenly (9x400=3600). But, since 9 seconds is smaller than MINSECS, which is equal to 60 seconds, SYNC is set to 60 seconds. The advection process has to be sub-cycled. Let’s see what the code above produces.

```
Top layer thru which sync step determined:  9

Synchronization step adjusted up to mimimum (SEC):     60
                                         (HHMMSS): 000100
Number of Synchronization steps: 60


Layer   Advection   per Sync
      Step (HHMMSS)  Step
  13      000000      61
  12      000001      31
  11      000002      21
  10      000003      16
   9      000008       7
   8      000008       7
   7      000008       7
   6      000008       7
   5      000008       7
   4      000008       7
   3      000008       7
   2      000008       7
   1      000008       7
```

For layers 1 through 9 the advection time step is 8 seconds and 7 sub-cycles are performed. Since 7x8=56, the advection process would not be applied for a period of 4 seconds out of every.

There is also a problem in layers above layer 9, where sub-cycling is more common. These layers are in the free troposphere where the wind speeds are much higher than the boundary layer. Here, wind speed over grid spacing increases by 0.08 in every layer above layer 9 (i.e., UOVDX(L) = MXUOVDX + (L-ADVLAYR)*MXUOVDX). Note that every 60 seconds, advection would be short by 12, 18, and 29 seconds in layers 10, 11, and 12, respectively. Advection would not be applied to layer 13 at all.

The source of the problem is the following piece of code.

```
T2 = TIME2SEC( TSTEP( 2 ) )
DO L = ADVLAYR + 1, NLAYS
   ADV = T2 + 1
   DO REP = 1, STEP
      ADV = ADV - 1    ! subtract 1 sec
      IF ( UOVDX( L ) * FLOAT( ADV ) .LT. CC ) THEN
         IF ( ADV .EQ. T2 ) THEN
            NADVS( L ) = T2 / ADV
            ELSE
            NADVS( L ) = T2 / ADV + 1
            END IF
         ASTEP( L ) = SEC2TIME( T2 / NADVS( L ) )
         GO TO 401
         END IF              !  if Courant condition satisfied
      END DO
```

Remedy: The problem in layers where SYNC is determined (layers 1 through 9) can be corrected as follows.

```
      DO REP = 1, STEP
         IF ( MOD( STEP, REP ) .EQ. 0 ) THEN  ! make TSTEP(2) divide TSTEP(1)
            SYNC = STEP / REP
            IF ( SYNC .LE. MAXSYNC ) THEN        ! force max TSTEP(2)
               ADV = SYNC
               IF ( MXUOVDX * FLOAT( ADV ) .LT. CC ) THEN
                  IF ( SYNC .GE. MINSECS ) THEN  ! force min TSTEP(2)
                     NREPS = REP
                     ELSE   ! make ADV divide TSTEP(2) evenly
                     SYNC = MINSECS
                     NREPS = STEP / MINSECS
                     DO SUB = 1, MINSECS
                        IF ( MOD ( MINSECS, SUB ) .EQ. 0 ) THEN
                           AD2 = MINSECS / SUB
                           IF ( AD2 .LE. ADV ) THEN
                              ADV = AD2
                              ADJFLG = .TRUE.
                              GO TO 301
                              END IF
                           END IF
                        END DO
                        GO TO 291
                     END IF   ! if/else
                  GO TO 301
                  END IF      ! if Courant condition satisfied
               END IF      ! if SYNC .le. MAXSYNC
            END IF      ! if REP divides STEP evenly
         END DO      ! step loop

 291     CONTINUE
C If got here: could not determine satisfactory advection time step.

         WRITE( *,94013 ) TSTEP( 1 ), MXUOVDX
         STOP

301      CONTINUE
```

The new variables SUB and AD2 are both INTEGER. There is also a slight problem in FORMAT statement labeled 94013, which can be corrected by commenting out the second continuation line as follows.

```
94013 FORMAT( / 5X, 'Could not determine Courant-condition safe sync
step',
     &          1X, 'for model step:', I7.6, ' HHMMSS',
!    &          1X, 'in layer:', I3
!    &          1X, '(Max vel) / (dX1 or dX2) =', 1PE10.3)
     &        / 5X, '(Max vel)/(dX) =', 1PE10.3)
```

The second problem in upper layers (layers 10 through 13) can be corrected as follows.

```
      T2 = TIME2SEC( TSTEP( 2 ) )
      DO L = ADVLAYR + 1, NLAYS
         DO REP = 1, T2
            IF ( MOD( T2, REP ) .EQ. 0 ) THEN ! MTO
               ADV = T2/REP     ! MTO
               IF ( UOVDX( L ) * FLOAT( ADV ) .LT. CC ) THEN
                  NADVS( L ) = REP
                  ASTEP( L ) = SEC2TIME( ADV )
                  GO TO 401
                  END IF            !  if Courant condition satisfied
               END IF   ! if REP divides STEP evenly
            END DO

C If you get here: could not determine satisfactory advection time
step.

         WRITE( *,94013 ) TSTEP( 1 ), L, UOVDX( L )
         STOP

401      CONTINUE

         END DO   ! layer loop
```

The corrected code produces the following output.

```
    Top layer thru which sync step determined:  9

     Synchronization step adjusted up to mimimum (SEC):     60
                                              (HHMMSS): 000100
     Number of Synchronization steps: 60


     Layer   Advection   per Sync
           Step (HHMMSS)  Step
       13      000001      60
       12      000002      30
       11      000003      20
       10      000004      15
        9      000006      10
        8      000006      10
        7      000006      10
        6      000006      10
        5      000006      10
        4      000006      10
        3      000006      10
        2      000006      10
        1      000006      10
```

Note that in layers 1 through 9 the advection time step is 6 seconds and 10 sub-cycles are performed. Since 6x10=60, the advection process is applied for the entire length of the synchronization time step. In upper layers, the advection time steps divide 60 evenly.