



UNITED STATES  
NUCLEAR REGULATORY COMMISSION  
WASHINGTON, D.C. 20555-0001

ERRATA  
July 1997

**Report Number:** NUREG/CR-6331, Rev. 1  
PNNL-10521

**Report Title:** Atmospheric Relative Concentrations in Building Wakes

**Prepared by:** Pacific Northwest National Laboratory

**Date Published:** May 1997

**Instructions:** An error has been found in Appendix A, "ARCON96 FORTRAN Computer Code".

Please remove pages 75/76, 77/78, 79/80, 119/120, 121/122, and 123/124 and replace them with the attached corrected pages.

DF02

250783



Publications Branch  
Office of Information Resources Management

9707290012 970731  
PDR NUREG  
CR-6331 R PDR

A.1 ARCON96

```

C*****
C
C   ARCON96
C
C   J. V. Ramsdell, Jr., C. A. Simonen
C   Pacific Northwest National Laboratory
C   P.O. Box 999
C   Richland, Washington  99352
C
C   Created:   April 24, 1996
C   Revised:   June 25, 1997
C
C   Description: Program to estimate X/Q for control room habitability
C                 assessments.
C
C   Relationship to other Modules:
C
C   Calls:      SCENREAD  --- Read user input
C               RECORD    --- Record model input in log file
C               INIT      --- Compute diffusion coefficients
C                       and other constants that don't change
C               METREAD   --- Read meteorological data
C               XOQCALC5  --- Calculate relative concentrations
C               PROC      --- Compute averages and accumulate
C                       counts in bins
C               SHIFT     --- Move old meteorological data and X/Qs
C                       to the beginning of the data arrays
C                       before reading new data
C               SUMMARY   --- Convert counts in bins to frequency
C                       distributions and determine 95th %
C                       X/Q values
C*****

```

```

IMPLICIT      NONE

INCLUDE       'PARAM.INC'
INCLUDE       'MET.INC'
INCLUDE       'SCENARIO.INC'
INCLUDE       'UNITS.INC'
INCLUDE       'XOQ.INC'

LOGICAL*1     UO

CHARACTER*8   RTIME
CHARACTER*10  RDATE

INTEGER*2     SHR, SMIN, SSEC, S100, EHR, EMIN, ESEC, E100
INTEGER*2     YY, MM, DD, HH, MIN, SS, HS

```

ARCON96

REAL            DETIME, PROCTIME, SUMTIME, XOQTIME

C    Get date and time of code execution

```
RDATE = ' / / '
RTIME = ' : : '
CALL GETDAT(YY,MM,DD)
CALL GETTIM(HH,MIN,SS,HS)
WRITE( RDATE(1:2), '(I2)' ) MM
WRITE( RDATE(4:5), '(I2)' ) DD
WRITE( RDATE(7:10), '(I4)' ) YY
WRITE( RTIME(1:2), '(I2.2)' ) HH
WRITE( RTIME(4:5), '(I2.2)' ) MIN
WRITE( RTIME(7:8), '(I2.2)' ) SS

DETIME = 0.0
PROCTIME = 0.0
XOQTIME = 0.0
SUMTIME = 0.0
```

C    Open Scenario Input File

```
OPEN( SCENUNIT, FILE=' ', STATUS='OLD', ERR=997 )
```

C    Read Scenario Data

```
CALL SCENREAD

IF( TEST_FLG ) THEN
  OPEN(UNIT=qa_unit,FILE=qa_file,status='unknown')
  WRITE(*,'(2X,A,A)' ) 'qa_file = ', qa_file
ENDIF
```

C    Open log file and output program name and run date

```
OPEN(UNIT=LOG_UNIT,FILE=LOG_FILE,STATUS='NEW', ERR=998)
```

```
WRITE(LOG_UNIT,1)
```

```
1 FORMAT(/
```

```
+ ' Program Title:  ARCON96.'//
+ ' Developed For:  U.S. Nuclear Regulatory Commission'//
+ '                 Office of Nuclear Reactor Regulation'//
+ '                 Division of Reactor Program Management'//
+ ' Date:           June 25, 1997   11:00 a.m. '//
+ ' NRC Contacts:   J. Y. Lee       Phone: (301) 415 1080'//
+ '                 e-mail: jyl1@nrc.gov '//
+ '                 J. J. Hayes    Phone: (301) 415 3167'//
+ '                 e-mail: jjh@nrc.gov '//
+ '                 L. A. Brown    Phone: (301) 415 1232'//
+ '                 e-mail: lab2@nrc.gov '//
+ ' Code Developer: J. V. Ramsdell  Phone: (509) 372 6316'//
+ '                 e-mail: j_ramsdell@pnl.gov'//
+ ' Code Documentation:  NUREG/CR-6331 Rev. 1 '//
```

```

+ ' The program was prepared for an agency of the United States',
+ ' Government. Neither '/' the United States Government nor any',
+ ' agency thereof, nor any of their '/' employees, makes any',
+ ' warranty, expressed or implied, or assumes any legal '/'
+ ' liability or responsibilities for any third party's use,',
+ ' or the results of such '/' use, of any portion of this',
+ ' program or represents that its use by such third '/' party',
+ ' would not infringe privately owned rights. '/' )

```

```

WRITE(LOG_UNIT, '(1X,A,2X,A,2X,A,2X,A)') 'Program Run',
+      RDATE, 'at', RTIME

```

C Record user input on log file

```
CALL RECORD
```

C Compute variables that are functions of the scenario and  
C stability but not functions of wind speed or direction

```
CALL INIT
```

C Read and process data

```
DO WHILE ( .NOT. END_MET )
```

C Check on status of met data file, open new file if necessary

```
INQUIRE( UNIT=MET_UNIT, OPENED=UO )
```

```
IF( .NOT. UO ) THEN
```

C Open Meteorological Data File if another file exists

```
IF( NEXT_MET .LE. NMETFILE ) THEN
```

```
OPEN( MET_UNIT, FILE=MET_FILE(NEXT_MET), STATUS='OLD',
+     ERR=999 )
```

```
NEXT_MET = NEXT_MET + 1
```

```
ELSE ! No more meteorological data files
```

```
END_MET = .TRUE.
```

```
ENDIF
```

```
ENDIF
```

```
CALL GETTIM(SHR, SMIN, SSEC, S100)
```

C Read meteorological data files (<= 10,000 hours)

```
CALL METREAD
```

```
CALL GETTIM( EHR, EMIN, ESEC, E100)
```

```
DETIME = DETIME + 3600.*(EHR-SHR) + 60.*(EMIN-SMIN)
```

```
+ (ESEC-SSEC) + (E100-S100)/100.
```

ARCON96

```
IF( TEST_FLG ) THEN
  WRITE(qa_unit, '(/2x,a)') 'ARCON95'
  WRITE(QA_UNIT, '(/5X,A,I6)') 'FIRST MET = ', FIRST_MET
  WRITE(QA_UNIT, '(5X,A,I6)') 'LAST MET = ', LAST_MET
ENDIF
```

C Compute hourly X/Q values

```
CALL GETTIM(SHR,SMIN,SSEC,S100)

WRITE(*,'(/A)') ' CALCULATING X/Q '

CALL XOQCALCS

CALL GETTIM(EHR, EMIN, ESEC, E100)
XOQTIME = XOQTIME + 3600.*(EHR-SHR) + 60.*(EMIN-SMIN)
          + (ESEC-SSEC) + (E100-S100)/100.
```

C Process first set of X/Q's -- set cfd limits, accumulate values

```
CALL GETTIM(SHR,SMIN,SSEC,S100)

WRITE(*,'(/A)') ' PROCESSING X/Q DATA'

CALL PROC

CALL GETTIM(EHR, EMIN, ESEC, E100)
PROCTIME = PROCTIME + 3600.*(EHR-SHR) + 60.*(EMIN-SMIN)
          + (ESEC-SSEC) + (E100-S100)/100.
```

```
IF( .NOT. END_MET ) THEN
```

C Move residual met. data AND X/Q's to front of data arrays

```
CALL SHIFT

ENDIF
```

```
ENDDO ! End of data input and processing
```

C Summarize and output results of calculations

```
CALL GETTIM(SHR,SMIN,SSEC,S100)

WRITE(*,'(/A)') ' SUMMARIZING CALCULATIONS '

CALL SUMMARY

CALL GETTIM(EHR, EMIN, ESEC, E100)
SUMTIME = 3600.*(EHR-SHR) + 60.*(EMIN-SMIN)
          + (ESEC-SSEC) + (E100-S100)/100.
```

C Normal end of program

```
IF( TEST_FLG ) THEN
  WRITE(qa_unit,'(/2x,a)') 'ARCON96'
  WRITE( QA_UNIT,100) DETIME, XOQTIME, PROCTIME, SUMTIME
100  FORMAT( /5X, 'DATA ENTRY TIME (SEC)           = ', F10.2,
.        /5X, 'X/Q COMPUTATION TIME (SEC)        = ', F10.2,
.        /5X, 'DATA PROCESSING TIME (SEC)       = ', F10.2,
.        /5X, 'SUMMARY TIME (SEC)              = ', F10.2 )
  CLOSE( QA_UNIT )
ENDIF

WRITE(LOG_UNIT,'(2X,/A)') ' NORMAL PROGRAM COMPLETION '

GO TO 1000

C   Error end of program

997 STOP   ' UNABLE TO OPEN SCENARIO INPUT DATA FILE '
998 STOP   ' UNABLE TO OPEN LOG FILE ... CHECK FOR EXISTING NAME'
999 WRITE( LOG_UNIT,'(2X,/A)')
      +   ' UNABLE TO OPEN METEOROLOGICAL DATA FILE '

1000 CONTINUE

END
```

ARCON96

A.2 BLOCK DATA

BLOCK DATA

INCLUDE 'PARAM.INC'  
INCLUDE 'MET.INC'  
INCLUDE 'SCENARIO.INC'  
INCLUDE 'UNITS.INC'  
INCLUDE 'XOQ.INC'

C Set I/O units

DATA SCENUNIT / 1 /, MET\_UNIT / 2 /, LOG\_UNIT / 3 /  
DATA QA\_UNIT / 10 /, CFD\_UNIT / 11 /

C Initialize the met data counters

DATA END\_MET / .FALSE. /, NEXT\_MET / 1 /, TOT\_MET / 0 /  
+ FIRST\_MET / 1 /, LAST\_MET / MAXHOURS /

C Initialize the XOQ counters

DATA FIRST\_XOQ / 0 /, LAST\_XOQ / 0 /, TOT\_XOQ / 0 /,  
+ calm\_xoq / 0 /, in\_sect / 0 /, out\_sect / 0 /, elevated / 0 /

C Wind Speed Conversion Factors

DATA CF / 1.0, .447, .5144 /

C Initialize the upper and lower X/Q bounds

DATA CLMAX / 0.0 /, CLMIN / 1.0 /, SAMAX / 0.0 /, SAMIN / 1.0 /

C Set flag for calculation of bin limits

DATA LIMITS / .TRUE. /

END

A.16 XOQCALC5

SUBROUTINE XOQCALC5

```

C*****
C
C   XOQCALC5
C
C   J. V. Ramsdell, Jr.
C   Pacific Northwest National Laboratory
C   P.O. Box 999
C   Richland, WA 99352
C
C   Created: December 16, 1994
C   Revised: June 24, 1997
C
C   Description: Computes hourly centerline and sector-average X/Q
C                 for control room habitability assessments using the
C                 final version of the 1995 model for diffusion in
C                 the vicinity of buildings.
C
C   Other modules required:
C
C       Calls:      WINDYXOQ, CURVEFIX, WAKECORR
C*****

```

```

IMPLICIT NONE

INCLUDE 'PARAM.INC'
INCLUDE 'MET.INC'
INCLUDE 'SCENARIO.INC'
INCLUDE 'UNITS.INC'
INCLUDE 'XOQ.INC'

INTEGER*2 I1, DIR

REAL      PI, RHU, U10, TESTHT, cxoq, sxoq, eff_ht, dwash

PI = 3.141593

C ** Add space to QA file if QA output selected

IF( TEST_FLG ) WRITE(QA_UNIT, '( /2X,A/ )' ) 'Subroutine XOQCALC5'

DO I1 = FIRST_MET, LAST_MET

C ** Set Rhu and U10 to missing

      u10 = 999.0
      rhu = 999.0

```



## ARCON96

C \*\* See if stability class missing ... if it is skip X/Q calculations

IF( (STAB(I1) .LT. 1) .OR. (STAB(I1) .GT. 7) ) THEN

C \*\* Stability class missing

CLXOQ(I1) = -1.0  
SAXOQ(I1) = -1.0  
MISS\_XOQ = MISS\_XOQ + 1

C \*\* if QA output selected write out status

IF( TEST\_FLG ) THEN  
WRITE(QA\_UNIT, '(2X,I5,2X,I2,3OX,2(1PE10.2))') I1, stab(I1),  
& clxoq(I1), saxoq(I1)  
ENDIF

CYCLE  
ENDIF

C \*\* Check for missing wind speeds

IF( (SPDL(I1) .EQ. 9999) .AND. (SPDU(I1) .EQ. 9999) ) THEN

C \*\* Wind speeds both missing no X/Q calculations possible

CLXOQ(I1) = -1.0  
SAXOQ(I1) = -1.0  
MISS\_XOQ = MISS\_XOQ + 1

C \*\* if QA output selected write out status

IF( TEST\_FLG ) THEN  
WRITE(QA\_UNIT, '(2X,I5,2X,I2,2X,2F6.2,16X,2(1PE10.2))')  
& I1, stab(I1), u10, rhu, clxoq(I1), saxoq(I1)  
ENDIF

CYCLE

ELSE IF( (SPDL(I1) .NE. 9999) .AND. (SPDU(I1) .NE. 9999) ) THEN

C \*\* both wind speeds good convert speed entries to m/s

U10 = (SPDL(I1) / 10.0) \* CF( SPD\_TYPE ) \* C1(STAB(I1))

C \*\* Convert upper speed

TESTHT = ( MHT1 + MHT2 ) / 2.0  
IF( RHT .LE. TESTHT ) THEN  
RHU = (SPDL(I1) / 10.0) \* CF( SPD\_TYPE ) \* C2(STAB(I1))  
ELSE  
RHU = (SPDU(I1) / 10.0) \* CF( SPD\_TYPE ) \* C4(STAB(I1))  
ENDIF

```

ELSE IF( SPDL(I1) .NE. 9999 ) THEN

C **   Lower level good but not upper level

      U10 = (SPDL(I1) / 10.0) * CF( SPD_TYPE ) * C1(STAB(I1))
      RHU = (SPDL(I1) / 10.0) * CF( SPD_TYPE ) * C2(STAB(I1))

ELSE IF( SPDU(I1) .NE. 9999 ) THEN

C **   Upper speed good but not lower speed

      U10 = (SPDU(I1) / 10.0) * CF( SPD_TYPE ) * C3(STAB(I1))
      RHU = (SPDU(I1) / 10.0) * CF( SPD_TYPE ) * C4(STAB(I1))

ENDIF

C **   Check to see if calm

      IF( ( (rtype .LE. 2) .AND.
&         ((u10 .LT. umin) .OR. (rhu .LT. umin)) ) .OR.
&         ( (rtype .GE. 2) .AND. (rhu .LT. umin) ) )
&         THEN

C **   Calm winds

      rhu = 0.0

C **   See if wind directions both missing ... if they are skip

      ELSE IF( (DIRL(I1) .EQ. 999) .AND. (DIRU(I1) .EQ. 999) ) THEN

C **   Wind directions both missing

      CLXOQ(I1) = -1.0
      SAXOQ(I1) = -1.0
      MISS_XOQ = MISS_XOQ + 1

C **   if QA output selected write out status

      IF( TEST_FLG ) THEN
        WRITE(QA_UNIT, '(2X,I5,2X,I2,2X,2f6.2,2X,2I6, 2X,
&          2(1PE10.2))' ) i1, stab(i1), u10, rhu, dirl(i1),
&          diru(i1), clxoq(i1), saxoq(i1)
      ENDIF

      CYCLE

C **   Establish wind direction

      ELSE IF( (DIRL(I1) .EQ. 999) .OR. (DIRU(I1) .EQ. 999) ) THEN
        DIR = MIN(DIRL(I1), DIRU(I1))

```

ARCON96

```
      ELSE
        IF( RHT .LT. TESTHT ) THEN
          DIR = DIRL(I1)
        ELSE
          DIR = DIRU(I1)
        ENDIF
      ENDIF

C ** calculate downwash (negative if w0 < 1.5 rhu)

      IF( rhu .GE. UMIN ) THEN
        dwash = AMIN1( 4.0 * srad * (w0 / rhu - 1.5), 0.0)
      ELSE
        dwash = 0.0
      ENDIF

C ** calculate effective release height

      IF( rtype .EQ. 1 ) THEN
        eff_ht = rht - recht + t_diff
      ELSE
        eff_ht = AMAX1(rht + dwash, 0.0) - recht + t_diff
      ENDIF

C ** do X/Q calculation

      IF( RHU .LT. UMIN ) THEN

C ** calm ... call windyxoq with umin

        calm_xoq = calm_xoq + 1

        CALL WINDYXOQ( rtype, barea, dist, eff_ht, stab(i1), umin,
&                    nsigy(stab(i1)), nsigz(stab(i1)), sigz0, sigy0,
&                    f0, w0, sw_cnst, cxoq, sxoq )
        clxoq(i1) = cxoq
        saxoq(i1) = sxoq

C ** Determine if wind blowing from release point to receptor

        ELSE IF( DIR1 .LE. DIR2 ) THEN

C ** Direction window doesn't straddle north

          IF( (DIR1 .LE. DIR) .AND. (DIR .LE. DIR2) ) THEN

C ** Wind direction is in window -- calculate X/Q

            in_sect = in_sect + 1

            CALL WINDYXOQ( rtype, barea, dist, eff_ht, stab(i1), rhu,
+                          nsigy(stab(i1)), nsigz(stab(i1)), sigz0,
&                          sigy0, f0, w0, sw_cnst, cxoq, sxoq)
```

```

        CLXOQ(I1) = cxoq
        SAXOQ(I1) = sxoq

    ELSE

C **      Wind direction is outside window

        out_sect = out_sect + 1

        CLXOQ(I1) = 0.0
        SAXOQ(I1) = 0.0

    ENDIF

ELSE

C **      Wind direction window straddles north

        IF( (DIR .LE. DIR2) .OR. (DIR .GE. DIR1) ) THEN

C **      Wind direction is in window -- calculate X/Q

        in_sect = in_sect + 1

        CALL WINDYXOQ( rtype, barea, dist, eff_ht, stab(i1), rhu,
+                    nsigy(stab(i1)), nsigz(stab(i1)), sigz0,
+                    sigy0, f0, w0, sw_cnst, cxoq, sxoq)

        CLXOQ(I1) = cxoq
        SAXOQ(I1) = sxoq

    ELSE

C **      Wind direction is outside of the window

        out_sect = out_sect + 1

        CLXOQ(I1) = 0.0
        SAXOQ(I1) = 0.0
    ENDIF

ENDIF          ! end of X/Q calculations

C **      if QA output selected write out status

        IF( TEST_FLG ) THEN
            WRITE(QA_UNIT, '(2X,I5,2X,I2,2X,2f6.2,2X,i6,8X,
&                2(1PE10.2))') i1, stab(i1), u10, rhu, dir,
&                clxoq(i1), saxoq(i1)
        ENDIF

ENDDO          ! met data do loop

```

ARCON96

C \*\* Update maximum and minimum values

```
DO I1=FIRST_MET, LAST_MET
  IF( CLXOQ(I1) .GT. 0.0 ) THEN
    CLMAX = AMAX1( CLMAX, CLXOQ(I1) )
    CLMIN = AMIN1( CLMIN, CLXOQ(I1) )
    SAMAX = AMAX1( SAMAX, SAXOQ(I1) )
    SAMIN = AMIN1( SAMIN, SAXOQ(I1) )
  ENDIF
ENDDO

FIRST_XOQ = FIRST_MET
LAST_XOQ = LAST_MET
TOT_XOQ = TOT_XOQ + (LAST_XOQ - FIRST_XOQ) + 1

RETURN
END
```

120555139531 1 1AN1RR1RE1RH1  
US NRC-OIRM  
PUBLICATIONS BRANCH  
TPS-FOP-NUREG  
WHEN-687  
WASHINGTON

DC 20555