NASA Contractor Report 181791

# Noise Produced by Turbulent Flow Into a Rotor: Users Manual for Atmospheric Turbulence Prediction and Mean Flow and Turbulence Contraction Prediction


J. C. Simonich and B. Caplin

United Technologies Research Center
East Hartford, CT



Contract NAS1-17763

June 1989



National Aeronautics and
Space Administration

**Langley Research Center**
Hampton, Virginia 23665-5225

Table of Contents

Rotor Mean Flow Programs



Atmospheric Turbulence Properties Programs

```
*        ROUTINE - ROTMAN
*
*        PURPOSE - THIS PROGRAM COMPUTES STREAMLINES, TIME LINES AND
*                  TURBULENCE DEFORMATION TENSORS FOR THE FLOW THROUGH A
*                  HELICOPTER ROTOR MOVING IN THE ATMOSPHERE
*
*        AUTHOR  - J.C. SIMONICH
*
*        INPUT
*          NOTE: EITHER METERS OR FEET CAN BE USED FOR INPUT
*          AS LONG AS THE USER IS CONSISTENT THROUGHOUT THE INPUTS.
*
*          USER PARAMETERS
*            TITLE
*            NSS
*            NISO
*            ISTR,  ITIME
*            XMIN,  XMAX,  YMIN,  YMAX,  ZMIN,  ZMAX
*            MU  ,  OMR,   ALPHA, VO,    R
*            VINF
*            RERROR,AERROR,DS
*            RECPOL
*            NR,    NT              OR  NX,  NY,  NZ
*            R2                     OR  X1,  X2,  Y1,  Y2,  Z1,  Z2
*            IH
*
*            NAME      TYPE     DESCRIPTION
*            ..................................................................
*            TITLE     ALPHA    60 ALPHA ALPHANUMERIC TITLE
*            NSS       I        RELATIVE TIME STEP RESOLUTION
*            NISO      I        NUMBER OF STREAMLINES TO COMPUTE
*            ISTR      I        FLAG FOR OUTPUTING A STREAMLINE FILE
*                               FOR PLOTTING
*                               0 FOR NO OUTPUT FILE
*                               1 FOR STREAMLINE FILE
*            ITIME     I        FLAG FOR OUTPUTING A DEFORMATION TENSOR FILE
*                               0 FOR NO OUTPUT FILE
*                               1 FOR DEFORMATION TENSOR FILE
*                               (NOTE: EITHER ISTR OR ITIME CAN BE 1 BUT
*                               NOT BOTH AT THE SAME TIME)
*            XMIN      RD       STREAMLINE DOMAIN LIMIT (PROGRAM WILL
*                               STOP IF IT ATTEMPTS TO COMPUTE OUTSIDE
*                               OF THIS DOMAIN)
*            XMAX      RD       STREAMLINE DOMAIN LIMIT
*            YMIN      RD       STREAMLINE DOMAIN LIMIT
*            YMAX      RD       STREAMLINE DOMAIN LIMIT
*            ZMIN      RD       STREAMLINE DOMAIN LIMIT
*            ZMAX      RD       STREAMLINE DOMAIN LIMIT
*            MU        RD       ROTOR ADVANCE RATIO
*            OMR       RD       ROTOR TIP SPEED (ROTATIONAL FREQUENCY
*                               TIMES RADIUS)
*            ALPHA     RD       ROTOR TIP PATH PLANE ANGLE OF ATTACK
*            VO        RD       INDUCED VELOCITY AT THE ROTOR
*            R         RD       ROTOR RADIUS
*            VINF      RD       V COMPONENT VELOCITY TO BE ADDED TO
```

```
*                               HOVER CASE TO SIMULATE VERTICAL ASCENT
*                               OR DESCENT
*          RERROR    RD         RELATIVE ERROR ALLOWED IN ODE SOLVER
*          AERROR    RD         ABSOLUTE ERROR ALLOWED IN ODE SOLVER
*          DS        RD         DELTA DISTANCE BETWEEN STREAMLINES USED
*                               TO COMPUTE THE DEFORMATION TENSOR, IN
*                               RADIUS UNITS
*          RECPOL    ALPHA      COORDINATE SPECIFICATION TRIGGER
*                               'P' FOR POLAR SPECIFICATION OF
*                                STREAMLINE STARTING COORDINATES
*                                'C' FOR CARTESIAN SPECIFICATION OF
*                                STREAMLINE STARTING COORDINATES
*          NX        I           NUMBER OF X COORDINATE SPECIFICATION
*                                POINTS
*          NY        I           NUMBER OF Y COORDINATE SPECIFICATION
*                                POINTS
*          NZ        I           NUMBER OF Z COORDINATE SPECIFICATION
*                                POINTS
*          X1        RD          MINIMUM X VALUE FOR CARTESIAN
*                                SPECIFICATION
*          X2        RD          MAXIMUM X VALUE FOR CARTESIAN
*                                SPECIFICATION
*          Y1        RD          MINIMUM Y VALUE FOR CARTESIAN
*                                SPECIFICATION
*          Y2        RD          MAXIMUM Y VALUE FOR CARTESIAN
*                                SPECIFICATION
*          Z1        RD          MINIMUM Z VALUE FOR CARTESIAN
*                                SPECIFICATION
*          Z2        RD          MAXIMUM Z VALUE FOR CARTESIAN
*          NR        I           NUMBER OF RADIAL SPECIFICATION POINTS
*          NT        I           NUMBER OF ANGULAR SPECIFICATION POINTS
*          R2        RD          MAXIMUM RADIUS FOR POLAR SPECIFICATION
*          IH        I           0 FOR HOMOGENOUS CASE
*                                1 FOR NONHOMOGENOUS CASE
*
*      OUTPUT
*
*        DATA MEMBER ROTNOP(ROTOT1)   IF TIME = 1
*          TITLE
*          NR,NT,IH
*          R2,ALPHA,R
*          STRENG
*          TSAV,ZZP(1),ZZP(2),ZZP(3),XTSAV,YTSAV,ZTSAV--|
*          XSL1,YSL1,ZSL1,XSL2,YSL2,ZSL2               |
*          USAV,VSAV,WSAV                              |
*          X1A1,X2A1,X3A1                              |--- FOR FIRST
*          X1A2,X2A2,X3A2                              |    STREAMLINE
*          X1A3,X2A3,X3A3                            --|
*          TSAV,ZZP(1),ZZP(2),ZZP(3),XTSAV,YTSAV,ZTSAV--|
*          XSL1,YSL1,ZSL1,XSL2,YSL2,ZSL2               |
*          USAV,VSAV,WSAV                              |
*          X1A1,X2A1,X3A1                              |--- FOR SECOND
*          X1A2,X2A2,X3A2                              |    STREAMLINE
*          X1A3,X2A3,X3A3                            --|
*          .
```

```
*           .
*           .
*
*           NAME      TYPE     DESCRIPTION
*           ............................................................
*           TITLE     ALPHA    60 CHARACTER ALPHANUMERIC TITLE
*           NR        I        NUMBER OF RADIAL SPECIFICATION POINTS
*           NT        I        NUMBER OF ANGULAR SPECIFICATION POINTS
*           IH        I        0 FOR HOMOGENOUS CASE
*                              1 FOR NONHOMOGENOUS CASE
*           R2        RD       MAXIMUM RADIUS FOR POLOR SPECIFICATION
*           ALPHA     RD       SEE USER PARAMETER INPUT
*           R         RD       SEE USER PARAMETER INPUT
*           STRENG    RD       VORTEX CIRCULATION STRENGTH
*           TSAV      RD       STREAMLINE DRIFT TIME
*           ZZP(1)    RD       U VELOCITY COMPONENT IN RADII/SEC
*           ZZP(2)    RD       V VELOCITY COMPONENT IN RADII/SEC
*           ZZP(3)    RD       W VELOCITY COMPONENT IN RADII/SEC
*           XTSAV     RD       DOWNSTREAM X COORDINATE OF STREAMLINE IN
*                              ROTOR PLANE
*           YTSAV     RD       DOWNSTREAM Y COORDINATE OF STREAMLINE IN
*                              ROTOR PLANE
*           ZTSAV     RD       DOWNSTREAM Z COORDINATE OF STREAMLINE IN
*                              ROTOR PLANE
*           XSL1      RD       DOWNSTREAM X COORDINATE OF STREAMLINE IN
*                              STANDARD COORDINATE SYSTEM
*           YSL1      RD       DOWNSTREAM Y COORDINATE OF STREAMLINE IN
*                              STANDARD COORDINATE SYSTEM
*           ZSL1      RD       DOWNSTREAM Z COORDINATE OF STREAMLINE IN
*                              STANDARD COORDINATE SYSTEM
*           XSL2      RD       UPSTREAM X COORDINATE OF STREAMLINE IN
*                              STANDARD COORDINATE SYSTEM
*           YSL2      RD       UPSTREAM Y COORDINATE OF STREAMLINE IN
*                              STANDARD COORDINATE SYSTEM
*           ZSL2      RD       UPSTREAM Z COORDINATE OF STREAMLINE IN
*                              STANDARD COORDINATE SYSTEM
*           USAV      RD       UPSTREAM U VELOCITY COMPONENT IN
*                              RADII/SEC
*           VSAV      RD       UPSTREAM V VELOCITY COMPONENT IN
*                              RADII/SEC
*           WSAV      RD       UPSTREAM W VELOCITY COMPONENT IN
*                              RADII/SEC
*           X1A1, ETC RD       DEFORMATION TENSOR
*
*         DATA MEMBER ROTNOP(ROTOT2)  IF ISTR = 1
*           TITLE
*           NISO,NSS
*           XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX
*           XPLOT1,YPLOT1,ZPLOT1
*           XPLOT2,YPLOT2,ZPLOT2
*           XPLOT3,YPLOT3,ZPLOT3
*           .
*           .
*           .
*
```

```
*          NAME       TYPE      DESCRIPTION
*          ...........................................................
*          TITLE      ALPHA     60 CHARACTER ALPHANUMERIC TITLE
*          NISO       I         NUMBER OF STREAMLINES TO COMPUTE
*          NSS        I         RELATIVE TIME STEP RESOLUTION
*          XMIN       RD        STREAMLINE DOMAIN LIMIT
*          XMAX       RD        STREAMLINE DOMAIN LIMIT
*          YMIN       RD        STREAMLINE DOMAIN LIMIT
*          YMAX       RD        STREAMLINE DOMAIN LIMIT
*          ZMIN       RD        STREAMLINE DOMAIN LIMIT
*          XPLOT      RD        X STREAMLINE COORDINATE (RADII)
*          YPLOT      RD        Y STREAMLINE COORDINATE (RADII)
*          ZPLOT      RD        Z STREAMLINE COORDINATE (RADII)
*
*      LOCAL VARIABLES
*
*          NAME       TYPE      DESCRIPTION
*          ...........................................................
*          ABSERR     RD        ABSOLUTE ERROR ALLOWED IN ODE
*                               SOLVER (CHANGED BY ODE)
*          COSA       RD        COSINE OF ALPHA
*          COSNA      RD        COSINE OF NEGATIVE ALPHA
*          DR         RD        STEP SIZE FOR RADIUS
*          DT         RD        STEP SIZE IN TIME FOR EACH INTEGRATION
*                               ALONG THE STREAMLINE
*          DTHETA     RD        STEP SIZE FOR POLAR ANGLE
*          DX         RD        SIZE OF THE STEP IN THE X DIRECTION
*          DY         RD        SIZE OF THE STEP IN THE Y DIRECTION
*          DZ         RD        SIZE OF THE STEP IN THE Z DIRECTION
*          I          I         DO LOOP COUNTER
*          IARRAY     I         STORAGE AREA FOR INTEGER VARIABLES BEING
*                               READ IN FROM DATA MEMBER
*          IDIR       I         DIRECTION FOR STREAMLINE CALCULATION
*                               1 FOR FORWARD IN TIME
*                              -1 FOR BACKWARD IN TIME
*          IFLAG      I         ERROR FLAG FROM ROUTINE ODE
*          IFMT       I         INDICATOR OF DATA MEMBER RECORD FORMAT
*          INFILE     NAME      INPUT FILE NAME
*          IPASS      I         COUNTER FOR STREAMLINE CALCULATION PASSES
*          ITYPE      I         DATA TYPE CODE FOR USER PARAMETER
*          IWORK      I         INTEGER WORK ARRAY FOR SUBROUTINE ODE
*          J          I         DO LOOP COUNTER
*          K          I         DO LOOP COUNTER
*          L          I         NUMBER OF POINTS IN DEFORMATION TENSOR
*          MEXIST     I         INDICATOR IF DATA MEMBER EXISTS
*          MNR        I         NUMBER OF VALUES IN DATA MEMBER RECORD
*          NEL        I         NUMBER OF ARRAY ELEMENTS
*          NEQN       I         NUMBER OF SIMULTANEOUS EQUATIONS TO BE
*                               SOLVED BY SUBROUTINE ODE (3 STREAMLINE
*                               EQUATIONS)
*          NERROR     I         ERROR COUNTER FOR CALLS TO ODE
*          PI         RD        3.14159...
*          RAD        RD        RADIUS OF A POINT ON THE ROTOR DISK
*          RARRAY     RD        STORAGE AREA FOR REAL VARIABLES, BEING
*                               READ IN FROM DATA MEMBER
```

```
*          RELERR     RD        RELATIVE ERROR ALLOWED IN ODE SOLVER
*                               (CHANGED BY ODE)
*          ROTOT1     RD        ARRAY CONTAINING DATA MEMBER NAME ROTNOP(ROTOT1)
*          ROTOT2     RD        ARRAY CONTAINING DATA MEMBER NAME ROTNOP(ROTOT2)
*          SINA       RD        SINE OF ALPHA
*          SINNA      RD        SINE OF NEGATIVE ALPHA
*          STIME      RD        STREAMLINE DRIFT TIME
*          T          RD        CURRENT TIME IN CALLS TO THE ODE SOLVER
*          THETA      RD        POLAR ANGLE OF A POINT ON THE ROTOR DISK
*          TOUT       RD        ENDING TIME FOR CURRENT TIME STEP
*                               INTEGRATION
*          TT         RD        DUMMY VARIABLE (TIME) - NOT USED IN THIS
*                               ROUTINE BUT REQUIRED FOR COMPATIBILITY
*                               WITH SUBROUTINE ODE
*          UINF       RD        HORIZONTAL COMPONENT OF FREESTREAM VELOCITY
*          UOUT       RD        X COMPONENT OF VELOCITY
*          VOUT       RD        Y COMPONENT OF VELOCITY
*          WORK       RD        REAL WORK ARRAY FOR SUBROUTINE ODE
*          WOUT       RD        Z COMPONENT OF VELOCITY
*          X          RD        CURRENT X COORDINATE OF STREAMLINE
*                               POINT IN NORMAL CARTESIAN COORDINATES
*          XI         RD        ANGLE OF THE ROTOR WAKE FROM THE
*                               VERTICAL DIRECTION
*          XSL        RD        X COORDINATE OF STARTING STREAMLINE AT
*                               ROTOR DISK
*          XT         RD        X COORDINATE OF THE CURRENT POINT IN
*                               THE 'TILTED' COORDINATE SYSTEM
*          Y          RD        CURRENT Y COORDINATE OF STREAMLINE
*                               POINT IN NORMAL CARTESIAN COORDINATES
*          YSL        RD        Y COORDINATE OF STARTING STREAMLINE AT
*                               ROTOR DISK
*          YT         RD        Y COORDINATE OF THE CURRENT POINT IN
*                               THE 'TILTED' COORDINATE SYSTEM
*          Z          RD        CURRENT Z COORDINATE OF STREAMLINE
*                               POINT IN NORMAL CARTESIAN COORDINATES
*          ZSL        RD        Z COORDINATE OF STARTING STREAMLINE AT
*                               ROTOR DISK
*          ZT         RD        Z COORDINATE OF THE CURRENT POINT IN
*                               THE 'TILTED' COORDINATE SYSTEM
*          ZZ         RD        ARRAY OF LENGTH 3 CONTAINING CURRENT
*                               STREAMLINE POSITION COMPONENTS X,Y,Z
*
*        COMMON BLOCKS
*          /ROTCA/
*          STRENG,UINF,VINF,XI,R,ALPHA,PI,IDIR - described above
*
*          /ROTCB/
*          UOUT,VOUT,WOUT,T - described above
*
*          /ROTCC/
*          COSA,SINA,COSNA,SINNA - described above
*
*      FUNCTIONS
*        1.  NONDIMENSIONALIZES ALL LENGTH UNITS BY DIAMETER
*        2.  SETS UP SPECIFICATION OF STREAMLINE STARTING POINTS ON
```

```
*          ROTOR DISK
*          3.  CALLS ROTIND TO CALCULATE VORTEX STRENGTH
*          4.  OPENS DISK FILES TO SAVE STREAMLINE COORDINATES AND
*          DISTORTION TENSORS
*          5.  CALULATES VELOCITIES AND DISTORTION TENSORS
*
*        SUBPROGRAMS CALLED
*          ROTIND, ROTVEL, ROTODE, XSTORE, XFETCH, MMPUTR,
*          MMCLOS, MMOPWD, XASKP,  XGETP
*
*        CALLING SUBPROGRAMS
*          ANOP EXECUTIVE
*
*        ERRORS
*          NON-FATAL
*            1. CAN'T FIND VARIABLE IN USER PARAMETER TABLE
*            2. STIFF MATRIX RETURN FROM ROTODE
*            3. ERROR FROM MEMBER MANAGER
*          FATAL
*            NONE
*
*                             ROTMAN
*                               *
*                               *
*        *********************************************
*        *                      *                   *
*        *                      *                   *
*      ROTIND                 ROTODE              ROTVEL
*        *                      *
*        *                      *
*      ROTVIN                 ROTDE
*        *                      *
*        *                      *
*      ROTVEL          ********************
*        *             *        *       *
*        *             *        *       *
*    ***********     ROTTRP   ROTSTP  ROTVEL
*    *         *                *
*    *         *                *
*  ROTFNE   ROTFNK            ROTVEL
*
*
*        ENTRY
*           CONVERT ANGLES FROM DEGREES TO RADIANS
*           COMPUTE TRANSFORMATION PARAMETERS
*           NONDIMENSIONALIZE LENGTH UNITS BY DIAMETER
*           COMPUTE UINF AND XI FROM INPUT
*           COMPUTE TIME STEP SIZE FOR STREAMLINE INTEGRATION
*                   IF RECPOL .EQ. C
*                           THEN
*                           COMPUTE CARTESIAN STREAMLINE STARTING POINTS
*                              ON THE ROTOR DISK
*                   ELSE
*                           COMPUTE POLAR COORDINATE STREAMLINE STARTING
*                              POINTS ON THE ROTOR DISK
```

```
*               ENDIF
*           CALL ROTIND FOR COMPUTE THE WAKE VORTEX STRENGTH
*               IF ISTR .EQ. 1
*                   THEN
*                   OPEN A FILE TO SAVE STREAMLINE COORDINATES
*                   WRITE PRELIMINARY OUTPUT PARAMETERS
*               ENDIF
*               IF ITIME .EQ. 1
*                   THEN
*                   OPEN A FILE TO SAVE THE DISTORTION TENSORS
*                   WRITE PRELIMINARY OUTPUT PARA RECPOL .EQ. C
*                   THEN
*                   COMPUTE CARTESIAN STREAMLINE STARTING POINTS
*                      ON THE ROTOR DISK
*               ELSE
*                   COMPUTE POLAR COORDINATE STREAMLINE STARTING
*                      POINTS ON THE ROTOR DISK
*               ENDIF
*           CALL ROTIND FOR COMPUTE THE WAKE VORTEX STRENGTH
*               IF ISTR .EQ. 1
*                   THEN
*                   OPEN A FILE TO SAVE STREAMLINE COORDINATES
*                   WRITE PRELIMINARY OUTPUT PARAMETERS
*               ENDIF
*               IF ITIME .EQ. 1
*                   THEN
*                   OPEN A FILE TO SAVE THE DISTORTION TENSORS
*                   WRITE PRELIMINARY OUTPUT PARAMETERS
*               ENDIF
*           CALL ROTODE TO INTEGRATE STREAMLINE TO NEXT TIME STEP
*               IF IFLAG .NE. 2
*                   THEN
*                   INCREMENT ERROR COUNTER
*               ENDIF
*               IF NERROR .GT.100
*                   THEN
*                   STOP COMPUTATION
*               ENDIF
*               WRITE STREAMLINE POSITION TO DISK FILE
*               IF DISTANCE FROM ROTOR IS TOO LARGE
*                   THEN
*                   STOP COMPUTATION
*               ENDIF
*               IF CALCULATION GOES OUT OF AREA OF INTEREST
*                   THEN
*                   STOP COMPUTATION
*               ENDIF
*               GO TO 515 TO COMPUTE NEXT STREAMLINE
*               INCREMENT PASS COUNTER
*               DO CASE (1,601) (2,602) (3,603) (4,604) ON IPASS
*                  601  WRITE POSITIONS AND VELOCITIES TO DISK
*                   SET X STREAMLINE STARTING POSITION TO
*                     ORIGINAL X STREAMLINE STARTING
*                     POSITION PLUS DS
*                   SET Y STREAMLINE STARTING POSITION TO
```

```
*                            ORIGINAL Y STREAMLINE STARTING POSITION
*                          SET Z STREAMLINE STARTING POSITION TO
*                            ORIGINAL Z STREAMLINE STARTING POSITION
*                          SAVE STREAMLINE DRIFT TIME
*
*                    602   SET X STREAMLINE STARTING POSITION TO
*                            ORIGINAL X STREAMLINE STARTING POSITION
*                          SET Y STREAMLINE STARTING POSITION TO
*                            ORIGINAL Y STREAMLINE STARTING
*                            POSITION PLUS DS
*                          SET Z STREAMLINE STARTING POSITION TO
*                            ORIGINAL Z STREAMLINE STARTING POSITION
*                          COMPUTE DEFORMATION TENSOR FROM STREAMLINE
*                            DISPLACEMENT
*
*                    603   SET X STREAMLINE STARTING POSITION TO
*                            ORIGINAL X STREAMLINE STARTING POSITION
*                          SET Y STREAMLINE STARTING POSITION TO
*                            ORIGINAL Y STREAMLINE STARTING POSITION
*                          SET Z STREAMLINE STARTING POSITION TO
*                            ORIGINAL Z STREAMLINE STARTING
*                            POSITION PLUS DS
*                          COMPUTE DEFORMATION TENSOR FROM STREAMLINE
*                            DISPLACEMENT
*
*                    604   COMPUTE DEFORMATION TENSOR FROM STREAMLINE
*                            DISPLACEMENT
*                 5 ENDCASE
*         ENDDO
*      EXIT
*
***
```

```
*
*       ROUTINE - ROTDE
*
*       PURPOSE - ROTDE  MERELY ALLOCATES STORAGE FOR  ROTODE  TO RELIEVE
*         THE USER OF THE INCONVENIENCE OF A LONG CALL LIST.  CONSEQUENTLY
*         ROTDE  IS USED AS DESCRIBED IN THE COMMENTS FOR ROTODE .
*
*       AUTHOR - L.F. SHAMPINE AND  M.K. GORDON
*
*        THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
*        COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS:  THE INITIAL
*        VALUE PROBLEM  BY L. F. SHAMPINE AND M. K. GORDON.
*
*       VARIABLES
*
*           Name     Type     Description
*
*           ..............................................................
*           NEQN     I        No. of simultaneous eqns.
*           Y        RD       Soln. vector
*           T        RD       Independent variable
*           TOUT     RD       point at which soln. is desired
*           RELERR   RD       relative error criterion
*           ABSERR   RD       absolute error criterion
*            IFLAG   I        Integration status flag
*
*       SUBPROGRAMS CALLED
*         ROTVEL, ROTTRP, ROTSTP
*
*       CALLING SUBPROGRAM
*         ROTODE
*
***
```

```
*
*       ROUTINE - ROTFNE
*
*       PURPOSE -  RETURNS THE VALUE OF THE ELLIPTIC INTEGRAL OF THE
*         SECOND KIND
*
*       AUTHOR - J.C. SIMONICH
*
*       INPUT
*
*         ARGUMENTS
*
*           NAME      TYPE      DESCRIPTION
*           ...............................................................
*           X         RS        ARGUMENT OF ELLIPTIC INTEGRAL
*
*       OUTPUT
*
*         ARGUMENTS
*
*           NAME      TYPE      DESCRIPTION
*           ...............................................................
*           ROTFNE    RS        ELLIPTIC INTEGRAL OF THE 2ND KIND
*
*
*       LOCAL VARIABLES
*
*           NAME      TYPE      DESCRIPTION
*           ...............................................................
*           A         RS        "A" COEFICIENTS USED IN FIT TO FUNCTION
*           ASUM      RS        SUMMATION OF "A" TERMS
*           B         RS        "B" COEFICIENTS USED IN FIT TO FUNCTION
*           BSUM      RS        SUMMATION OF "B" TERMS
*           XD        RS        MODIFIED ELLIPTIC INTEGRAL ARGUMENT
*
*       FUNCTIONS
*         1. COMPUTES MODIFIED ELIPTIC INTEGRAL OF THE 2ND KIND
*
*       SUBPROGRAMS CALLED
*         NONE
*
*       CALLING SUBPROGRAMS
*         ROTVEL
*
*       ERRORS
*         NONE
*
*       ENTRY
*            COMPUTE ELLIPTIC INTEGRAL
*       EXIT
***
```

```
*        ROUTINE - ROTFNK
*
*        PURPOSE -  RETURNS THE VALUE OF THE ELLIPTIC INTEGRAL OF THE
*          FIRST KIND
*
*        AUTHOR - J.C. SIMONICH
*
*        INPUT
*
*          ARGUMENTS
*
*            NAME      TYPE     DESCRIPTION
*            ..............................................................
*            X         RS       ARGUMENT OF ELLIPTIC INTEGRAL
*
*        OUTPUT
*
*          FUNCTION
*
*            NAME      TYPE     DESCRIPTION
*            ..............................................................
*            ROTFNK    RS       ELLIPTIC INTEGRAL OF THE 1ST KIND
*
*        LOCAL VARIABLES
*
*            NAME      TYPE     DESCRIPTION
*            ..............................................................
*            A         RS       "A" COEFICIENTS USED IN FIT TO FUNCTION
*            ASUM      RS       SUMMATION OF "A" TERMS
*            B         RS       "B" COEFICIENTS USED IN FIT TO FUNCTION
*            BSUM      RS       SUMMATION OF "B" TERMS
*            XD        RS       MODIFIED ELLIPTIC INTEGRAL ARGUMENT
*
*        FUNCTIONS
*          1. COMPUTES MODIFIED ELIPTIC INTEGRAL OF THE 1ST KIND
*
*        SUBPROGRAMS CALLED
*          NONE
*
*        CALLING SUBPROGRAMS
*          ROTVEL
*
*        ERRORS
*          NONE
*
*        ENTRY
*             COMPUTE ELLIPTIC INTEGRAL
*        EXIT
***
```

```
*        ROUTINE - ROTIND
*
*        PURPOSE - THIS SUBROUTINE CALCULATES THE STRENGTH OF THE WAKE VORTICES
*          IS REQUIRED TO MATCH THE INPUT INDUCED VELOCITY
*
*        AUTHOR - J.C. SIMONICH
*
*        INPUT
*
*          ARGUMENTS
*
*            NAME      TYPE     DESCRIPTION
*            ..........................................................
*            V0        RD       INDUCED VELOCITY
*
*          COMMON BLOCK ROTCA
*
*            NAME      TYPE     DESCRIPTION
*            ..........................................................
*            VINF      RD       VERTICAL COMPONENT VELOCITY TO BE ADDED TO
*                               HOVER CASE TO SIMULATE VERTICAL ASCENT
*                               OR DESCENT
*        OUTPUT
*
*          COMMON BLOCK ROTCA
*
*            NAME      TYPE     DESCRIPTION
*            ............................................................
*            STRENG    RD       COMBINED VORTEX CIRCULATION STRENGTH
*            IDIR      I        DIRECTION FOR STREAMLINE CALCULATION
*                               1  FOR FORWARD IN TIME
*                              -1 BACKWARD IN TIME
*
*        LOCAL VARIABLES
*
*            NAME      TYPE     DESCRIPTION
*            ...........................................................
*            ALPHA     RD       ROTOR TIP PATH PLANE ANGLE OF ATTACK
*            IDIR      I        DIRECTION FOR STREAMLINE CALCULATION
*                               1 FOR FORWARD IN TIME
*                              -1 FOR BACKWARD IN TIME
*            PI        RD       3.14159...
*            R         RD       ROTOR RADIUS
*            S1        RD       LEFT HAND SOURCE STRENGTH FOR SECANT METHOD
*            S2        RD       RIGHT HAND SOURCE STRENGTH FOR SECANT METHOD
*            SNEW      RD       NEW PREDICTION FOR SOURCE STRENGTH IN SECANT
*                               METHOD
*            UINF      RD       HORIZONTAL COMPONENT OF FREESTREAM VELOCITY
*            V1        RD       LEFT HAND INDUCED VELOCITY FOR SECANT METHOD
*            V2        RD       RIGHT HAND INDUCED VELOCITY FOR SECANT METHOD
*            VVINF     RD       ORIGINAL VALUE OF VERTICAL VELOCITY
*            XI        RD       ANGLE OF THE ROTOR WAKE FROM THE
*                               VERTICAL DIRECTION
*
*          COMMON BLOCKS
```

```
*          /ROTCA/
*            STRENG,UINF,VINF,XI,R,ALPHA,PI,IDIR - as described in SUBROUTINE
*            ROTMAN
*
*       FUNCTIONS
*         1.  CALCULATE COMBINED VORTEX CIRCULATION STRENGTH
*
*       SUBPROGRAMS CALLED
*         ROTVIN
*
*       CALLING SUBPROGRAMS
*         ROTMAN
*
*       ERRORS
*         NONE
*
*       ENTRY
*
*       SET INITIAL VORTEX STRENGTH
*       SAVE ORIGINAL VERTICAL VELOCITY
*       SET VERTICAL VELOCITY TO ZERO
*    9       DO WHILE ABS(V2/V0) .GT. 0.001
*               SET STRENG TO NEW GUESS FOR S2
*               COMPUTE V2 BY CALLING ROTVIN
*               COMPUTE NEW VORTEX STRENGTH BY SECANT METHOD
*               SET OLD VALUES OF STRENGTH AND INDUCED VELOCITY TO
*                   LAST COMPUTED VALUES
*           ENDDO
*       RESET ORIGINAL VERTICAL VELOCITY
*      EXIT
*
***
```

```
*
*       ROUTINE - ROTODE
*
*       PURPOSE - Integrate a system of first order ODE'S
*
*       AUTHOR -  L.F. SHAMPINE AND M.K. GORDON
*
*       DESCRIPTION
*
*         DOUBLE PRECISION SUBROUTINE ROTODE INTEGRATES A SYSTEM OF NEQN
*         FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM:
*              DY(I)/DT = F(T,Y(1),Y(2),...,Y(NEQN))
*              Y(I) GIVEN AT  T .
*         THE SUBROUTINE INTEGRATES FROM  T  TO  TOUT .  ON RETURN THE
*         PARAMETERS IN THE CALL LIST ARE SET FOR CONTINUING THE INTEGRATION.
*         THE USER HAS ONLY TO DEFINE A NEW VALUE  TOUT  AND CALL  ROTODE
*         AGAIN.
*
*         THE DIFFERENTIAL EQUATIONS ARE ACTUALLY SOLVED BY A SUITE OF CODES
*         DE ,  STEP , AND  INTRP .  ROTODE  ALLOCATES VIRTUAL STORAGE IN THE
*         ARRAYS  WORK  AND  IWORK  AND CALLS  DE .  DE  IS A SUPERVISOR WHICH
*         DIRECTS THE SOLUTION.  IT CALLS ON THE ROUTINES  STEP  AND  INTRP
*         TO ADVANCE THE INTEGRATION AND TO INTERPOLATE AT OUTPUT POINTS.
*         STEP  USES A MODIFIED DIVIDED DIFFERENCE FORM OF THE ADAMS PECE
*         FORMULAS AND LOCAL EXTRAPOLATION.  IT ADJUSTS THE ORDER AND STEP
*         SIZE TO CONTROL THE LOCAL ERROR PER UNIT STEP IN A GENERALIZED
*         SENSE.  NORMALLY EACH CALL TO  STEP  ADVANCES THE SOLUTION ONE STEP
*         IN THE DIRECTION OF  TOUT .  FOR REASONS OF EFFICIENCY  DE
*         INTEGRATES BEYOND  TOUT  INTERNALLY, THOUGH NEVER BEYOND
*         T+10*(TOUT-T), AND CALLS  INTRP  TO INTERPOLATE THE SOLUTION AT
*         TOUT .  AN OPTION IS PROVIDED TO STOP THE INTEGRATION AT  TOUT  BUT
*         IT SHOULD BE USED ONLY IF IT IS IMPOSSIBLE TO CONTINUE THE
*         INTEGRATION BEYOND  TOUT .
*
*         THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
*         COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS:  THE INITIAL
*         VALUE PROBLEM  BY L. F. SHAMPINE AND M. K. GORDON.
*
*         THIS CODE WAS UPDATED BY DOUG BAXTER ON 1-14-80 TO CORRECT AN
*         ERROR IN THE HANDLING OF IMPROPER PARAMETERS.
*
*         THE PARAMETERS REPRESENT:
*            NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED (I         )
*            Y(*) -- SOLUTION VECTOR AT T                 (RD     )
*            T -- INDEPENDENT VARIABLE                    (RD     )
*            TOUT -- POINT AT WHICH SOLUTION IS DESIRED   (RD     )
*            RELERR,ABSERR -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR LOCAL
*               ERROR TEST (RD     ).  AT EACH STEP THE CODE REQUIRES
*               DABS(LOCAL ERROR) .LE. DABS(Y)*RELERR + ABSERR
*               FOR EACH COMPONENT OF THE LOCAL ERROR AND SOLUTION VECTORS
*            IFLAG -- INDICATES STATUS OF INTEGRATION     (I         )
*            WORK(*)  (RD     )  -- ARRAYS TO HOLD INFORMATION INTERNAL TO
*            IWORK(*) (I         )    WHICH IS NECESSARY FOR SUBSEQUENT CALLS
*
*         FIRST CALL TO ROTODE --
```

```
*
*        THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR THE ARRAYS
*        IN THE CALL LIST,
*        Y(NEQN), WORK(100+21*NEQN), IWORK(5),
*        DECLARE  F  IN AN EXTERNAL STATEMENT, SUPPLY THE DOUBLE PRECISION
*        SUBROUTINE F(T,Y,YP)  TO EVALUATE
*           DY(I)/DT = YP(I) = F(T,Y(1),Y(2),...,Y(NEQN))
*           AND INITIALIZE THE PARAMETERS:
*           NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
*           Y(*) -- VECTOR OF INITIAL CONDITIONS
*           T -- STARTING POINT OF INTEGRATION
*           TOUT -- POINT AT WHICH SOLUTION IS DESIRED
*           RELERR,ABSERR -- RELATIVE AND ABSOLUTE LOCAL ERROR TOLERANCES
*           IFLAG -- +1,-1.  INDICATOR TO INITIALIZE THE CODE.  NORMAL INPUT
*              IS +1.  THE USER SHOULD SET IFLAG=-1 ONLY IF IT IS
*              IMPOSSIBLE TO CONTINUE THE INTEGRATION BEYOND  TOUT .
*        ALL PARAMETERS EXCEPT  F ,  NEQN  AND  TOUT  MAY BE ALTERED BY THE
*        CODE ON OUTPUT SO MUST BE VARIABLES IN THE CALLING PROGRAM.
*
*     OUTPUT FROM  ROTODE  --
*
*        NEQN -- UNCHANGED
*        Y(*) -- SOLUTION AT  T
*        T -- LAST POINT REACHED IN INTEGRATION.  NORMAL RETURN HAS
*           T = TOUT .
*        TOUT -- UNCHANGED
*        RELERR,ABSERR -- NORMAL RETURN HAS TOLERANCES UNCHANGED.  IFLAG=3
*           SIGNALS TOLERANCES INCREASED
*        IFLAG = 2 -- NORMAL RETURN.  INTEGRATION REACHED  TOUT
*              = 3 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE ERROR
*                     TOLERANCES TOO SMALL.  RELERR ,  ABSERR  INCREASED
*                     APPROPRIATELY FOR CONTINUING
*              = 4 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE MORE THAN
*                     500 STEPS NEEDED
*              = 5 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE EQUATIONS
*                     APPEAR TO BE STIFF
*              = 6 -- INVALID INPUT PARAMETERS (FATAL ERROR)
*              THE VALUE OF  IFLAG  IS RETURNED NEGATIVE WHEN THE INPUT
*              VALUE IS NEGATIVE AND THE INTEGRATION DOES NOT REACH  TOUT ,
*              I.E., -3, -4, -5.
*        WORK(*),IWORK(*) -- INFORMATION GENERALLY OF NO INTEREST TO THE
*           USER BUT NECESSARY FOR SUBSEQUENT CALLS.
*
*     SUBSEQUENT CALLS TO  ROTODE --
*
*       SUBROUTINE  ROTODE  RETURNS WITH ALL INFORMATION NEEDED TO CONTINUE
*       THE INTEGRATION.  IF THE INTEGRATION REACHED  TOUT , THE USER NEED
*      ONLY DEFINE A NEW  TOUT  AND CALL AGAIN.  IF THE INTEGRATION DID NOT
*      REACH  TOUT  AND THE USER WANTS TO CONTINUE, HE JUST CALLS AGAIN.
*      THE OUTPUT VALUE OF  IFLAG  IS THE APPROPRIATE INPUT VALUE FOR
*      SUBSEQUENT CALLS.  THE ONLY SITUATION IN WHICH IT SHOULD BE ALTERED
*      IS TO STOP THE INTEGRATION INTERNALLY AT THE NEW  TOUT , I.E.,
*      CHANGE OUTPUT  IFLAG=2  TO INPUT  IFLAG=-2 .  ERROR TOLERANCES MAY
*      BE CHANGED BY THE USER BEFORE CONTINUING.  ALL OTHER PARAMETERS MUST
*      REMAIN UNCHANGED.
```

```
*
*****************************************************************************
**     SUBROUTINES ROTDE AND ROTSTP CONTAIN MACHINE DEPENDENT CONSTANTS.   *
**     BE SURE THEY ARE SET BEFORE USING  ROTODE .                          *
*****************************************************************************
*
```

```
*        ROUTINE - ROTOUT
*
*        PURPOSE - READS IN DATA MEMBERS CREATED BY MODULE
*                  ROT AND PRINTS THEM TO STANDARD OUTPUT
*                  WITH PAGINATION
*
*       AUTHOR - BARRY CAPLIN
*
*       INPUT
*
*    NAME           TYPE                 DESCRIPTION
*.......................................................................
*
*    ARGUMENTS
*
*    IS             I                    streamline file flag
*                                        =1 if streamline output was requested
*
*    IT             I                    distortion tensor file flag
*                                        =1 if dist. ten. output was requested
*
*    OUTPUT
*
*    Contents of MEMBERS I02(I01M2) and I02(I02M1)
*    written to standard output
*
*    LOCAL VARIABLES
*
*    NAME           TYPE                 DESCRIPTION
*.......................................................................
*
*    NAME           I                    MM name array
*
*    IHDR           I                    MM header array
*
*    IA             I                    array for retrieval of integer records
*
*    SA             RS                   array for retrieval of RD records
*
*    SUBPROGRAMS CALLED
*
*    XFETCH, XSTORE, XPLAB, MMOPRD, MMREW, XPAGE, XXPLINE, MMGETR, MMCLOS
*
*    CALLING SUBPROGRAM
*
*    ROTMAN
*
*    ERRORS
*    NON-FATAL
*
*    1. MEMBER MANAGER ERROR
*
*    Entry
*      start:
*      if(streamline output requested) then
```

```
*          open streamline output member
*          goto read section
*        elseif(tensor output requested) then
*          open tensor output member
*          goto read section
*        else
*          return
*        endif
*
*        read section:
*        do until no more records
*          read record from member
*          write record to standard output
*        continue
*        set current member read flag to 'no read'
*        goto start
*      Exit
***
```

```
*         ROUTINE - ROTSTP
*
*         PURPOSE - DOUBLE PRECISION SUBROUTINE ROTSTP
*           INTEGRATES A SYSTEM OF FIRST ORDER ORDINARY
*           DIFFERENTIAL EQUATIONS ONE STEP, NORMALLY FROM X TO X+H, USING A
*           MODIFIED DIVIDED DIFFERENCE FORM OF THE ADAMS PECE FORMULAS.  LOCAL
*           EXTRAPOLATION IS USED TO IMPROVE ABSOLUTE STABILITY AND ACCURACY.
*           THE CODE ADJUSTS ITS ORDER AND STEP SIZE TO CONTROL THE LOCAL ERROR
*           PER UNIT STEP IN A GENERALIZED SENSE.  SPECIAL DEVICES ARE INCLUDED
*           TO CONTROL ROUNDOFF ERROR AND TO DETECT WHEN THE USER IS REQUESTING
*           TOO MUCH ACCURACY.
*
*         THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
*         COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS:  THE INITIAL
*         VALUE PROBLEM  BY L. F. SHAMPINE AND M. K. GORDON.
*
*         AUTHOR - L.F. SHAMPINE AND M.K. GORDON
*
*
*      THE PARAMETERS REPRESENT:
*       X -- INDEPENDENT VARIABLE                (RD     )
*       Y(*) -- SOLUTION VECTOR AT X             (RD     )
*       YP(*) -- DERIVATIVE OF SOLUTION VECTOR AT  X  AFTER SUCCESSFUL
*            STEP                                (RD     )
*       NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED (I         )
*       H -- APPROPRIATE STEP SIZE FOR NEXT STEP.  NORMALLY DETERMINED BY
*            CODE                                (RD     )
*       EPS -- LOCAL ERROR TOLERANCE.  MUST BE VARIABLE  (RD     )
*       WT(*) -- VECTOR OF WEIGHTS FOR ERROR CRITERION   (RD     )
*       START -- LOGICAL VARIABLE SET .TRUE. FOR FIRST STEP,  .FALSE.
*            OTHERWISE                           (LOGICAL*4)
*       HOLD -- STEP SIZE USED FOR LAST SUCCESSFUL STEP  (RD     )
*       K -- APPROPRIATE ORDER FOR NEXT STEP (DETERMINED BY CODE)
*       KOLD -- ORDER USED FOR LAST SUCCESSFUL STEP
*       CRASH -- LOGICAL VARIABLE SET .TRUE. WHEN NO STEP CAN BE TAKEN,
*             .FALSE. OTHERWISE.
*      THE ARRAYS  PHI, PSI  ARE REQUIRED FOR THE INTERPOLATION SUBROUTINE
*      INTRP.  THE ARRAY P IS INTERNAL TO THE CODE.  ALL ARE RD
*
*      INPUT TO  ROTSTP
*
*       FIRST CALL --
*
*      THE USER MUST PROVIDE STORAGE IN HIS DRIVER PROGRAM FOR ALL ARRAYS
*      IN THE CALL LIST, NAMELY
*
*        DIMENSION Y(NEQN),WT(NEQN),PHI(NEQN,16),P(NEQN),YP(NEQN),PSI(12)
*
*      THE USER MUST ALSO DECLARE  START  AND  CRASH  LOGICAL VARIABLES
*      AND  F  AN EXTERNAL SUBROUTINE, SUPPLY THE SUBROUTINE  F(X,Y,YP)
*      TO EVALUATE
*         DY(I)/DX = YP(I) = F(X,Y(1),Y(2),...,Y(NEQN))
*      AND INITIALIZE ONLY THE FOLLOWING PARAMETERS:
*         X -- INITIAL VALUE OF THE INDEPENDENT VARIABLE
*         Y(*) -- VECTOR OF INITIAL VALUES OF DEPENDENT VARIABLES
```

```
*        NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
*        H -- NOMINAL STEP SIZE INDICATING DIRECTION OF INTEGRATION
*             AND MAXIMUM SIZE OF STEP.  MUST BE VARIABLE
*        EPS -- LOCAL ERROR TOLERANCE PER STEP.  MUST BE VARIABLE
*        WT(*) -- VECTOR OF NON-ZERO WEIGHTS FOR ERROR CRITERION
*        START -- .TRUE.
*
*     ROTSTP  REQUIRES THE L2 NORM OF THE VECTOR WITH COMPONENTS
*     LOCAL ERROR(L)/WT(L)  BE LESS THAN  EPS  FOR A SUCCESSFUL STEP.  THE
*     ARRAY  WT  ALLOWS THE USER TO SPECIFY AN ERROR TEST APPROPRIATE
*     FOR HIS PROBLEM.  FOR EXAMPLE,
*        WT(L) = 1.0  SPECIFIES ABSOLUTE ERROR,
*              = DABS(Y(L))  ERROR RELATIVE TO THE MOST RECENT VALUE OF
*                   THE L-TH COMPONENT OF THE SOLUTION,
*              = DABS(YP(L))  ERROR RELATIVE TO THE MOST RECENT VALUE OF
*                   THE L-TH COMPONENT OF THE DERIVATIVE,
*              = DMAX1(WT(L),DABS(Y(L)))  ERROR RELATIVE TO THE LARGEST
*                   MAGNITUDE OF L-TH COMPONENT OBTAINED SO FAR,
*              = DABS(Y(L))*RELERR/EPS + ABSERR/EPS  SPECIFIES A MIXED
*                   RELATIVE-ABSOLUTE TEST WHERE  RELERR  IS RELATIVE
*                   ERROR,  ABSERR  IS ABSOLUTE ERROR AND  EPS =
*                   DMAX1(RELERR,ABSERR) .
*
*      SUBSEQUENT CALLS --
*
*     SUBROUTINE  ROTSTP  IS DESIGNED SO THAT ALL INFORMATION NEEDED TO
*     CONTINUE THE INTEGRATION, INCLUDING THE STEP SIZE  H  AND THE ORDER
*     K , IS RETURNED WITH EACH STEP.  WITH THE EXCEPTION OF THE STEP
*     SIZE, THE ERROR TOLERANCE, AND THE WEIGHTS, NONE OF THE PARAMETERS
*     SHOULD BE ALTERED.  THE ARRAY  WT  MUST BE UPDATED AFTER EACH STEP
*     TO MAINTAIN RELATIVE ERROR TESTS LIKE THOSE ABOVE.  NORMALLY THE
*     INTEGRATION IS CONTINUED JUST BEYOND THE DESIRED ENDPOINT AND THE
*     SOLUTION INTERPOLATED THERE WITH SUBROUTINE  INTRP .  IF IT IS
*     IMPOSSIBLE TO INTEGRATE BEYOND THE ENDPOINT, THE STEP SIZE MAY BE
*     REDUCED TO HIT THE ENDPOINT SINCE THE CODE WILL NOT TAKE A STEP
*     LARGER THAN THE  H  INPUT.  CHANGING THE DIRECTION OF INTEGRATION,
*     I.E., THE SIGN OF  H , REQUIRES THE USER SET  START = .TRUE. BEFORE
*     CALLING  ROTSTP  AGAIN.  THIS IS THE ONLY SITUATION IN WHICH  START
*     SHOULD BE ALTERED.
*
*     OUTPUT FROM  ROTSTP
*
*      SUCCESSFUL STEP --
*
*     THE SUBROUTINE RETURNS AFTER EACH SUCCESSFUL STEP WITH  START  AND
*     CRASH  SET .FALSE. .  X  REPRESENTS THE INDEPENDENT VARIABLE
*     ADVANCED ONE STEP OF LENGTH  HOLD  FROM ITS VALUE ON INPUT AND  Y
*     THE SOLUTION VECTOR AT THE NEW VALUE OF  X .  ALL OTHER PARAMETERS
*     REPRESENT INFORMATION CORRESPONDING TO THE NEW  X  NEEDED TO
*     CONTINUE THE INTEGRATION.
*
*      UNSUCCESSFUL STEP --
*
*     WHEN THE ERROR TOLERANCE IS TOO SMALL FOR THE MACHINE PRECISION,
*     THE SUBROUTINE RETURNS WITHOUT TAKING A STEP AND  CRASH = .TRUE. .
```

```
*       AN APPROPRIATE STEP SIZE AND ERROR TOLERANCE FOR CONTINUING ARE
*       ESTIMATED AND ALL OTHER INFORMATION IS RESTORED AS UPON INPUT
*       BEFORE RETURNING.  TO CONTINUE WITH THE LARGER TOLERANCE, THE USER
*       JUST CALLS THE CODE AGAIN.  A RESTART IS NEITHER REQUIRED NOR
*       DESIRABLE.
***
```

```
*        ROUTINE - ROTTRP
*
*        PURPOSE - THE METHODS IN SUBROUTINE  STEP  APPROXIMATE THE
*          SOLUTION NEAR  X BY A POLYNOMIAL.  SUBROUTINE  ROTTRP
*          APPROXIMATES THE SOLUTION AT XOUT  BY EVALUATING THE POLYNOMIAL
*          THERE.  INFORMATION DEFINING THIS POLYNOMIAL IS PASSED FROM
*          ROTSTP SO  ROTTRP  CANNOT BE USED ALONE.
*
*        THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
*        COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS:  THE INITIAL
*        VALUE PROBLEM  BY L. F. SHAMPINE AND M. K. GORDON.
*
*        AUTHOR - L.F. SHAMPINE AND M.K. GORDON
*
*        INPUT TO ROTTRP --
*
*          ALL FLOATING POINT VARIABLES ARE DOUBLE PRECISION
*          THE USER PROVIDES STORAGE IN THE CALLING PROGRAM FOR THE ARRAYS IN
*          THE CALL LIST
      DIMENSION Y(NEQN),YOUT(NEQN),YPOUT(NEQN),PHI(NEQN,16),PSI(12)
*          AND DEFINES
*             XOUT -- POINT AT WHICH SOLUTION IS DESIRED.
*          THE REMAINING PARAMETERS ARE DEFINED IN  STEP  AND PASSED TO  ROTTRP
*          FROM THAT SUBROUTINE
*
*        OUTPUT FROM  ROTTRP --
*
*          YOUT(*) -- SOLUTION AT  XOUT
*          YPOUT(*) -- DERIVATIVE OF SOLUTION AT  XOUT
*        THE REMAINING PARAMETERS ARE RETURNED UNALTERED FROM THEIR INPUT
*        VALUES.  INTEGRATION WITH  STEP  MAY BE CONTINUED.
***
```

```
*       ROUTINE - ROTVEL
*
*        PURPOSE - THIS SUBROUTINE CALCULATES THE U, V AND W VELOCITY COMPONENTS
*          AT A POINT (X,Y,Z).  THE ROTOR AND WAKE ARE REPRESENTED BY A
*          ARRAY OF 20 RING VORTICES WITH A SUPERIMPOSED FREE STREAM.
*          THE VORTICES ARE SPACED CLOSER TOGETHER NEAR THE ROTOR WHERE
*          GRADIENTS ARE LARGEST.  THE VELOCITY AT A POINT IS FOUND BY THE
*          SUPERPOSITION OF THE VELOCITY FIELD FROM EACH OF THE 20
*          VORTICES PLUS THE FREE STREAM
*
*        THE FORMULAS FOR THE VELOCITY FROM A RING VORTEX ARE TAKEN FROM
*        NACA TR 1184, "THE NORMAL COMPONENT OF THE INDUCED VELOCITY IN
*        THE VICINITY OF A LIFTING ROTOR AND SOME EXAMPLES OF ITS
*        APPLICATION", BY W. CASTLES, JR, AND J.H. DE LEEUW
*
*        AUTHOR - J.C. SIMONICH
*
*        INPUT
*
*          ARGUMENTS
*
*            NAME      TYPE     DESCRIPTION
*            ..............................................................
*            TT        RD       DUMMY VARIABLE (TIME) - NOT USED IN THIS
*                               ROUTINE BUT REQUIRED FOR COMPATIBILITY
*                               WITH SUBROUTINE ODE
*            ZZZ(1)    RD       X COORDINATE WHERE VELOCITY IS TO BE FOUND
*            ZZZ(2)    RD       Y COORDINATE WHERE VELOCITY IS TO BE FOUND
*            ZZZ(3)    RD       Z COORDINATE WHERE VELOCITY IS TO BE FOUND
*
*          COMMON BLOCK ROTCA
*
*            NAME      TYPE     DESCRIPTION
*            ..............................................................
*            ALPHA     RD       ROTOR TIP PATH PLANE ANGLE OF ATTACK
*            IDIR      I        DIRECTION FOR STREAMLINE CALCULATION
*                               1 FOR FORWARD IN TIME
*                              -1 FOR BACKWARD IN TIME
*            PI        RD       3.14159...
*            R         RD       ROTOR RADIUS
*            STRENG    RD       COMBINED VORTEX CIRCULATION STRENGTH
*
*            UINF      RD       HORIZONTAL COMPONENT OF FREESTREAM VELOCITY
*            VINF      RD       VERTICAL COMPONENT VELOCITY TO BE ADDED TO
*                               OR DESCENT
*            XI        RD       ANGLE OF THE ROTOR WAKE FROM THE
*                               VERTICAL DIRECTION
*
*          COMMON BLOCK ROTCC
*
*            NAME      TYPE     DESCRIPTION
*            ..............................................................
*            COSNA     RD       COSINE OF NEGATIVE ALPHA
*            SINNA     RD       SINE OF NEGATIVE ALPHA
*
```

```
*       OUTPUT
*
*         ARGUMENTS
*
*           NAME      TYPE     DESCRIPTION
*           .................................................................
*           ZZZP(1)   RD       U COMPONENT OF VELOCITY
*           ZZZP(2)   RD       V COMPONENT OF VELOCITY
*           ZZZP(3)   RD       W COMPONENT OF VELOCITY
*
*         COMMON BLOCK ROTCB
*
*           NAME      TYPE     DESCRIPTION
*           .................................................................
*           UOUT      RD       X COMPONENT OF VELOCITY
*           VOUT      RD       Y COMPONENT OF VELOCITY
*           WOUT      RD       Z COMPONENT OF VELOCITY
*
*       LOCAL VARIABLES
*
*           NAME      TYPE     DESCRIPTION
*           ................................................................
*           A         RD       FORMULA FROM CASTLES AND DE LEEUW
*           B         RD       FORMULA FROM CASTLES AND DE LEEUW
*           BP        RD       FORMULA FROM CASTLES AND DE LEEUW
*           C         RD       FORMULA FROM CASTLES AND DE LEEUW
*           COSA      RD       COSINE OF ALPHA
*           D         RD       FORMULA FROM CASTLES AND DE LEEUW
*           D1        RD       NONDIMENSIONAL SHORTEST DISTANCE FROM A
*                              POINT P TO A VORTEX RING
*           D2        RD       NONDIMENSIONAL LARGEST DISTANCE FROM A
*                              POINT P TO A VORTEX RING
*           DR        RD       DELTA DISTANCE BETWEEN VORTEX RINGS
*           DX        RD       X DISPLACEMENT DISTANCE BETWEEN VORTEX
*                              RINGS
*           DY        RD       Y DISPLACEMENT DISTANCE BETWEEN VORTEX
*                              RINGS
*           F         RD       FORMULA FROM CASTLES AND DE LEEUW
*           FP        RD       FORMULAS FROM CASTLES AND DE LEEUW
*           I         I        DO LOOP COUNTER
*           N         I        NUMBER OF VORTEX RINGS (20)
*           P         RD       EXPONENT IN POWER LAW FOR VORTEX
*                              DISPLACEMENT (3)
*           RI        RD       VORTEX RING DISPLACEMENT PARAMETER
*           RN        RD       NUMBER OF VORTEX RINGS (20)
*           RP        RD       RADIAL DISTANCE FROM A POINT P FROM THE
*                              AXIS OF A VORTEX RING
*           SINA      RD       SINE OF ALPHA
*           SS        RD       SINGLE RING VORTEX SOURCE STRENGTH
*           T         RD       TIME
*           TAU       RD       FORMULA FROM CASTLES AND DE LEEUW
*           U         RD       VELOCITY COMPONENT IN THE PLANE OF THE
*                              ROTOR
*           V         RD       VELOCITY COMPONENT PERPENDICULAR TO THE
*                              PLANE OF THE ROTOR
```

```
*          VR          RD          RADIAL COMPONENT OF VELOCITY INDUCED
*                                  AT A POINT P BY A VORTEX RING
*          VZ          RD          AXIAL COMPONENT OF VELOCITY INDUCED AT
*                                  A POINT P BY A VORTEX RING
*          W           RD          VELOCITY COMPONENT IN THE PLANE OF THE
*                                  ROTOR
*          X           RD          NONDIMENSIONALIZED RADIAL DISTANCE OF A
*                                  POINT FROM THE AXIS OF A VORTEX RING
*          XV          RD          INDIVIDUAL RING VORTEX COORDINATE IN
*                                  TILTED ROTOR COORDINATE SYSTEM
*          XVIN        RD          X COORDINATE OF INDIVIDUAL RING
*                                  VORTEX IN NORMAL CARTESIAN COORDINATE
*                                  SYSTEM (X AXIS ALONG FREE STREAM LINE
*                                  AT INFINITY)
*          XX          RD          X COORDINATE IN TITLED COORDINATE
*                                  SYSTEM WHERE VELOCITY IS TO BE FOUND
*          XV          RD          INDIVIDUAL RING VORTEX COORDINATE IN
*                                  TILTED ROTOR COORDINATE SYSTEM
*          YVIN        RD          Y COORDINATE OF INDIVIDUAL RING
*                                  VORTEX IN NORMAL CARTESIAN COORDINATE
*                                  SYSTEM (X AXIS ALONG FREE STREAM LINE
*                                  AT INFINITY)
*          YY          RD          Y COORDINATE IN TITLED COORDINATE
*                                  SYSTEM WHERE VELOCITY IS TO BE FOUND
*          Z           RD          NONDIMENSIONALIZED AXIAL DISTANCE OF A
*                                  POINT FROM THE PLANE OF A VORTEX RING
*          ZP          RD          ZP IS THE DISTANCE OF A POINT P FROM
*                                  THE PLANE OF A VORTEX RING
*          ZZ          RD          Z COORDINATE IN TITLED COORDINATE
*                                  SYSTEM WHERE VELOCITY IS TO BE FOUND
*
*        COMMON BLOCKS
*          /ROTCA/
*          STRENG,UINF,VINF,XI,R,ALPHA,PI,IDIR - described in SUBROUTINE ROTMA
*
*          /ROTCB/
*          UOUT,VOUT,WOUT,T - described in SUBROUTINE ROTMAN
*
*          /ROTCC/
*         COSA,SINA,COSNA,SINNA - described in SUBROUTINE ROTMAN
*
*      FUNCTIONS
*        1.  CALCULATES DISPLACEMENT OF RING VORTICES
*        2.  ROTATE COORDINATE SYSTEM INTO TILTED ROTOR
*        3.  CALCULATE VELOCITY CONTRIBUTION FROM INDIVIDUAL RING VORTEX
*        4.  SUM CONTRIBUTIONS FROM ALL RING VORTICES
*        5.  ROTATE COORDINATE SYSTEM BACK TO NORMAL
*
*      SUBPROGRAMS CALLED
*        ROTFNE, ROTFNK
*
*      CALLING SUBPROGRAMS
*        ROTODE
*
*      ERRORS
```

```
*          NONE
*
*        ENTRY
*        SET POSITION OF VORTEX RINGS
*        INITIALIZE VELOCITY CONTRIBUTIONS TO ZERO
*        TRANSFORM INPUT POSITION FROM NORMAL TO TILTED COORDINATES
*         DO WHILE I .LT. N+1
*            COMPUTE COORDINATES OF CENTER OF VORTEX RINGS
•            TRANSFORM VORTEX POSITIONS TO TILTED COORDINATE
*            COMPUTE SOURCE STRENGTH FOR VORTEX RING
*            COMPUTE NONDIMENSIONAL DISTANCE FROM POINT TO VORTEX RING
*               IF X EQUALS ZERO
*               THEN
*                  SET RADIAL COMPONENT TO ZERO
*                  COMPUTE SPECIAL CASE AXIAL VELOCITY
*               ELSE
*                  COMPUTE RADIAL VELOCITY
*                  COMPUTE AXIAL VELOCITY
*         ENDDO
*        TRANSFORM VELOCITES FROM TILTED TO NORMAL COORDINATE SYSTEM
*        EXIT
*
***
```

```
*        ROUTINE - ROTVIN
*
*        PURPOSE - THIS SUBROUTINE CALCULATES THE VELOCITY INDUCED
*          THROUGH THE ROTOR AS A FUNCTION OF VORTEX STRENGTH.  THE
*          INDUCED VELOCITY IS TAKEN TO BE THE AREA WEIGHTED AVERAGE OF 19
*          POINTS ON THE ROTOR DISK.  THE POINTS ARE EQUALLY SPACED OUT 60
*          DEGREES APART, AND 2/7, 4/7, AND 6/7 OF THE RADIUS OUT FROM THE
*          CENTER PLUS ONE POINT AT THE CENTER.
*
*        AUTHOR - J.C. SIMONICH
*
*        INPUT
*
*          COMMON BLOCK ROTCA
*
*            NAME       TYPE     DESCRIPTION
*            ...............................................................
*            PI         RD       3.14159...
*            R          RD       ROTOR RADIUS
*
*          COMMON BLOCK ROTCC
*
*            NAME       TYPE     DESCRIPTION
*            ...............................................................
*            COSNA      RD       COSINE OF NEGATIVE ALPHA
*            SINNA      RD       SINE OF NEGATIVE ALPHA
*
*        OUTPUT
*
*          ARGUMENTS
*
*            NAME       TYPE     DESCRIPTION
*            ..............................................................
*            VI         RD       AVERAGE INDUCED VELOCITY
*
*        LOCAL VARIABLES
*
*            NAME       TYPE     DESCRIPTION
*            ..............................................................
*            A          RD       AREA OF ROTOR DISK "PIE" SEGMENT
*            ALPHA      RS       ROTOR TIP PATH PLANE ANGLE OF ATTACK
*            COSA       RD       COSINE OF ALPHA
*            IDIR       I        DIRECTION FOR STREAMLINE CALCULATION
*                                 1 FOR FORWARD IN TIME
*                                -1 FOR BACKWARD IN TIME
*            NR         I        NUMBER OF RADIAL SPECIFICATION POINTS
*            NT         I        NUMBER OF ANGULAR SPECIFICATION POINTS
*            RAD        RD       RADIUS OF CONTRIBUTING POINT
*            SINA       RD       SINE OF ALPHA
*            STRENG     RD       COMBINED VORTEX CIRCULATION STRENGTH
*            T          RD       POLAR ANGLE
*            TT         RD       DUMMY VARIABLE (TIME) - NOT USED IN THIS
*                                ROUTINE BUT REQUIRED FOR COMPATIBILITY
*                                WITH SUBROUTINE ODE
*            UINF       RD       HORIZONTAL COMPONENT OF FREESTREAM VELOCITY
```

```
*        VINF      RD       VERTICAL COMPONENT VELOCITY TO BE ADDED TO
*                           HOVER CASE TO SIMULATE VERTICAL ASCENT
*                           OR DESCENT
*        XI        RD       ANGLE OF THE ROTOR WAKE FROM THE
*                           VERTICAL DIRECTION
*        ZZ(1)     RD       X COORDINATE WHERE VELOCITY IS TO BE FOUND
*        ZZ(2)     RD       Y COORDINATE WHERE VELOCITY IS TO BE FOUND
*        ZZ(3)     RD       Z COORDINATE WHERE VELOCITY IS TO BE FOUND
*        ZZP(1)    RD       U COMPONENT OF VELOCITY
*        ZZP(2)    RD       V COMPONENT OF VELOCITY
*        ZZP(3)    RD       W COMPONENT OF VELOCITY
*
*      COMMON BLOCKS
*        /ROTCA/
*        STRENG,UINF,VINF,XI,R,ALPHA,PI,IDIR - described in SUBROUTINE ROTMA
*
*        /ROTCC/
*        COSA,SINA,COSNA,SINNA - described in SUBROUTINE ROTMAN
*
*    FUNCTIONS
*      1.  CALCULATES AVERAGE INDUCED VELOCITY OVER ROTOR FACE
*
*    SUBPROGRAMS CALLED
*      ROTVEL
*
*    CALLING SUBPROGRAMS
*      ROTIND
*
*    ERRORS
*      NONE
*
*    ENTRY
*    SET INDUCED VELOCITY TO ZERO
*        DO WHILE NR .LT. 3
*            CALCULATE RADIUS OF POINT
*            DO WHILE NT .LT. 4
*                CALCULATE POLAR ANGLE OF POINT
*                CALCULATE AREA OF SEGMENT
*                TRANSFORM FROM POLAR TO CARTESIAN COORDINATES
*                CALCULATE THE VELOCITY
*            ENDDO
*        ENDDO
*    CALCULATE VELOCITY AT THE CENTER
*    ADD CONTRIBUTION OF CENTER TO SUM
*    COMPUTE AVERAGE AREA WEIGHTED INDUCED VELOCITY
*   EXIT
*
*..........................................................................
***
```

```
***
*       ROUTINE - ABL
*
*       PURPOSE - TO CALCULATE THE MEAN AND TURBULENCE PROPERTIES FOR
*                 AN ATMOSPHERIC BOUNDARY LAYER
*
*       AUTHOR - J.C. SIMONICH
*
*
*       INPUT
*
*         USER PARAMETERS
*
*           NAME      TYPE      DESCRIPTION                                DEFAULT
*           ...........................................................................
*           G         RS        GEOSTROPHIC WIND SPEED (M/S)                 3.89
*           L         RS        MONIN-OBUKHOV STABILITY LENGTH (M)           0.0
*                               (NOTE: FOR NEUTRAL CONDITIONS, INPUT
*                               0.0)
*           THETA     RS        GEOGRAPHIC LATITUDE IN DEGREES               45
*           ZO        RS        ROUGHNESS HEIGHT (M)                         0.02M
*           Z         RS        HEIGHT ABOVE GROUND (M)                     100M
*
*       OUTPUT
*
*         DATA MEMBER ROTNOP(ABLOT1)
*         NOTE: IF FUNCTION MODULES ROT AND NOP ARE TO BE RUN USING
*               THE OUTPUT FROM FUNCTION MODULE ABL, A LIBRARY FILE
*               MUST BE CREATED CONTAINING DATA MEMBERS ROTNOP(ABLOT1)
*               AND ROTNOP(ROTOT1).
*
*           G,LWX,WUINF
*
*           NAME      TYPE      DESCRIPTION                                DEFAULT
*           ...........................................................................
*           G         RS        GEOSTROPHIC WIND SPEED (M/S)                 5M/S
*           LWX       RS        CORRELATION LENGTH SCALE
*           WUINF     RS        VERTICAL COMPONENT OF THE RMS TURBULENCE
*                               NORMALIZED BY THE FREE STREAM VELOCITY
*
*         LOCAL VARIABLES
*
*           NAME      TYPE      DESCRIPTION                                DEFAULT
*           ...........................................................................
*           DELTA     RS        BOUNDARY LAYER THICKNESS (M)
*           FC        RS        CORIOLIS PARAMETER = 2 OMEGA SIN (THETA)  1.0E-5
*                               WHERE OMEGA IS THE ROTATIONAL SPEED
*                               AND THETA IS THE LATITUDE
*           IFMT      I         FORMAT CODE FOR DATA MEMBER RECORD
*           ISTAT     I         STATUS OF MM CALLS
*           K         RS        VON KARMAN'S CONSTANT                        0.41
*           MEXIST    I         STATUS OF DATA MEMBER NAME
*           MNR       I         NO. OF VALUES READ FROM DATA MEMBER RECORD
*           P         RS        EXPONENT FOR THE POWER LAW DISCRIPTION
*                               OF THE MEAN VELOCITY PROFILE
```

```
*           PI          RS         3.14159
*           USTR        RS         SKIN FRICTION VELOCITY (M/S)
*           W15         RS         VERTICAL COMPONENT OF TURBULENCE
*                                  INTENSITY AT 15% OF THE BOUNDARY LAYER
*                                  THICKNESS
*           ZT          RS         ZT IS THE TROPOPAUSE HEIGHT IN METERS     11000M
*
*       FUNCTIONS
*
*         1.  CALCULATE THE CORIOLIS PARAMETER, SKIN FRICTION VELOCITY,
*             BOUNDARY LAYER THICKNESS, TURBULENCE INTENSITY,
*             CORRELATION LENGTH SCALE, AXIAL AND RADIAL MACH NUMBERS,
*             AND NORMALIZED TURBULENCE INTENSITY
*
*       SUBPROGRAMS CALLED
*
*         ABLUNS, ABLNTL, ABLSTA, MMCLOS, MMOPWD, MMPUTR, MMVUM
*         XFETCH, XSTORE, XPSUBT, XPLAB,  XPAGE,  XASKP,  XGETP, XPUTP
*
*       CALLING SUBPROGRAMS
*
*         ANOP EXECUTIVE
*
*       ERRORS
*
*         NON-FATAL
*           1. ERROR FINDING DATA MEMBER FOR OUTPUT
*           2. ERROR OPENING DATA MEMBER FOR OUTPUT
*           3. ERROR FINDING PARAMETER IN USER PARAMETER TABLE
*         FATAL
*           NONE
*                                    ABL
*                                     *
*                                     *
*           ************************************************
*           *                         *                        *
*           *                         *                        *
*         ABLUNS                    ABLNTL                   ABLSTA
*           *                         *                        *
*           *                         *                        *
*         ABLFUN                    ABLFNT                   ABLFST
*
*
*    ENTRY
*           CALCULATE CORIOLIS PARAMETER
*           IF L .LT. 0.0 THEN USE UNSTABLE FORMULAE TO:
*                 CALCULATE SKIN FRICTION VELOCITY
*                 CALCULATE BOUNDARY LAYER THICKNESS
*                 CALCULATE TURBULENCE INTENSITY
*                 CALCULATE CORRELATION LENGTH SCALE
*           ENDIF
*           IF L .EQ. 0.0 THEN USE NEUTRAL FORMULAE TO:
*                 CALCULATE SKIN FRICTION VELOCITY
*                 CALCULATE BOUNDARY LAYER THICKNESS
*                 CALCULATE TURBULENCE INTENSITY
```

```
*                   CALCULATE CORRELATION LENGTH SCALE
*              ENDIF
*              IF L .GT. 0.0 THEN USE STABLE FORMULAE TO:
*                   CALCULATE SKIN FRICTION VELOCITY
*                   CALCULATE BOUNDARY LAYER THICKNESS
*                   CALCULATE TURBULENCE INTENSITY
*                   CALCULATE CORRELATION LENGTH SCALE
*              ENDIF
*              IF Z .GT. 15% OF DELTA THEN
*                   INTERPOLATE BETWEEEN THE VALUE AT 15% AND THE TOP
*                   OF THE BOUNDARY LAYER
*         EXIT
***
```

```
*        ROUTINE - ABLFNT
*
*        PURPOSE - TO CALCULATE THE GEOSTROPHIC DRAG LAW FUNCTION
*                  FOR THE NEUTRAL ATMOSPHERE
*
*        AUTHOR - J.C. SIMONICH
*
*        INPUT
*
*          ARGUMENTS
*
*            NAME      TYPE     DESCRIPTION                               DEFAULT
*            ..............................................................
*            FC        RS       CORIOLIS PARAMETER = 2 OMEGA SIN (THET)   1.0E-5
*                               WHERE OMEGA IS THE ROTATIONAL SPEED
*                               AND THET IS THE LATITUDE
*            ZO        RS       ROUGHNESS HEIGHT (M)                       0.02M
*            G         RS       GEOSTROPHIC WIND SPEED (M/S)               5M/S
*            K         RS       VON KARMAN'S CONSTANT                      0.41
*            USTR      RS       SKIN FRICTION VELOCITY (M/S)               0.1M/S
*            DELTA     RS       BOUNDARY LAYER THICKNESS (M)               600M
*            A         RS       CONSTANT A IN GEOSTROPHIC DRAG LAW         1.7
*            B         RS       CONSTANT B IN GEOSTROPHIC DRAG LAW         4.7
*
*        OUTPUT
*
*          FUNCTION
*
*            NAME      TYPE     DESCRIPTION
*            ..............................................................
*            ABLFNT    RS       GEOSTROPHIC DRAG LAW FUNCTION
*
*        FUNCTIONS
*          1.  COMPUTE GEOSTROPHIC DRAG LAW FUNCTION
*
*        SUBROUTINES CALLED
*          NONE
*
*        CALLING SUBPROGRAMS
*          ABLNTL
*
*        ERRORS
*          NONE
*
*        ENTRY
*           COMPUTE ABLFNT
*        EXIT
***
```

```
*       ROUTINE - ABLFST
*
*       PURPOSE - TO CALCULATE THE GEOSTROPHIC DRAG LAW FUNCTION
*
*       AUTHOR - J.C. SIMONICH
*
*       INPUT
*
*         ARGUMENTS
*
*           NAME      TYPE     DESCRIPTION                             DEFAULT
*           ..................................................................
*           B         RS       CONSTANT B IN GEOSTROPHIC DRAG LAW        1.1
*           L         RS       MONIN-OBUKHOV STABILITY LENGTH (M)        20M
*                              (NOTE: FOR NEUTRAL CONDITIONS, INPUT
*                              0.0)
*           FC        RS       CORIOLIS PARAMETER = 2 OMEGA SIN (THET)  1.0E-5
*                              WHERE OMEGA IS THE ROTATIONAL SPEED
*                              AND THET IS THE LATITUDE
*           ZT        RS       ZT IS THE TROPOPAUSE HEIGHT IN METERS    11000M
*           ZO        RS       ROUGHNESS HEIGHT (M)                     0.02M
*           G         RS       GEOSTROPHIC WIND SPEED (M/S)             5M/S
*           K         RS       VON KARMAN'S CONSTANT                    0.41
*
*       OUTPUT
*
*         ARGUMENTS
*
*           NAME      TYPE     DESCRIPTION
*           ..................................................................
*           ABLFST    RS       GEOSTROPHIC DRAG LAW FUNCTION
*           DELTA     RS       BOUNDARY LAYER THICKNESS (M)
*           USTR      RS       SKIN FRICTION VELOCITY (M/S)
*
*       FUNCTIONS
*         1.  COMPUTE BOUNDARY LAYER THICKNESS
*         2.  COMPUTE SKIN FRICTION VELOCITY
*         3.  COMPUTE CONSTANT A IN GEOSTROPHIC DRAG LAW
*         4.  COMPUTE GEOSTROPHIC DRAG LAW FUNCTION
*
*       SUBROUTINES CALLED
*         NONE
*
*       CALLING SUBPROGRAMS
*         ABLSTA
*
*       ERRORS
*         NONE
*
*       ENTRY
*          COMPUTE DELTA
*          COMPUTE USTR
*          COMPUTE A
*          COMPUTE ABLFST
*       EXIT
```

```
*        ROUTINE - ABLFUN
*
*        PURPOSE - TO CALCULATE THE GEOSTROPHIC DRAG LAW FUNCTION
*                  FOR THE UNSTABLE ATMOSPHERE
*
*
*        AUTHOR - J.C. SIMONICH
*
*        INPUT
*
*          ARGUMENTS
*            NAME      TYPE     DESCRIPTION                                DEFAULT
*            ..................................................................
*            L         RS       MONIN-OBUKHOV STABILITY LENGTH (M)          20M
*                               (NOTE: FOR NEUTRAL CONDITIONS, INPUT
*                               0.0)
*            FC        RS       CORIOLIS PARAMETER = 2 OMEGA SIN (THET)   1.0E-5
*                               WHERE OMEGA IS THE ROTATIONAL SPEED
*                               AND THET IS THE LATITUDE
*            Z0        RS       ROUGHNESS HEIGHT (M)                       0.02M
*            G         RS       GEOSTROPHIC WIND SPEED (M/S)               5M/S
*            K         RS       VON KARMAN'S CONSTANT                      0.41
*            USTR      RS       SKIN FRICTION VELOCITY (M/S)               0.1M/S
*            DELTA     RS       BOUNDARY LAYER THICKNESS (M)               600M
*
*        OUTPUT
*
*          FUNCTION
*
*            NAME      TYPE     DESCRIPTION
*            ..................................................................
*            ABLFUN    RS       GEOSTROPHIC DRAG LAW FUNCTION
*
*        FUNCTIONS
*          1.  COMPUTE CONSTANT A IN GEOSTROPHIC DRAG LAW
*          2.  COMPUTE CONSTANT B IN GEOSTROPHIC DRAG LAW
*          3.  COMPUTE GEOSTROPHIC DRAG LAW FUNCTION
*
*        SUBROUTINES CALLED
*          NONE
*
*        CALLING SUBPROGRAMS
*          ABLUNS
*
*        ERRORS
*          NONE
*
*        ENTRY
*            COMPUTE A
*            COMPUTE B
*            COMPUTE ABLFUN
*        EXIT
***
```

```
*        ROUTINE - ABLNTL
*
*        PURPOSE - TO PREDICT THE BOUNDARY LAYER THICKNESS AND SKIN FRICTION
*                  VELOCITY FOR THE NEUTRAL ATMOSPHERE
*
*        AUTHOR - J.C. SIMONICH
*
*        INPUT
*
*          ARGUMENTS
*
*            NAME       TYPE     DESCRIPTION                              DEFAULT
*
*            .................................................................
*            FC         RS       CORIOLIS PARAMETER = 2 OMEGA SIN (THET)  1.0E-5
*                                WHERE OMEGA IS THE ROTATIONAL SPEED
*            Z0         RS       ROUGHNESS HEIGHT (M)                      0.02M
*            G          RS       GEOSTROPHIC WIND SPEED (M/S)              5M/S
*            K          RS       VON KARMAN'S CONSTANT                     0.41
*
*        OUTPUT
*
*          ARGUMENTS
*
*            NAME       TYPE     DESCRIPTION
*
*            .................................................................
*            DELTA      RS       BOUNDARY LAYER THICKNESS (M)
*            USTR       RS       SKIN FRICTION VELOCITY (M/S)
*
*        LOCAL VARIABLES
*
*             NAME       TYPE     DESCRIPTION
*
*             ................................................................
*             A          RS       "CONSTANT" IN GEOSTROPHIC DRAG LAW
*             B          RS        "CONSTANT" IN GEOSTROPHIC DRAG LAW
*             F1         RS        GEOSTROPHIC DRAG LAW FUNCTION EVALUATED AT
*                                  LOWER LIMIT OF B
*             F3         RS        GEOSTROPHIC DRAG LAW FUNCTION EVALUATED AT
*                                  UPPER LIMIT OF B
*             USTR1      RS        LOWER LIMIT OF SKIN FRICTION
*                                  VELOCITY IN INTERVAL HALVING
*             USTR2      RS        UPPER LIMIT OF SKIN FRICTION
*                                  VELOCITY IN INTERVAL HAVLING
*
*        FUNCTIONS
*          1.  COMPUTE BOUNDARY LAYER THICKNESS
*          2.  COMPUTE SKIN FRICTION VELOCITY
*
*        SUBPROGRAMS CALLED
*          ABLFNT
*
*        CALLING SUBPROGRAMS
*          ABLMAN
*
*        ERRORS
*          NONE
```

```
*
*        ENTRY
*           COMPUTE BOUNDARY LAYER THICKNESS
*           COMPUTE CONSTANTS A AND B IN GEOSTROPHIC DRAG LAW
*           COMPUTE LOWER AND UPPER LIMITS ON USTR
*     10            DO WHILE ABS(F3) LE. 1.0E-4
*                       COMPUTE NEXT GUESS FOR USTR
*                       COMPUTE F1 BY CALLING ABLFNT
*                       COMPUTE F3 BY CALLING ABLFNT
*                             IF FA*F3 IS POSITIVE
*                             THEN USTR1=USTR
*                             ELSE USTR2=USTR
*                 ENDDO
*        EXIT
***
```

```
*         ROUTINE - ABLSTA
*
*         PURPOSE - TO PREDICT THE BOUNDARY LAYER THICKNESS AND SKIN FRICTION
*                   VELOCITY FOR THE STABLE ATMOSPHERE
*
*         AUTHOR - J.C. SIMONICH
*
*         INPUT
*
*           ARGUMENTS
*
*             NAME      TYPE     DESCRIPTION                                DEFAULT
*             ..................................................................
*             L         RS       MONIN-OBUKHOV STABILITY LENGTH (M)          20M
*                                (NOTE: FOR NEUTRAL CONDITIONS, INPUT
*                                0.0)
*             FC        RS       CORIOLIS PARAMETER = 2 OMEGA SIN (THET)   1.0E-5
*                                WHERE OMEGA IS THE ROTATIONAL SPEED
*                                AND THET IS THE LATITUDE
*             ZT        RS       ZT IS THE TROPOPAUSE HEIGHT IN METERS      11000M
*             Z0        RS       ROUGHNESS HEIGHT (M)                       0.02M
*             G         RS       GEOSTROPHIC WIND SPEED (M/S)               5M/S
*             K         RS       VON KARMAN'S CONSTANT                      0.41
*
*         OUTPUT
*
*           ARGUMENTS
*
*             NAME      TYPE     DESCRIPTION
*             ................................................................
*             DELTA     RS       BOUNDARY LAYER THICKNESS (M)
*             USTR      RS       SKIN FRICTION VELOCITY (M/S)
*
*         LOCAL VARIABLES
*
*             NAME      TYPE     DESCRIPTION
*             ................................................................
*             A         RS       "CONSTANT" IN GEOSTROPHIC DRAG LAW
*             B1        RS       LOWER LIMIT ON "CONSTANT" B IN
*                                INTERVAL HALVING
*             B2        RS       CURRENT GUESS FOR "CONSTANT" B IN
*                                INTERVAL HALVING
*             B3        RS       UPPER LIMIT ON "CONSTANT" B IN
*                                INTERVAL HALVING
*             B         RS       "CONSTANT" IN GEOSTROPHIC DRAG LAW
*             C         RS       "CONSTANT" IN GEOSTROPHIC DRAG LAW
*             F1        RS        GEOSTROPHIC DRAG LAW FUNCTION EVALUATED AT
*                                 LOWER LIMIT OF B
*             F3        RS        GEOSTROPHIC DRAG LAW FUNCTION EVALUATED AT
*                                 UPPER LIMIT OF B
*
*         FUNCTIONS
*           1.  COMPUTE BOUNDARY LAYER THICKNESS
*           2.  COMPUTE SKIN FRICTION VELOCITY
*
```

```
*        SUBPROGRAMS CALLED
*          ABLFST
*
*        CALLING SUBPROGRAMS
*          ABLMAN
*
*        ERRORS
*          NONE
*
*        ENTRY
*           COMPUTE CONSTANTS A AND C IN GEOSTROPHIC DRAG LAW
*           COMPUTE LOWER AND UPPER LIMITS ON CONSTANT B
*     10            DO WHILE ABS(F3) LE. 1.0E-4
*                       COMPUTE NEXT GUESS FOR B
*                       COMPUTE F1 BY CALLING ABLFST
*                       COMPUTE F3 BY CALLING ABLFST
*                             IF FA*F3 IS POSITIVE
*                             THEN B1=B3
*                             ELSE B2=B3
*                 ENDDO
*        EXIT
***
```

```
*        ROUTINE - ABLUNS
*
*        PURPOSE - TO PREDICT THE BOUNDARY LAYER THICKNESS AND SKIN FRICTION
*                  VELOCITY FOR THE UNSTABLE ATMOSPHERE
*
*        AUTHOR - J.C. SIMONICH
*
*        INPUT
*
*          ARGUMENTS
*
*            NAME      TYPE     DESCRIPTION                                 DEFAULT
*            ..................................................................
*            L         RS       MONIN-OBUKHOV STABILITY LENGTH (M)            20M
*                               (NOTE: FOR NEUTRAL CONDITIONS, INPUT
*                               0.0)
*            FC        RS      CORIOLIS PARAMETER = 2 OMEGA SIN (THET)  1.0E-5
*                              WHERE OMEGA IS THE ROTATIONAL SPEED
*                              AND THET IS THE LATITUDE
*            ZO        RS      ROUGHNESS HEIGHT (M)                        0.02M
*            G         RS      GEOSTROPHIC WIND SPEED (M/S)               5M/S
*            K         RS      VON KARMAN'S CONSTANT                      0.41
*
*        OUTPUT
*
*          ARGUMENTS
*
*            NAME      TYPE     DESCRIPTION
*            ..................................................................
*            DELTA     RS       BOUNDARY LAYER THICKNESS (M)
*            USTR      RS       SKIN FRICTION VELOCITY (M/S)
*
*
*        LOCAL VARIABLES
*
*            NAME      TYPE     DESCRIPTION
*            ..................................................................
*            F1        RS       VALUE OF GEOSTROPHIC DRAG LAW
*                               FUNCTION EVALUATED AT LOWER LIMIT OF B
*            F3        RS       VALUE OF GEOSTROPHIC DRAG LAW FUNCTION
*                               EVALUATED AT UPPER LIMIT OF B
*            USTR1     RS       LOWER LIMIT OF SKIN FRICTION
*                               VELOCITY IN INTERVAL HALVING
*            USTR2     RS       UPPER LIMIT OF SKIN FRICTION
*                               VELOCITY IN INTERVAL HAVLING
*
*        FUNCTIONS
*          1.  COMPUTE BOUNDARY LAYER THICKNESS
*          2.  COMPUTE SKIN FRICTION VELOCITY
*
*        SUBPROGRAMS CALLED
*          ABLFUN
*
*        CALLING SUBPROGRAMS
*          ABLMAN
```

```
*
*       ERRORS
*         NONE
*
*       ENTRY
*          SET BOUNDARY LAYER THICKNESS
*          SET LIMITS ON USTR FOR ITERATION
*    10            DO WHILE ABS(F3) LE. 1.0E-4
*                      COMPUTE NEXT GUESS FOR USTR
*                      COMPUTE F1 BY CALLING ABLFUN
*                      COMPUTE F3 BY CALLING ABLFUN
*                            IF FA*F3 IS POSITIVE
*                            THEN USTR1=USTR
*                            ELSE USTR2=USTR
*                ENDDO
*       EXIT
***
```

NASA
National Aeronautics and Space Administration

# Report Documentation Page

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| NASA CR-181791 | | |

| 4. Title and Subtitle | 5. Report Date |
|---|---|
| Noise Produced by Turbulent Flow Into a Rotor: Users Manual for Atmospheric Turbulence Prediction and Mean Flow and Turbulence Contraction Prediction | June 1989 |
| | 6. Performing Organization Code |
| 7. Author(s) | 8. Performing Organization Report No. |
| J. C. Simonich and B. Caplin | |
| | 10. Work Unit No. |
| | 505-63-51 |
| 9. Performing Organization Name and Address | 11. Contract or Grant No. |
| United Technologies Research Center<br>East Hartford, CT 06108 | NAS1-17763 |
| | 13. Type of Report and Period Covered |
| 12. Sponsoring Agency Name and Address | Contractor Report |
| National Aeronautics and Space Administration<br>Langley Research Center<br>Hampton, VA 23665-5225 | 14. Sponsoring Agency Code |

15. Supplementary Notes

Langley Technical Monitor: Thomas F. Brooks
Final Report

16. Abstract

This document describes a users manual for a computer program for predicting atmospheric turbulence and mean flow and turbulence contraction as part of a noise prediction scheme for nonisotropic turbulence ingestion noise in helicopters. The report includes descriptions of the various program modules and subroutines, their function, programming structure, and the required input and output variables. This routine is incorporated as one module of NASA's ROTONET helicopter noise prediction program.

| 17. Key Words (Suggested by Author(s)) | 18. Distribution Statement |
|---|---|
| noise prediction, helicopter noise, turbulence ingestion noise, rotor noise | Unclassified - Unlimited<br><br>Subject Category 71 |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | 44 | A03 |