SANDIA REPORT

SAND2013-xxxx
Unlimited Release
Printed June 2013

Sandia National Laboratories Environmental Fluid Dynamics Code: Marine Hydrokinetic Module User's Manual

Scott C. James

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: 1
Online order: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2013-xxxx
Unlimited Release
Printed June 2013

Sandia National Laboratories Environmental Fluid Dynamics Code: Marine Hydrokinetic Module User's Manual

Scott C. James
E^{x}ponent, Inc.
Civil Engineering Practice
320 Goddard Way, Suite 200
Irvine, CA 92618
sjames@exponent.com

Abstract

This document describes the marine hydrokinetic (MHK) input file and subroutines for the Sandia National Laboratories Environmental Fluid Dynamics Code (SNL-EFDC), which is a combined hydrodynamic, sediment transport, and water quality model based on the Environmental Fluid Dynamics Code (EFDC) developed by John *Hamrick* [1992], formerly sponsored by the U.S. Environmental Protection Agency, and now maintained by Tetra Tech, Inc. SNL-EFDC has been previously enhanced with the incorporation of the SEDZLJ sediment dynamics model developed by Ziegler, Lick, and Jones [*Jones*, 2001; *Jones and Lick*, 2001; *Ziegler and Lick*, 1986]. SNL-EFDC has also been upgraded to more accurately simulate algae growth with specific application to optimizing biomass in an open-channel raceway for biofuels production [*James and Boriah*, 2010]. A detailed description of the input file containing data describing the MHK device/array is provided, along with a description of the MHK FORTRAN routine. Both a theoretical description of the MHK dynamics as incorporated into SNL-EFDC and an explanation of the source code are provided. This user manual is meant to be used in conjunction with the original EFDC [*Hamrick*, 2007a] and sediment dynamics SNL-EFDC manuals [*Thanh et al.*, 2008].

Through this document, the authors provide information for users who wish to model the effects of an MHK device (or array of devices) on a flow system with EFDC and who also seek a clear understanding of the source code, which is available from staff in the Water Power Technologies Department at Sandia National Laboratories, Albuquerque, New Mexico.

Acknowledgement

The research and development described in this document was funded by the U.S. Department of Energy. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Contents

Introduction

Marine hydrokinetic (MHK) projects will generate power from currents, tides, or waves thereby altering water velocities and wave patterns in the site's waterway. These hydrodynamics changes can potentially affect the ecosystem, both near the MHK installation and in surrounding (i.e., far field) regions. In both marine and freshwater environments, devices will remove energy (momentum) from the flow with commensurate changes to turbulent kinetic energy and its dissipation rate, potentially altering sediment dynamics and water quality. In estuaries, tidal ranges and residence times could change (either increasing or decreasing depending on system flow properties and where the effects are being measured). Changes to flow rates and shear stresses could alter sediment dynamics. Modified flushing rates and residence times could have effects on water quality (e.g., nutrient concentrations and algae blooms). Effects will be proportional to the number and size of structures installed, with large MHK projects having the greatest potential effects and requiring the most in-depth analyses.

This manual describes modification to an existing flow, sediment dynamics, and water-quality code, Sandia National Laboratories Environmental Fluid Dynamics Code (SNL-EFDC) to qualify, quantify, and visualize the influence of MHK-device momentum/energy extraction at representative sites. New algorithms simulate changes to system fluid dynamics due to removal of momentum and reflect commensurate changes in turbulent kinetic energy and its dissipation rate. SNL-EFDC is a modified version of the EPA's public-domain surface-water flow, sediment transport, and water-quality model developed by John Hamrick while at the Virginia Institute of Marine Sciences [*Hamrick*, 1992]. EFDC is now proprietarily maintained by Tetra Tech, Inc. EFDC simulates flow and transport of sediment (in bedload and in suspended load), algae (including growth kinetics), and toxic substance (kinetic reactions and transport). SNL-EFDC improves EFDC with updated sediment dynamics [*James et al.*, 2010a; *Thanh et al.*, 2008], water-quality (algae growth) [*James and Boriah*, 2010], and, of particular interest here, MHK-device simulation [*James et al.*, 2010b; *James et al.*, 2011] subroutines.

The new SNL-EFDC MHK subroutine is *MHKPWR.f90*. Other standard EFDC routines were also modified to ensure that proper adjustments are made to the turbulent kinetic energy and its dissipation rate. Still other routines were adjusted in small ways to appropriately read in and pass MHK-relevant variables. The MHK subroutines are based on one primary input file (*MHK.inp*), but appropriate changes must also be made to *DXDY.INP* to properly locate the MHK devices within the EFDC model grid. The source code for SNL-EFDC is available at `http://sourceforge.net/projects/snl-efdc/`.

MHK Input File

MHK.INP

The example input file *MHK.inp* shown below (text in `Courier` font) contains all of the relevant data describing the MHK device including its support structure (note that long lines have been truncated). The first 29 lines are the file headers that describe what each input flag/variable represents. Line 26 names the four input flags; line 27 indicates that they are unitless. Table 1 describes the MHK-relevant flags.

```
C  MHK.INP file, in free format across line,
C
C  NMHKTYPE is the number of MHK types
C  NFLAGPWR is a flag to indicate power curve calculation type (1=calculate from…
C  UPSTREAM is a flag to indicate if you use the velocity in the cell of the…
C  OUTPUTFLAG (see below)
C
C  If NFLAGPWR=1 then input NMHKTYPE lines of the following variables:
C  WIDTHMHK is the width of MHK device type
C  WIDTHSUP is the width of MHK support structure type
C  BOFFMHK is the bottom offset of the MHK device type (how far from the bottom)
C  BOFFSUP is the bottom offset of the MHK support structure type
C  TOFFMHK is the top offset of the MHK device type
C  TOFFSUP is the top offset of the MHK support structure type
C  CTMHK is the thrust coefficient of MHK device type
C  CDSUP is the coefficient of power dissipation of MHK support structure type
C  VMINCUT is the minimum velocity cut-in for MHK device type power curve
C  VMAXCUT is the maximum velocity cut-out for MHK device type power curve
C  DENMHK is the number of MHK devices in a cell
C
C  If NFLAGPWR=2 then input NMHKTYPE entries of the following variables:
C  NPWRCRV is the number of points on user-defined power curve, by device type
C  VPWRCRV is the velocity value on user-defined power curve
C  PPWRCRV is the corresponding power dissipation value on user-defined power curve
C
C NMHKTYPE NFLAGPWR UPSTREAM OUTPUTFLAG
C   (-)       (-)     (-)        (-)
C WIDTHMHK WIDTHSUP BOFFMHK BOFFSUP TOFFMHK TOFFSUP CTMHK CDSUP VMINCUT VMAXCUT DENMHK
C   (m)      (m)      (m)     (m)     (m)     (m)    (-)   (-)   (m/s)   (m/s)    (-)
     1         1       1         0
    30.28     3.0    9.0      0.0    13.3     11.2    0.8   1.2     0.0    2.7    1.0
C BETAMHK_P BETAMHK_D CE4MHK PBCOEF these are turbulence coefficients…
  0.05      1.3       1.5     5
C NPWRCRV
C   (-)
C    5
C VPWRCRV PPWRCRV
C  (m/s)  (Watts)
C   0.0     0.0
C   0.5     0.0
C   1.0    1.0e4
C   2.0    1.0e5
C   3.0    1.0e6
OUTPUT FLAG
0 - no specific output
1 - energy fluxes across a transverse section upstream and downstream of a device…
2 - average velocity and the z-profile for the tidal reference model at the throat
3 - average velocity and surface velocity for the river reference model
4-7 - outputs for straight-channel calibration model of wake structure
```

Table 1: MHK input flags and their description.

Flag	Description
NMHKTYPE	The number of MHK types in the model. For each type, there must be a corresponding line of MHK device data. Essentially, this number tells SNL-EFDC how many lines of MHK input data to read starting on line 31.
NFLAGPWR	This instructs the model to use either a specified thrust coefficient that is multiplied by a velocity cubed, $U_{\infty}^{2}U$ to determine the amount of power withdrawn from the flow or whether to use a user-defined power curve. Currently, the user-defined power curve function is not fully operational because it has not been needed.
UPSTREAM	Instructs the power calculation to use the local cell velocity (when 0) or one cell upstream (when 1) for U_{∞}.
OUTPUTFLAG	This is for research purposes and instructs SNL-EFDC what results to print to output files (as specified in *tecplot.f90*).

MHK variables describing the MHK device itself and any corresponding support structure are listed on line 28; line 29 indicates the units of these variables. The variables are general enough to account for horizontal- or vertical-axis turbines as well as for devices that are emplaced at the bottom of the flow system or suspended from the water surface by a floating structure. Table 2 lists the MHK variables and their definitions that are read in as NMHKTYPE lines starting at line 31.

Table 2: MHK and support structure variable definitions.

Variable	Description
WIDTHMHK	Width (m) of the MHK turbine.
WIDTHSUP	Width (m) of the MHK support structure.
BOFFMHK	Bottom offset of the MHK turbine (m); distance from the bottom of the MHK turbine to the sediment bed.
BOFFSUP	Bottom offset of the MHK support structure (m); distance from the bottom of the MHK support structure to the sediment bed.
TOFFMHK	Top offset of the MHK turbine (m); distance from the top of the MHK turbine to the sediment bed.
TOFFSUP	Top offset of the MHK support structure (m); distance from the top of the MHK support structure to the sediment bed.
CTMHK	Coefficient of thrust for the MHK turbine.
CDSUP	Coefficient of drag for the MHK support structure.
VMINCUT	Cut-in velocity (m/s) for the MHK device; velocity at which the MHK turbine starts to rotate and generate electricity. This sets $U_{\infty} = 0$ for U_{∞} < VMINCUT.
VMAXCUT	Cut-out velocity (m/s) for the MHK turbine; velocity at which the MHK turbine will generate no additional power. This limits U_{∞} to VMAXCUT.
DENMHK	Density of the MHK device/support structure per cell. This is used when the device spans multiple cells or if multiple devices are present in a cell. For example, if a device spans four cells, DENMHK = 0.25. Similarly, in a model with large cells, there may be three turbines in a cell and in this case DENMHK = 3.0.
BETAMHK_P	The term accounting for wake turbulence representing the ratio of mean kinetic energy transferred directly into turbulence (should be between 0 and 1).
BETAMHK_D	The term accounting for the transfer of energy between large- and smaller-scale turbulence (the short circuit of the turbulence cascade).
CE4MHK	The term compensating for the production of turbulence with a commensurate increase in its dissipation.
PBCOEF	This partial blockage coefficient accounts for the physical displacement of fluid by the solid portions of the MHK device. Because SNL-EFDC implements MHK devices as if they were porous obstructions to flow, this term allows for additional flow re-routing around the device in

	the vertical to account for its solidity. None of the last four parameters are relevant in a 2D-horizontal (single-layer) model.

It is important to note that the MHK routine is designed after the vegetative resistance algorithms already present in EFDC. This afforded a more seamless integration of this new algorithm into the existing code. Specifically, the `ISVEG` flag should be turned on (set to 1) in EFDC Card 5 to enable MHK simulation capabilities. This instructs EFDC to read an additional column in the *DXDY.INP* that specifies the cells where vegetation is present (`VEG TYPE` > 1). The value of `veg TYPE` (EFDC variable name: `MVEGL(L))` specifies the type of vegetation present. When this number (`MVEGL(L)`) is greater than 90, an MHK device is present in this cell. There is a simple, one-to-one correlation between the `veg TYPE` variable (`MVEGL(L))` and the MHK type described in the `MHK.inp` file. That is, 91 corresponds to the first line of MHK data (line 31, which is the first type of MHK device), 92 corresponds to the second MHK type on line 32, etc. It is also noted that for each MHK type, there is one line of data describing the geometry and characteristics of the device (see the descriptions in Table 2). However, there is only one set of data describing the K–ε parameters (`BETAMHK_P`, `BETAMHK_D`, and `CE4MHK`) and the partial blockage coefficient (`PBCOEF`) for all devices. This could easily be adjusted in future updates to the code.

MHK Source Code

INPUT.FOR

This is the subroutine responsible for reading the *EFDC.INP* as well as several others including *DXDY.INP*. Code was added to specifically read the *MHK.INP* file when `ISVEG=1` and when `MVEGL(L)` has a component larger than 90. Several error checks are included. FORTRAN programmers who are familiar with EFDC should have no difficulty understanding how inputs are read. There is a variable dictionary at the end of this document.

```
!!!BEGIN SCJ BLOCK
  IF(MAXVAL(MVEGL(2:LA))>90)THEN !MHK devices
    PRINT *,'READING MHK.INP'
    OPEN(1,FILE='MHK.INP',STATUS='UNKNOWN')
    DO NS=1,29
      READ(1,*) !skip 29 header lines
    ENDDO
    READ(1,*,ERR=3122)MHKTYP,NFLAGPWR,UPSTREAM,OUTPUTFLAG
    ALLOCATE(BOFFMHK(MHKTYP),BOFFSUP(MHKTYP))
    ALLOCATE(TOFFMHK(MHKTYP),TOFFSUP(MHKTYP))
    IF(NFLAGPWR==1)THEN
      DO M=1,MHKTYP  !read 1 line per MHK tupe
        READ(1,*,ERR=3122)WIDTHMHK(M),WIDTHSUP(M),
 &BOFFMHK(M),BOFFSUP(M),TOFFMHK(M),TOFFSUP(M),
 &CTMHK(M),CDSUP(M),VMINCUT(M),VMAXCUT(M),DENMHK(M)
    CTMHK(M)=CTMHK(M)*DENMHK(M)
    CDSUP(M)=CDSUP(M)*DENMHK(M)
        DO L=2,LA
          IF(M+90==MVEGL(L))THEN !set min/max elevations
            ZMINMHK(M,L)=BELV(L)+BOFFMHK(M)
            ZMAXMHK(M,L)=BELV(L)+TOFFMHK(M)
            ZMINSUP(M,L)=BELV(L)+BOFFSUP(M)
```

```
            ZMAXSUP(M,L)=BELV(L)+TOFFSUP(M)
            DIAMMHK=ZMAXMHK(M,L)-ZMINMHK(M,L)
            IF(DIAMMHK<0.0)THEN !error check
              PRINT*,'MHK ZMIN > ZMAX'
              STOP
            ENDIF
          ENDIF
        ENDDO
      ENDDO
      READ(1,*) !skip the header line
      READ(1,*)BETAMHK_D,BETAMHK_P,CE4MHK,PB_COEF
    ELSEIF(NFLAGPWR==2)THEN
      PRINT*,'Not available yet'
    ELSEIF(NFLAGPWR==3)THEN !FFP input style
      READ(1,*,ERR=3122)WIDTHMHK(M),WIDTHSUP(M),
&BOFFMHK(M),HEIGHTMHK(M),HEIGHTSUP(M),REFELEV(M),
&CTMHK(M),CDSUP(M),VMINCUT(M),VMAXCUT(M),DENMHK(M)
 CTMHK(M)=CTMHK(M)*DENMHK(M)
 CDSUP(M)=CDSUP(M)*DENMHK(M)
      DO L=2,LA
        IF(M+90==MVEGL(L))THEN
          ZMINMHK(M,L)=BELV(L)+REFELEV(M)
          ZMAXMHK(M,L)=BELV(L)+REFELEV(M)+HEIGHTMHK(M)
          ZMINSUP(M,L)=BELV(L)
          ZMAXSUP(M,L)=BELV(L)+REFELEV(M)+HEIGHTSUP(M)
          DIAMMHK=ZMAXMHK(M,L)-ZMINMHK(M,L)
          IF(DIAMMHK<0.0)THEN
            PRINT*,'MHK ZMIN > ZMAX'
            STOP
          ENDIF
        ENDIF
      ENDDO
      READ(1,*) !skip the header line
      READ(1,*)BETAMHK_D,BETAMHK_P,CE4MHK,PB_COEF
    ENDIF
    CLOSE(1)
    DO L=2,LA
      IF(MVEGL(L)>90)THEN
        IF((DIAMMHK>DXP(L).OR.DIAMMHK>DYP(L))
&.AND.DENMHK(MVEGL(L)-90)>1.0)THEN !error check
          PRINT*,'MHK DIAMETER EXCEEDS CELL SIZE'
          PRINT*,'AND DENSITY >= 1'
          STOP
        ENDIF
      ENDIF
    ENDDO
  ENDIF
!!!END SCJ BLOCK
```

MHKPWR.f90

MHKPWR.f90 is the primary routine that calculates the effects of MHK-device momentum (energy) removal from the local flow. After defining and initializing the necessary variables in the MHK algorithm, the first thing done in the subroutine is to determine the fraction of each model layer (in a cell where an MHK device is present) occupied by an MHK turbine. In the code shown below, the fraction of each layer occupied by the device is calculated for each cell where an MHK device is present. A device may be wholly contained within a layer or may span several layers by completely filling one or more layers with partial occupancy in end-member layers (the uppermost and lowermost cells that the MHK device occupies). Layer fraction sums and layer fractions minus one are also calculated for later use in the algorithm. The same technique is used to calculate the layer occupancy of the MHK support structure. Layer fraction information is used in the internal-mode solution in EFDC where no net force is applied to the water column, but equal and opposite forces are applied across model layers to yield shear flows. In EFDC, the bulk advection is calculated as a depth-averaged force applied to the water column (external model solution) while the inter-layer shear flows in a cell are calculated in the internal-model solution. An MHK device applies a net force to the water column in the cell in which it is present while it also produces shear flows in a multi-layer model (when the device is present is some layers and not others). The net force from the MHK device is applied to the depth-averaged water column, but the shear flow (internal-mode solution) applies no net force to the water column by applying equal and opposite forces to the layers in the shear-flow calculation. In layers where the device is present, a force is applied against the flow distributed appropriately to the layers where the device is present according to the fraction of the layer that is occupied by the device. In layers where the device is not present, the same net force is distributed against the flow in layers where the device is absent (in the direction of the flow). While no net force is applied to the water column in these calculations, these inter-layer forces serve to develop a shear flow where flow is resisted in layers where the device is present and rerouted to accelerate flows in layers above and below the device (these calculations only apply to shear flows in the vertical direction).

```
DO K=1,KC !MHK device layer filler - which layers does the device occupy and at what fraction
  ZTOP=HP(L)*Z(K)+BELV(L) !layer top elevation
  IF(ZTOP<ZMINMHK(M,L))CYCLE !layer is below device
    ZBOTTOM=HP(L)*Z(K-1)+BELV(L) !layer bottom elevation
    IF(ZBOTTOM>ZMAXMHK(M,L))CYCLE !layer is above device
    IF(ZTOP>=ZMAXMHK(M,L).AND.ZBOTTOM<=ZMINMHK(M,L))THEN !device is wholly contained in this layer (special case)
      LAYFRACM(K)=(ZMAXMHK(M,L)-ZMINMHK(M,L))/(HP(L)*DZC(K)) !calculate fraction of layer that is occupied
      EXIT
    ENDIF
    IF(ZMAXMHK(M,L)>=ZTOP.AND.ZMINMHK(M,L)<=ZBOTTOM)THEN !this layer is fully occupied by the device
      LAYFRACM(K)=1.0
      CYCLE
    ENDIF
    IF(ZBOTTOM<ZMINMHK(M,L).AND.ZMAXMHK(M,L)>=ZTOP)THEN !this layer is partially occupied by the device (bottom)
    LAYFRACM(K)=(ZTOP-ZMINMHK(M,L))/(HP(L)*DZC(K)) !calculate the fraction of layer that is occupied
      CYCLE
    ENDIF
    IF(ZTOP>=ZMAXMHK(M,L).AND.ZMINMHK(M,L)<ZBOTTOM)THEN !this layer is partially occupied by the device (top)
      LAYFRACM(K)=(ZMAXMHK(M,L)-ZBOTTOM)/(HP(L)*DZC(K)) !calculate the fraction of layer that is occupied
      CYCLE
    ENDIF
  ENDDO
  NEGLAYFRACM(:)=LAYFRACM(:)-1.0 !negative of the layer fraction occupied by the MHK turbine
  SUMLAYM=SUM(LAYFRACM(1:KC));SUMNEGLAYM=SUM(NEGLAYFRACM(1:KC)) !Sum of MHK layer fractions
```

Next, local flow speeds are calculated – both the cell-center velocity vectors in the I (x) and J (y) directions, `UVEC` and `VVEC`, as well as the layer flow speed at the cell center, `FLOWSPEED(K)`. If there is no MHK device or support in the layer under consideration, then the rest of the subroutine is skipped. Following that, there is some extensive logic determining the flow speed one cell upstream from the device. This identifies U_∞ for calculation of the power generated by the MHK turbine. If `UPSTREAM=0` in *MHK.inp*, then the flow speed U_∞ = `FLOWSPEED(K)` and its components are simply those within the cell containing the MHK device (not the upstream cell).

```
UVEC=0.5*(U(L,K)+U(LE,K))      !I,J cell center u-speed
VVEC=0.5*(V(L,K)+V(LN,K))      !I,J cell center v-speed
FLOWSPEED(K)=SQRT(UVEC*UVEC+VVEC*VVEC) !I,J cell center speed
IF((LAYFRACM(K)==0.0.AND.LAYFRACS(K)==0.0).OR.FLOWSPEED(K)<1.0E-03)CYCLE !no MHK or support or velocity in
this layer
IF(UPSTREAM==1)THEN !use the upstream flowspeed to assess power extraction
  UATVFACE= 0.25*(U(L,K)+U(LE,K)+U(LS,K)+U(LSE,K))  !u-velocity at south face (the v-face)
  VATUFACE= 0.25*(V(L,K)+V(LW,K)+V(LN,K)+V(LNW,K))  !v-velocity at west face  (the u-face)
  UATVFACEN=0.25*(U(L,K)+U(LE,K)+U(LN,K)+U(LNE,K))  !u-velocity at north face (the u-north-face)
  VATUFACEE=0.25*(V(L,K)+V(LE,K)+V(LN,K)+V(LNE,K))  !v-velocity at east face  (the v-east-face)
  FS_WF=U(L ,K);FS_EF=U(LE,K) !velocities on the west/east   faces (u velocities into the cell)
  FS_NF=V(LN,K);FS_SF=V(LS,K) !velocities on the north/south faces (v velocities into the cell)
  SPDN=SQRT(UATVFACEN*UATVFACEN+V(LN,K)*V(LN,K)) !speed at north face
  SPDS=SQRT(UATVFACE*UATVFACE+V(L,K)*V(L,K)) !speed at south face
  SPDE=SQRT(U(LE,K)*U(LE,K)+VATUFACEE*VATUFACEE) !speed at east face
  SPDW=SQRT(U(L,K)*U(L,K)+VATUFACE*VATUFACE) !speed at west face
  IF(FS_NF>-0.01)SPDN=0.0 !flow is OUT of north face
  IF(FS_SF<0.01)SPDS=0.0 !flow is OUT of south face
  IF(FS_WF<0.01)SPDW=0.0 !flow is OUT of west face
  IF(FS_EF>-0.01)SPDE=0.0 !flow is OUT of east face
  MAXSPD=MAX(SPDN,SPDS,SPDE,SPDW) !identify maximum speed
  VVELUP = MAX(SPDN,SPDS)
  IF(V(L,K)<0)VVELUP=-VVELUP
  UVELUP = MAX(SPDW,SPDE)
  IF(U(L,K)<0)UVELUP=-UVELUP
  IF(MAXSPD==SPDN)THEN !what face is it on?
    VELUP=SQRT((0.25*(U(LN,K)+U(LN+1,K)+U(LNC(LN),K)+U(LNC(LN)+1,K)))**2+V(LN,K)**2)
  ELSEIF(MAXSPD==SPDS)THEN !South
    VELUP=SQRT((0.25*(U(LS,K)+U(LS+1,K)+U(LSC(LS),K)+U(LSC(LS)+1,K)))**2+V(LS,K)**2)
  ELSEIF(MAXSPD==SPDE)THEN !East
    VELUP=SQRT(U(LE+1,K)**2+(0.25*(V(LE,K)+V(LE+1,K)+V(LN+1,K)+V(MIN(LC,LN+2),K)))**2)
  ELSE !West
    VELUP=SQRT(U(LW  ,K)**2+(0.25*(V(LW,K)+V(LW-1,K)+V(LN-1,K)+V(LN-2,K)))**2)
  ENDIF
```

Now that the flow speeds (local cell-center velocity vectors and flow speed and also upstream flow speed as needed) are calculated, the force applied to the water column by the MHK turbine (momentum removed from the system and converted to power) is estimated according to:

$$F_{\mathrm{MHK}} = \frac{1}{2} C_{\mathrm{T}} \rho U_{\infty}^{2} A_{\mathrm{MHK}}, \tag{1}$$

where C_{T} (−) is the thrust coefficient (`CTMHK`), ρ (kg/m^3) is the fluid density (it does not show up in the code below because the force is normalized by the density for the flow calculations), U_{∞} (m/s) is the incoming flow speed (`VELUP`), and A_{MHK} (m^2) is the flow-facing area of the MHK turbine (`HP(L)*DZC(K)*WIDTHMHK(M)`). Note that the preceding equation is applied to each layer by multiplying by the fraction of the layer occupied by the MHK turbine (`LAYFRACM(K)`). The power generated by the MHK turbine, P_{MHK}, is simply the force, F_{MHK}, multiplied by the local velocity, U (`FLOWSPEED(K)`):

$$P_{\mathrm{MHK}} = F_{\mathrm{MHK}} U. \tag{2}$$

The forces applied to the flow by the MHK turbine are then applied to each face of the model cell where the device is present according to a flow-face area weighting. Finally, power is dimensionalized by multiplying by the (potentially variable) fluid density. The same algorithm described above is applied to the support structure with the only differences being that C_T is replaced by the drag coefficient C_D and that U_∞ in (1) is always replaced by the local flow speed, U. Ultimately, the MHK device results in a volumetric momentum extraction rate due to energy removal (as well as due to form and viscous drag from the MHK structure), S_Q (m^4/s^2), of [*James et al.*, 2010b]

$$S_Q = \frac{1}{2} C_T U_\infty^2 A_{MHK}. \tag{3}$$

```
!FMHK=0.5*ThrustCoef*Area*(U_inf)^2 where U_inf is the upstream velocity, VELUP [m^4/s^2]
FMHK=0.5*LAYFRACM(K)*THRSTCOEF*VELUP*VELUP*HP(L)*DZC(K)*WIDTHMHK(M) !area is ASSUMED square
!PMHK=FMHK*U where U is the local flowspeed
PMHK(L,K)=FMHK*FLOWSPEED(K) !ThrustCoef*|u|u^2*area [m^5/s^3] (will yield different power outputs depending on
UPSTREAM)
AWEIGHTXW=DYU(L)*HU(L)/(DYU(L)*HU(L)+DYU(LE)*HU(LE));AWEIGHTXE=1.0-AWEIGHTXW !area-weight for west/east faces
AWEIGHTYS=DXV(L)*HV(L)/(DXV(L)*HV(L)+DXV(LN)*HV(LN));AWEIGHTYN=1.0-AWEIGHTYS !area-weight for south/north faces
!To get the x and y components, multiply by a velocity vector divided by the local flow speed
FXMHK=FMHK*UVEC/FLOWSPEED(K)
FXMHK(L ,K)=FXMHK(L ,K)+AWEIGHTXW*SUB(L )*FMHK*UVEC/FLOWSPEED(K) !SUB(L)*FMHK(L,K)*(Uvel/q) [m^4/s^2]
FXMHK(LE,K)=FXMHK(LE,K)+AWEIGHTXE*SUB(LE)*FMHK*UVEC/FLOWSPEED(K) !distribute forces on each U-face of the cell
FYMHK(L ,K)=FYMHK(L ,K)+AWEIGHTYS*SVB(L )*FMHK*VVEC/FLOWSPEED(K) !y components of "forces" [m^4/s^2]
FYMHK(LN,K)=FYMHK(LN,K)+AWEIGHTYN*SVB(LN)*FMHK*VVEC/FLOWSPEED(K) !distribute forces on each V-face of the cell
IF(BSC>0.0)THEN !if variable density, take it into account
  PMHK(L,K)=PMHK(L,K)*(B(L,K)+1.0)*1000.0
ELSE
  PMHK(L,K)=PMHK(L,K)*1024. !density of seawater is ~1024kg/m^3
ENDIF
```

Next, the internal-mode calculations (forces that change flow speeds in individual layers without affecting the bulk or column flow velocities) are made for all model layers. These are the forces applied to each layer of the water-column cell containing an MHK device. It is noted that the net force applied in the internal mode is zero. For every unit of force pushing against the flow (in layers where the MHK turbine or support structure are present), there is an equal and opposite force applied in the direction of flow (in layers where the MHK turbine or device are not present). Obviously, for a single-layer model these internal-mode forces are zero. The partial blockage coefficient comes into play here by essentially amplifying the force differentials across layers for the internal-mode solution (the sum of internal-mode forces is still zero). This is justified by the physical displacement of fluid by the device (solidity) and is required to ensure proper wake characteristics.

```
DO K=1,KC
  IF(SUMLAYS==0.0)THEN !No total force can be added to the internal-mode solution the way this is written,
the sum across layers is zero. Internal forces are directional, so the sum of FXMHK,FYMHK,FXSUP,FYSUP are
used
    FX(L ,K)=FX(L ,K)+PB_COEF*SUM(FXMHK(L ,1:KC))*(LAYFRACM(K)/SUMLAYM-NEGLAYFRACM(K)/SUMNEGLAYM) !pull x-
force out of MHK layer for internal mode (no support structure) - push forces in other layers
    FX(LE,K)=FX(LE,K)+PB_COEF*SUM(FXMHK(LE,1:KC))*(LAYFRACM(K)/SUMLAYM-NEGLAYFRACM(K)/SUMNEGLAYM) !pull x-
force out of MHK layer for internal mode (east face) - push forces in other layers
    FY(L ,K)=FY(L ,K)+PB_COEF*SUM(FYMHK(L ,1:KC))*(LAYFRACM(K)/SUMLAYM-NEGLAYFRACM(K)/SUMNEGLAYM) !pull y-
force out of MHK layer for internal mode (no support structure) - push forces in other layers
    FY(LN,K)=FY(LN,K)+PB_COEF*SUM(FYMHK(LN,1:KC))*(LAYFRACM(K)/SUMLAYM-NEGLAYFRACM(K)/SUMNEGLAYM) !pull y-
force out of MHK layer for internal mode (north face) - push forces in other layers
```

```
  ELSE
    FX(L ,K)=FX(L ,K)+(PB_COEF*SUM(FXMHK(L ,1:KC))*(LAYFRACM(K)/SUMLAYM-
NEGLAYFRACM(K)/SUMNEGLAYM)+SUM(FXSUP(L ,1:KC))*(LAYFRACS(K)/SUMLAYS-NEGLAYFRACS(K)/SUMNEGLAYS)) !pull x-
force out of MHK/support layer for internal mode - push forces in other layers
    FX(LE,K)=FX(LE,K)+(PB_COEF*SUM(FXMHK(LE,1:KC))*(LAYFRACM(K)/SUMLAYM-
NEGLAYFRACM(K)/SUMNEGLAYM)+SUM(FXSUP(LE,1:KC))*(LAYFRACS(K)/SUMLAYS-NEGLAYFRACS(K)/SUMNEGLAYS)) !pull x-
force out of MHK/support layer for internal mode - push forces in other layers
    FY(L ,K)=FY(L ,K)+(PB_COEF*SUM(FYMHK(L ,1:KC))*(LAYFRACM(K)/SUMLAYM-
NEGLAYFRACM(K)/SUMNEGLAYM)+SUM(FYSUP(L ,1:KC))*(LAYFRACS(K)/SUMLAYS-NEGLAYFRACS(K)/SUMNEGLAYS)) !pull x-
force out of MHK/support layer for internal mode - push forces in other layers
    FY(LN,K)=FY(LN,K)+(PB_COEF*SUM(FYMHK(LN,1:KC))*(LAYFRACM(K)/SUMLAYM-
NEGLAYFRACM(K)/SUMNEGLAYM)+SUM(FYSUP(LN,1:KC))*(LAYFRACS(K)/SUMLAYS-NEGLAYFRACS(K)/SUMNEGLAYS)) !pull x-
force out of MHK/support layer for internal mode - push forces in other layers
  ENDIF
ENDDO
```

Next, the external-mode forces are calculated by summing the absolute value of the forces applied against the water column by the MHK turbine and support structure. Absolute values are used because the forces are later "directionalized" by multiplying by the local flow velocity vectors in *CALPUV.FOR*. Also, to ensure proper units supplied to *CALEXP2T.FOR* and then to *CALPUV.FOR* the external-mode forces must be multiplied by an absolute value of the average directional speed normalized by the average speed (not directional). This is again divided by the average speed to ensure units of m^3/s. Finally, the energy generated by the turbine (EMHK) is calculated in Megawatt-hours using (2) multiplied by the model time step. The energy dissipated by the support structure (ESUP) is calculated similarly.

```
FXMHKE(L)=FXMHKE(L)+SUM(ABS(FXMHK(L,1:KC)));FXMHKE(LE)=FXMHKE(LE)+SUM(ABS(FXMHK(LE,1:KC))) !Sum layer force
magnitudes for external mode solution (need absolute value of forces)
FYMHKE(L)=FYMHKE(L)+SUM(ABS(FYMHK(L,1:KC)));FYMHKE(LN)=FYMHKE(LN)+SUM(ABS(FYMHK(LN,1:KC))) !Sum layer force
magnitudes for external mode solution (these forces are later multiplied by a directional velocity so they need to
be absolute values here)
FXSUPE(L)=FXSUPE(L)+SUM(ABS(FXSUP(L,1:KC)));FXSUPE(LE)=FXSUPE(LE)+SUM(ABS(FXSUP(LE,1:KC))) !Sum layer force
magnitudes for external mode solution (absolute values because these are later multiplied by the local velocity to
apply a direction)
FYSUPE(L)=FYSUPE(L)+SUM(ABS(FYSUP(L,1:KC)));FYSUPE(LN)=FYSUPE(LN)+SUM(ABS(FYSUP(LN,1:KC))) !Sum layer force
magnitudes for external mode solution
AVGSPD=SUM(FLOWSPEED(1:KC)*DZC(1:KC))
IF(AVGSPD==0.0)CYCLE
!CALEXP2T is expecting units of [m^3/s] for FXMHKE, which is the sum of absolute values of FXMHK
!CALEXP2T divides by water-column volume before passing this "force" onto FUHDYE (in units of [1/s]), which is
used for momentum conservation in CALPUV
!Units of FXMHKE (etc) are same as FX and FXMHK (etc) [m^4/s^2] so they must be divided by the average speed in
this water column
!Because these forces should be close to zero when the U or V speeds are zero, we need to include a normalized
form as either U/AVGSPD or V/AVGSPD. Without this "directional normalization," when AVGSPD is small, external mode
forces are too large.
!Multiply by a directional velocity normalized by AVGSPD and then divide by AVGSPD to get the units correct
FXMHKE(L )=FXMHKE(L )*ABS(SUM(U(L ,1:KC)*DZC(1:KC)))/AVGSPD/AVGSPD !external mode solution units of [m^3/s]
FXSUPE(L )=FXSUPE(L )*ABS(SUM(U(L ,1:KC)*DZC(1:KC)))/AVGSPD/AVGSPD !external mode solution units of [m^3/s]
FXMHKE(LE)=FXMHKE(LE)*ABS(SUM(U(LE,1:KC)*DZC(1:KC)))/AVGSPD/AVGSPD !external mode solution units of [m^3/s]
FXSUPE(LE)=FXSUPE(LE)*ABS(SUM(U(LE,1:KC)*DZC(1:KC)))/AVGSPD/AVGSPD !external mode solution units of [m^3/s]
FYMHKE(L )=FYMHKE(L )*ABS(SUM(V(L ,1:KC)*DZC(1:KC)))/AVGSPD/AVGSPD !external mode solution units of [m^3/s]
FYSUPE(L )=FYSUPE(L )*ABS(SUM(V(L ,1:KC)*DZC(1:KC)))/AVGSPD/AVGSPD !external mode solution units of [m^3/s]
FYMHKE(LN)=FYMHKE(LN)*ABS(SUM(V(LN,1:KC)*DZC(1:KC)))/AVGSPD/AVGSPD !external mode solution units of [m^3/s]
FYSUPE(LN)=FYSUPE(LN)*ABS(SUM(V(LN,1:KC)*DZC(1:KC)))/AVGSPD/AVGSPD !external mode solution units of [m^3/s]
EMHK(MHKCOUNT,L)=EMHK(MHKCOUNT,L)+DT*SUM(PMHK(L,1:KC))*2.7778E-10 !factor converts to MW-hr
ESUP(MHKCOUNT,L)=ESUP(MHKCOUNT,L)+DT*SUM(PSUP(L,1:KC))*2.7778E-10 !factor converts to MW-hr
```

CALEXP2T.FOR

Subroutine *CALEXP2T.FOR* accumulates momentum-equation terms using the two-time-level scheme (i.e., changes to the momentum due to various external forces including bottom shear stresses, Coriolis forces, boundary inflows, vegetative drag, MHK forces, nonhydrostatic pressures, and wave Reynolds stresses). In the portion of the routine responsible for adding vegetative resistance to the momentum equation, modifications were made to add MHK-device resistance. The MHK-device momentum terms (`FXMHKE`, `FYMHKE`, `FXSUPE`, and `FYSUPE`) are added to the vegetative resistance terms (`FXVEGE` and `FYVEGE`). Internal-mode forces, `FX` and `FY`, were updated directly in *MHKPWR.f90*. Note that the original EFDC code present in this block was re-written to make it more understandable. The section of code was also thoroughly commented.

```
!!!Begin SCJ block
IF(ISVEG>=1)THEN
  FXVEGE(:)=0.0;FYVEGE(:)=0.0
  DO L=2,LA  !loop over the model area
    IF(.NOT.LMASKDRY(L).OR.MVEGL(L)==MVEGOW)CYCLE  !if the cell is dry, or if it is open water, or if 
there is no vegetation in cell L, skip this cell
    IF(MVEGL(L)==0.AND.MVEGL(L-1)==0.AND.MVEGL(LSC(L))==0)CYCLE !if not this cell and no surrounding cells 
are vegetation, skip
    DO K=1,KC  !loop over the model layers
      LW=L-1       !west cell
      LE=L+1       !east cell
      LS=LSC(L)    !south cell
      LN=LNC(L)    !north cell
      LNW=LNWC(L)  !northwest cell
      LSE=LSEC(L)  !southeast cell
      UTMPATV=0.25*(U(L,K)+U(LE,K)+U(LS,K)+U(LSE,K))  !u-velocity at v face
      VTMPATU=0.25*(V(L,K)+V(LW,K)+V(LN,K)+V(LNW,K))  !v-velocity at u face
      UMAGTMP=SQRT( U(L,K)*U(L,K)+VTMPATU*VTMPATU )   !u-face velocity vector magnitude
      VMAGTMP=SQRT( UTMPATV*UTMPATV+V(L,K)*V(L,K) )   !v-face velocity vector magnitude
!FXVEG/FYVEG come from CALTBXY unitless, but they are really just a form of the drag coefficient with 
terms accounting for the area density
!FXVEG/FYVEG only change inasmuch as the water depth changes and are zero in layers not penetrated by 
vegetation
!FXVEG/FYVEG are C_d(N/L^2)
!FXVEG/FYVEG are now multiplied by the cell area and cell-averaged velocity
!FXVEG/FYVEG are C_d(N/L^2)A|q|
      FXVEG(L,K)=UMAGTMP*SUB(L)*FXVEG(L,K)  ![m/s] |q_x|C_d
      FYVEG(L,K)=VMAGTMP*SVB(L)*FYVEG(L,K)  ![m/s] |q_y|C_d
    ENDDO
!FXVEG/FXVEGE are multiplied by the local velocity and face-centered area to yield units of [m^4/s^2]
!FXVEG/FXVEGE are added to the body forces as C_d(N/L^2)A|q|q
    FXVEGE(L)=SUM(FXVEG(L,:)*DZC(:))  !Columns of vegetative resistance [m/s] used in FUHDXE (this is the 
average force on the water column)
    FYVEGE(L)=SUM(FYVEG(L,:)*DZC(:))  !Columns of vegetative resistance [m/s] used in FVHDYE (this is the 
average force on the water column)
    FX(L,:)=FX(L,:)+(FXVEG(L,:)-FXVEGE(L))*U(L,:)*DXYU(L) ![m^4/s^2] adding vegetative resistance to the 
body force (no net force added) FXVEGE goes into FUHDXE for momentum conservation
    FY(L,:)=FY(L,:)+(FYVEG(L,:)-FYVEGE(L))*V(L,:)*DXYV(L) ![m^4/s^2] adding vegetative resistance to the 
body force (no net force added) FYVEGE goes into FVHDYE for momentum conservation
  ENDDO
  IF(MAXVAL(MVEGL(2:LA))>90)CALL MHKPWRDIS !MHK devices exist
  FXVEGE(:)=FXVEGE(:)*HUI(:)*FLOAT(KC) !Calculate vegetative dissipation for FUHDYE for momentum 
conservation in CALPUV (need to have sum of forces, not average provided to CALPUV)
```

```
  FYVEGE(:)=FYVEGE(:)*HVI(:)*FLOAT(KC) !Calculate vegetative dissipation for FVHDXE for momentum conservation in CALPUV (need to have sum of forces, not average provided to CALPUV)
  FXVEGE(:)=FXVEGE(:)+FXMHKE(:)*HUI(:)*DXYIU(:) !Add MHK to vegetative dissipation in FUHDYE for momentum conservation in CALPUV (divide by volume)
  FYVEGE(:)=FYVEGE(:)+FYMHKE(:)*HVI(:)*DXYIV(:) !Add MHK to vegetative dissipation in FVHDXE for momentum conservation in CALPUV (divide by volume)
  FXVEGE(:)=FXVEGE(:)+FXSUPE(:)*HUI(:)*DXYIU(:) !Add MHK support to vegetative dissipation in FUHDYE for momentum conservation in CALPUV (divide by volume)
  FYVEGE(:)=FYVEGE(:)+FYSUPE(:)*HVI(:)*DXYIV(:) !Add MHK support to vegetative dissipation in FVHDXE for momentum conservation in CALPUV (divide by volume)
ENDIF
!!!End SCJ block
```

CALQQ2T.FOR

Both vegetation and MHK devices have an impact on K–ε turbulence transport (turbulence intensity and its dissipation rate). The implementation of the mathematics of changes to K–ε are thoroughly outlined by *James et al.* [2010b]. MHK devices remove momentum from a system according to S_Q (3), but also alter the turbulent kinetic energy, K, and turbulent kinetic energy dissipation rate, ε. These effects are captured with appropriate sink terms. S_K (m^5/s^3) represents the volumetric change in net turbulent kinetic energy in the appropriate model cell due to the MHK device (support), with $S\varepsilon$ (m^5/s^3) as its analogous term for the volumetric kinetic energy dissipation rate equation [*Poggi et al.*, 2004]. These quantities are advected and dispersed downstream of the MHK device according to the standard conservation equations used in EFDC [*Hamrick*, 2007b]. The term S_K arises because MHK devices break up the mean flow motion and generate wake turbulence ($\approx \tfrac{1}{2}C_{\mathrm{T}}A_{\mathrm{MHK}}U^3$) [*Kaimal and Finnigan*, 1994; *Raupach and Shaw*, 1982]. However, such wakes dissipate fairly rapidly, speculatively within about 20 MHK device lengths (turbine diameters). MHK flow models may show overly persistent wakes if the K–ε terms are not taken into account. The canonical (or physics-based) form for S_K reflecting the effects of a momentum sink (or partial flow obstruction) is [*Sanz*, 2003]

$$S_K = \frac{1}{2} C_{\mathrm{T}} A_{\mathrm{MHK}} \left(\beta_p U^3 - \beta_d U K \right), \tag{4}$$

where dimensionless β_p ($\approx$ 0−1) is the fraction of mean flow kinetic energy converted to wake-generated K (m^2/s^2) by drag (i.e., a source term in the K budget), and term proportional to dimensionless β_d ($\approx$ 1.0–5.0) accounts for the short circuiting of turbulence cascade (the transfer of energy between the large-scale turbulence to smaller scales of turbulence, which acts as a sink term in the K budget).

The most obvious weakness of the K–ε approach is its least understood term, S_ε [*Wilson et al.*, 1998]. Over the last decade or so, various formulae have been proposed for S_ε [*Green*, 1992; *Liu et al.*, 1996], but the simplest is used in this model:

$$S_\varepsilon = \frac{1}{2} C_{\varepsilon 4} \frac{\varepsilon}{K} S_K, \tag{5}$$

where $C_{\varepsilon 4}$ is a closure constant that enhances turbulence dissipation in proportion to turbulence generation [*Katul et al.*, 2004]. Adding this term around the MHK turbine is basically compensating the production of turbulence created by the axial velocity shear by a proportional increase in the dissipation. It is justified physically by arguing that it represents the "energy transfer rate from large-scale turbulence to small-scale turbulence controlled by the production range scale and the dissipation rate time scale [*El Kasmi and Masson*, 2008]." The formulation for (5) is based on standard dimensional analysis common to all K–ε approaches. Upon adding (3)−(5) to the momentum and K–ε equations, it is possible to solve for momentum, K, and ε after appropriate upper and lower boundary conditions are specified that account for shears at the sediment bed and water surface. For this implementation, $C_{\varepsilon 4} = 0.9$, $\beta_p = 1.0$ and $\beta_d = 5.1$ for the MHK support structure, but are inputs for the MHK turbine from *MHK.inp*. In SNL-EFDC, momentum is defined as the product of flow depth, H (m), and velocity (u and v); conservation of kinetic energy is solved in terms of $\frac{1}{2}Hq^2$, where q (m/s) is the turbulent intensity, and conservation of turbulent energy dissipation rate takes the form Hq^2l, where l (m) is the turbulence length scale. A detailed explanation of the implementation of the K–ε equations in SNL-EFDC is provided by *James et al.* [2010b].

Following the same technique in EFDC where turbulence intensity and its dissipation rate (length scale) due to vegetative resistance are calculated, the equivalent terms are calculated for the MHK device and its support structure. Both internal- and external-mode components are calculated (`PQQMHKE`, `PQQMHKI`, `PQQSUPE`, and `PQQSUPI`). Then, for each subsequent appearance of `PQQVEGE` and `PQQVEGI` in *CALQQ2T.FOR*, the terms `PQQMHKE` + `PQQSUPE` and `PQQMHKI` + `PQQSUPI` are appropriately added, respectively.

```
IF(ISVEG.GT.0)THEN !SCJ vegetative/MHK impact on K-epsilon
  DO K=1,KS
    DO L=2,LA
      LE=L+1
      LN=LNC(L)
      TMPQQI=0.25*BETAVEG_P
      TMPQQE=0.25*BETAVEG_D
      PQQVEGI(L,K)=TMPQQI*(FXVEG(L ,K  )+FXVEG(L ,K+1)
&                         +FXVEG(LE,K  )+FXVEG(LE,K+1)
&                         +FYVEG(L ,K  )+FYVEG(L ,K+1)
&                         +FYVEG(LN,K  )+FYVEG(LN,K+1))
      PQQVEGE(L,K)=TMPQQE*(FXVEG(L ,K  )*U(L ,K  )*U(L ,K  )
&                         +FXVEG(L ,K+1)*U(L ,K+1)*U(L ,K+1)
&                         +FXVEG(LE,K  )*U(LE,K  )*U(LE,K  )
&                         +FXVEG(LE,K+1)*U(LE,K+1)*U(LE,K+1)
&                         +FYVEG(L ,K  )*V(L ,K  )*V(L ,K  )
&                         +FYVEG(L ,K+1)*V(L ,K+1)*V(L ,K+1)
&                         +FYVEG(LN,K  )*V(LN,K  )*V(LN,K  )
&                         +FYVEG(LN,K+1)*V(LN,K+1)*V(LN,K+1))
      IF(MVEGL(L)>90)THEN
        TMPQQI=0.25*BETAMHK_P
        TMPQQE=0.25*BETAMHK_D
        PQQMHKI(L,K)=TMPQQI*(FXMHK(L ,K  )+FXMHK(L ,K+1)
&                           +FXMHK(LE,K  )+FXMHK(LE,K+1)
&                           +FYMHK(L ,K  )+FYMHK(L ,K+1)
&                           +FYMHK(LN,K  )+FYMHK(LN,K+1))
        PQQMHKE(L,K)=TMPQQE*(FXMHK(L ,K  )*U(L ,K  )*U(L ,K  )
&                           +FXMHK(L ,K+1)*U(L ,K+1)*U(L ,K+1)
&                           +FXMHK(LE,K  )*U(LE,K  )*U(LE,K  )
&                           +FXMHK(LE,K+1)*U(LE,K+1)*U(LE,K+1)
&                           +FYMHK(L ,K  )*V(L ,K  )*V(L ,K  )
&                           +FYMHK(L ,K+1)*V(L ,K+1)*V(L ,K+1)
&                           +FYMHK(LN,K  )*V(LN,K  )*V(LN,K  )
&                           +FYMHK(LN,K+1)*V(LN,K+1)*V(LN,K+1))
        TMPQQI=0.25*BETASUP_P
        TMPQQE=0.25*BETASUP_D
        PQQSUPI(L,K)=TMPQQI*(FXSUP(L ,K  )+FXSUP(L ,K+1)
&                           +FXSUP(LE,K  )+FXSUP(LE,K+1)
&                           +FYSUP(L ,K  )+FYSUP(L ,K+1)
&                           +FYSUP(LN,K  )+FYSUP(LN,K+1))
        PQQSUPE(L,K)=TMPQQE*(FXSUP(L ,K  )*U(L ,K  )*U(L ,K  )
&                           +FXSUP(L ,K+1)*U(L ,K+1)*U(L ,K+1)
&                           +FXSUP(LE,K  )*U(LE,K  )*U(LE,K  )
&                           +FXSUP(LE,K+1)*U(LE,K+1)*U(LE,K+1)
&                           +FYSUP(L ,K  )*V(L ,K  )*V(L ,K  )
&                           +FYSUP(L ,K+1)*V(L ,K+1)*V(L ,K+1)
&                           +FYSUP(LN,K  )*V(LN,K  )*V(LN,K  )
&                           +FYSUP(LN,K+1)*V(LN,K+1)*V(LN,K+1))
      ENDIF
    ENDDO
  ENDDO
ENDIF
```

Variable Dictionary

The table below lists the SNL-EFDC variables in alphabetical order with a brief description. Variables from EFDC are denoted as such in the description.

Variable name	Description
AVGSPD	Average flow speed in a column of water (all layers) (m/s)
AWEIGHTXE	Area weight for the east face of a cell layer with an MHK device present
AWEIGHTXW	Area weight for the west face of a cell layer with an MHK device present
AWEIGHTYN	Area weight for the north face of a cell layer with an MHK device present
AWEIGHTYS	Area weight for the south face of a cell layer with an MHK device present
BETAMHK_D	K–ε coefficient for the MHK turbine type (turbulent kinetic energy equation)
BETAMHK_P	K–ε coefficient for the MHK turbine type (turbulent kinetic energy equation)
BETASUP_D	K–ε coefficient = 5.1 for the MHK support structure (turbulent kinetic energy equation)
BETASUP_P	K–ε coefficient = 1.0 for the MHK support structure (turbulent kinetic energy equation)
BOFFMHK(M)	Distance from the sediment bed to the bottom of an MHK turbine type (m)
BOFFSUP(M)	Distance from the sediment bed to the bottom of the MHK support structure (m)
CDSUP(M)	Drag coefficient for the MHK support structure
CE4MHK	K–ε coefficient for the MHK turbine type (turbulent kinetic energy dissipation rate/length scale equation)
CE4SUP	K–ε coefficient = 0.9 for the MHK support structure (turbulent kinetic energy dissipation rate/length scale equation)
CTMHK(M)	Thrust coefficient of the MHK turbine
DENMHK(M)	Density (number or fraction) of MHK devices per cell
EMHK(M,L)	Cumulative flow energy removed from the system by the MHK turbine type at a cell (MW-hr)
ESUP(M,L)	Cumulative energy dissipated by the MHK support structure type at a cell (MW-hr)
FLOWSPEED(K)	Flow speed in a model layer where an MHK device is present (m/s)
FMHK	Total "force" from the MHK turbine against the flow in a model layer (m^4/s^2)
FS_EF	Outward flow speed at the east face of a cell (m/s)
FS_NF	Outward flow speed at the north face of a cell (m/s)
FS_SF	Outward flow speed at the south face of a cell (m/s)
FS_WF	Outward flow speed at the west face of a cell (m/s)
FSUP	Total "force" from the MHK support structure against the flow in a model layer (m^4/s^2)
FXMHK(L,K)	I (x) "force" exerted against the flow in a layer where an MHK turbine is present (m^4/s^2)
FXMHKE(L)	Absolute value of the total I (x) "force" exerted against the entire water column in a cell by an MHK turbine used in the external model solution: `SUM(ABS(FXMHK(L,1:KC)))` (m^4/s^2)
FXSUP(L,K)	I (x) "force" exerted against the flow in a layer where the MHK support structure is present (m^4/s^2)
FXSUPE(L)	Absolute value of the total I (x) "force" exerted against the entire water column in a cell by

	the MHK support structure used in the external model solution: SUM(ABS(FXSUP(L,1:KC))) (m^4/s^2)
FYMHK(L,K)	*J* (*y*) "force" exerted against the flow in a layer where an MHK turbine is present (m^4/s^2)
FYMHKE(L)	Absolute value of the total *J* (*y*) "force" exerted against the entire water column in a cell by an MHK turbine used in the external model solution: SUM(ABS(FYMHK(L,1:KC))) (m^4/s^2)
FYSUP(L,K)	*J* (*y*) "force" exerted against the flow in a layer where the MHK support structure is present (m^4/s^2)
FYSUPE(L)	Absolute value of the total J (y) "force" exerted against the entire water column in a cell by the MHK support structure used in the external model solution: SUM(ABS(FYSUP(L,1:KC))) (m^4/s^2)
K	Loop control variable for model layers
L	Model cell index from EFDC
LAYFRACM(K)	Fraction of a model layer occupied by the MHK turbine
LAYFRACS(K)	Fraction of a model layer occupied by the MHK support structure
LE	L of the cell to the east of the current cell LE=L+1
LN	L of the cell to the north of the current cell LN=LNC(L)
LNE	L of the cell to the northeast of the current cell LNE=LNEC(L)
LNW	L of the cell to the northwest of the current cell LNW=LNWC(L)
LS	L of the cell to the south of the current cell LS=LSC(L)
LSE	L of the cell to the southeast of the current cell LSE=LSEC(L)
LW	L of the cell to the west of the current cell LW=L-1
M	This identifies the MHK type MVEGL(L)-90
MAXSPD	Maximum of the speeds on each face of a cell (m/s)
MHKCOUNT	Running count of cells with MHK devices present; this controls the counter for the energy variables
MHKTYPE	The integer number of MHK types, one line of turbine type data is needed in the *MHK.INP* file for each MHK type
NEGLAYFRACM(K)	LAYFRACM(K)-1
NEGLAYFRACS(K)	LAYFRACS(K)-1
NFLAGPWR	Flag specifying how power is generated by the MHK turbine (calculated from the standard equation or a look-up table)
OUTPUTFLAG	Flag specifying the type of output printed (e.g., for calibration efforts or just general output data)
PB_COEF	Partial blockage coefficient to account for the physical displacement of fluid by the MHK device
PMHK(L,K)	Power removed from the flow from a layer of the model where an MHK turbine is present (Watts)
PQQMHKE(L,K)	K–ε component supplied to the external mode solution due to the MHK turbine (e.g., BETAMHK_D*FXMHK(L,K)*U(L,K)*U(L,K))
PQQMHKI(L,K)	K–ε component supplied to the internal mode solution due to the MHK turbine (e.g., BETAMHK_P*FXMHK(L,K))
PQQSUPE(L,K)	K–ε component supplied to the external mode solution due to the MHK support structure (e.g., BETASUP_D*FXSUP(L,K)*U(L,K)*U(L,K))
PQQSUPI(L,K)	K–ε component supplied to the internal mode solution due to the MHK support structure (e.g., BETASUP_P*FXSUP(L,K))

PSUP(L,K)	Power removed from the flow from a layer of the model when the MHK support structure is present (Watts)
SPDE	Flow speed at the east face of a cell (m/s)
SPDN	Flow speed at the north face of a cell (m/s)
SPDS	Flow speed at the south face of a cell (m/s)
SPDW	Flow speed at the west face of a cell (m/s)
SUMLAYM	Sum of the layer fractions with an MHK turbine present SUM(LAYFRACM(1:KC))
SUMLAYS	Sum of the layer fractions with MHK support structure present SUM(LAYFRACS(1:KC))
SUMNEGLAYM	SUM(NEGLAYFRACM(1:KC))
SUMNEGLAYS	SUM(NEGLAYFRACS(1:KC))
THRSTCOEF	MHK turbine thrust coefficient equal to the thrust coefficient for the device CTMHK(M) multiplied by the device density DENMHK(M)
TOFFMHK(M)	Distance from the sediment bed to the top of the MHK turbine type (m)
TOFFSUP(M)	Distance from the sediment bed to the top of the MHK support structure (m)
UATVFACE	Velocity in the *I* (*x*) direction at the south face of a cell (m/s)
UATVFACEN	Velocity in the *I* (*x*) direction at the north face of a cell (m/s)
UPSTREAM	Flag specifying whether the upstream or local cell velocity is used in the MHK power calculations
UVEC	Velocity in the *I* (*x*) direction where the MHK device is present (m/s)
UVELUP	Upstream velocity in the *I* (*x*) direction (m/s)
VATUFACE	Velocity in the *J* (*y*) direction at the west face of a cell (m/s)
VATUFACEE	Velocity in the *J* (*y*) direction at the east face of a cell (m/s)
VELUP	Upstream velocity (either one cell upstream or at the MHK cell depending on the user's specification) (m/s)
VMAXCUT(M)	Velocity beyond which the MHK turbine type generates no additional power (m/s)
VMINCUT(M)	Cut-in (minimum velocity) before the MHK turbine type starts generating power (m/s)
VVEC	Velocity in the *J* (*y*) direction where the MHK device is present (m/s)
VVELUP	Upstream velocity in the *J* (*y*) direction (m/s)
WIDTHMHK(M)	Width of the MHK turbine type (m)
WIDTHSUP(M)	Width of the MHK support structure type (m)
ZBOTTOM	Bottom elevation of a model layer (m)
ZMAXMHK(M,L)	Maximum elevation of the MHK turbine type in a cell - used to calculate LAYFRACM (m)
ZMAXSUP(M,L)	Maximum elevation of the MHK support structure in a cell - used to calculate LAYFRACS (m)
ZMINMHK(M,L)	Minimum elevation of the MHK turbine type in a cell - used to calculate LAYFRACM (m)
ZMINSUP(M,L)	Minimum elevation of the MHK support structure in a cell - used to calculate LAYFRACS (m)
ZTOP	Top elevation of a model layer (m)

Special Notes

Currently, the MHK routine is not set up to handle user-defined power curves for MHK devices. Should the need arise, this can be easily incorporated.

The model has a provision to specify the turbine location in the vertical as a minimum distance from the top of the water column.

There is a single set of K–ε parameters and partial blockage coefficients for all MHK devices. This is because these are generally fairly insensitive parameters to the flow solution (however, the partial blockage coefficient can be important, especially as it increases significantly past unity). In the future, these parameters could be imported for each MHK device type.

References

El Kasmi, A., and C. Masson (2008), An extended k-ε model for turbulent flow through horizontal-axis wind turbines, *Journal of Wind Engineering and Industrial Aerodyamics*, *96*(1), 103-122.

Green, S. (1992), Modelling turbulent air flow in a stand of widely spaced trees, *Journal of Computational Fluid Dynamics Applications*, *5*, 294-312.

Hamrick, J. M. (1992), A Three-Dimensional Environmental Fluid Dynamics Computer Code: Theoretical and Computational Aspects, *Special Report 317*, 63 pp, The College of William and Mary.

Hamrick, J. M. (2007a), The Environmental Fluid Dynamics Code: User Manual, *Version 1.01*, US EPA, Fairfax, VA.

Hamrick, J. M. (2007b), The Environmental Fluid Dynamics Code: Theory and Computation, *Version 1.01*, US EPA, Fairfax, VA.

James, S. C., and V. Boriah (2010), Modeling algae growth in an open-channel raceway, *Journal of Computational Biology*, *17*(7), 895-906.

James, S. C., C. A. Jones, M. D. Grace, and J. D. Roberts (2010a), Advances in sediment transport modelling, *Journal of Hydraulic Research*, *48*(6), 754-763.

James, S. C., E. Seetho, C. Jones, and J. Roberts (2010b), Simulating environmental changes due to marine hydrokinetic energy installations, in *OCEANS 2010*, edited by B. Spindel and T. Brockett, pp. 1-10, Seattle, WA.

James, S. C., J. Barco, E. Johnson, J. D. Roberts, and S. Lefantzi (2011), Verifying marine-hydro-kinetic energy generation simulations using SNL-EFDC, in *Oceans 2011*, edited by B. Kenoi and B. Taylor, pp. 1-9, Kona, HI.

Jones, C. A. (2001), A Sediment Transport Model, PhD thesis, 119 pp, University of California Santa Barbara, Santa Barbara.

Jones, C. A., and W. Lick (2001), Sediment erosion rates: Their measurement and use in modeling, in *Texas A&M Dredging Seminar*, edited by L. P. Sanford, pp. 1-15, ASCE, College Station, Texas.

Kaimal, J. C., and J. J. Finnigan (1994), *Atmospheric Boundary Layer Flows: Their Structure and Measurement*, Oxford University Press, New York, NY.

Katul, G. G., L. Mahrt, D. Poggi, and C. Sanz (2004), One- and two-equation models for canopy turbulence, *Boundary-Layer Meteorology*, *113*(1), 81-109.

Liu, J., T. A. Black, and M. D. Novak (1996), E-ε modeling of turbulent air flow downwind of a model forest edge, *Boundary-Layer Meteorology*, *77*(1), 21-44.

Poggi, D., A. Porporato, L. Ridolfi, J. D. Albertson, and G. G. Katul (2004), The effect of vegetation density on canopy sublayer turbulence, *Boundary-Layer Meteorology*, *111*(3), 565-587.

Raupach, M. R., and R. H. Shaw (1982), Averaging procedures for flow within vegetation canopies, *Boundary-Layer Meteorology*, *22*(1), 79-90.

Sanz, C. (2003), A note on K-ε modelling of vegetation canopy air-flow, *Boundary-Layer Meteorology*, *108*(1), 191-197.

Thanh, P. X. H., M. D. Grace, and S. C. James (2008), Sandia National Laboratories Environmental Fluid Dynamics Code: Sediment Transport User Manual, *SAND2008-5621*, 52 pp, Sandia National Laboratories, Livermore, CA.

Wilson, J. D., J. J. Finnigan, and M. R. Raupach (1998), A first-order closure for disturbed plant-canopy flows, and its application to winds in a canopy on a ridge, *Quarterly Journal of the Royal Meteorological Society*, *124*(547), 705-732.

Ziegler, C. K., and W. Lick (1986), A numerical model of the resuspension, deposition and transport of fine-grained sediments in shallow water, *Technical Report ME-86-3*, University of California, Santa Barbara, Santa Barbara, CA.