ERDC WQTN-AM-15
October 2007

A Primer for the Linkage between Unstructured Water Quality Model CE-QUAL-ICM and Structured Three-Dimensional Hydrodynamic Model CH3D-WES

by Sung-Chan Kim

PURPOSE: This Water Quality Research Program (WQRP) Technical Note (TN) describes the linkages between water quality model CE-QUAL-ICM (ICM) and three-dimensional hydrodynamic model CH3D-WES: CH3D. This TN also presents a primer for the linkages through a set of MATLAB programs. The created linkage files enable seamless operation from CH3D to ICM.

BACKGROUND: CE-QUAL-ICM was developed for a eutrophication study of Chesapeake Bay. This model, which operates in one-, two-, or three-dimensional configurations, originally had 22 state variables that can be independently activated, and is configured to interact with a fully predictive sediment diagenesis model (Cerco and Cole 1993). Currently, the number of state variables was expanded to 36. Hydrodynamic information required for ICM is generated externally prior to water quality simulations using a hydrodynamic model. Hydrodynamic information required by ICM consists of horizontal and vertical flows, cell volumes, cell dimensions, cell surface areas, and horizontal and vertical flow face areas.

The hydrodynamic model most frequently used to link to CE-QUAL-ICM is CH3D-WES. Since the inception of the Chesapeake Bay eutrophication modeling study (Johnson et al. 1993), it has been used in subsequent studies (Cerco and Noel 2004) as well as applications to other surface water systems such as Lake Washington (Kim et al. 2006, Cerco et al. 2006).

Grid Structures. CH3D uses a grid in which cells are identified in the model by their locations in a boundary-fitted Cartesian coordinate system. This type of grid is referred to as a structured grid since there is a direct relationship between a cell's identity and its location. Both the z-grid and σ-grid versions of CH3D use this approach in identifying cells and flow faces.

CH3D-z allows a different number of layers below each surface cell. All subsurface layers have a constant thickness and only the depth of the surface layer is allowed to change (Figure 1). Consequently, any change in water surface elevation results in a change in the surface layer's thickness. The water column at a given location has the same shape as the surface cell, which results in all subsurface cells having the same length and width as the surface cell. Deeper portions of a system have more layers than shallow portions, which results in the number of cells within a layer decreasing with increasing depth.

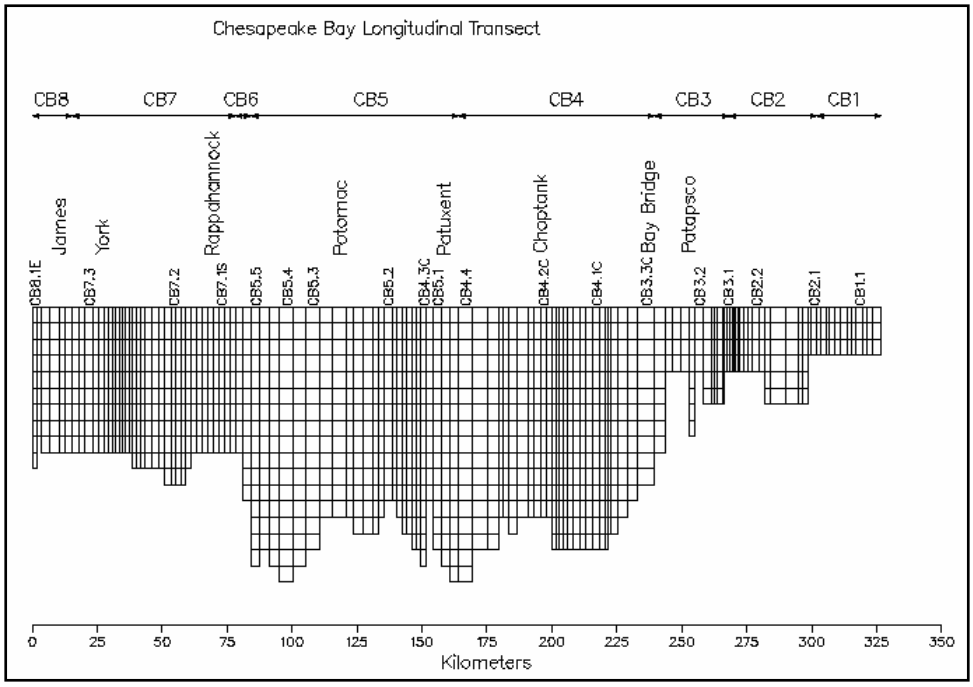

Figure 1. Elevation of Z-grid.

CH3D-σ uses a vertically σ-coordinate. Thus all the cells have the same number of subsurface layers. The water column for each cell is divided among all layers in a pre-specified ratio (Figure 2). Therefore, the thickness of all layers in a column fluctuates in response to changes in water surface elevation. CH3D-σ will have the same number of cells in all layers, which typically results in more cells than a z-grid version for the same water body.

ICM utilizes an unstructured grid in which cells are identified by a unique number referred to as a cell or box number (Figure 3). Flow faces in an unstructured grid are also identified by a unique integer number. There are no requirements concerning the methods used to assign box numbers or flow face numbers other than they be unique integer values. However, it is suggested that an orderly method be used for clarity.

Linkage Files. In a structured grid, the location of one cell in relation to another is easily determined by comparing their locations in computational domain (i.e. on i-j-k coordinate). In an unstructured grid, each cell is identified by a box (or cell) number and no such comparison is possible based upon box number information alone.

A system comprised of three ASCII data files has been developed to link CE-QUAL-ICM and CH3D-WES cells. A cell information file (e.g. "fort.94") contains the relationships between ICM box and CH3D cell and is used by CH3D (Figure 3). A file for face information (e.g. "fort.95" or "map.inp") shows relationships of a box face to adjacent boxes and is used by both CH3D and ICM (Figure 4). Because ICM's advection scheme includes both Upwind and QUICKEST, two cells must be known in both the upstream and downstream directions relative to the face. A file for the location information of ICM boxes (e.g. 'wqmgeo.inp') contains the relative location of boxes for a water column, which is used by ICM in calculating vertical dimensions. Distribution of boxes for the water column is also saved into a file (e.g. 'wqmcol.inp") for post-processing purposes. A subroutine

"WQS.F" (appended in CH3D) uses two of these linkage files ("fort.94" and "fort.95") to pass hydrodynamic information from CH3D to ICM.

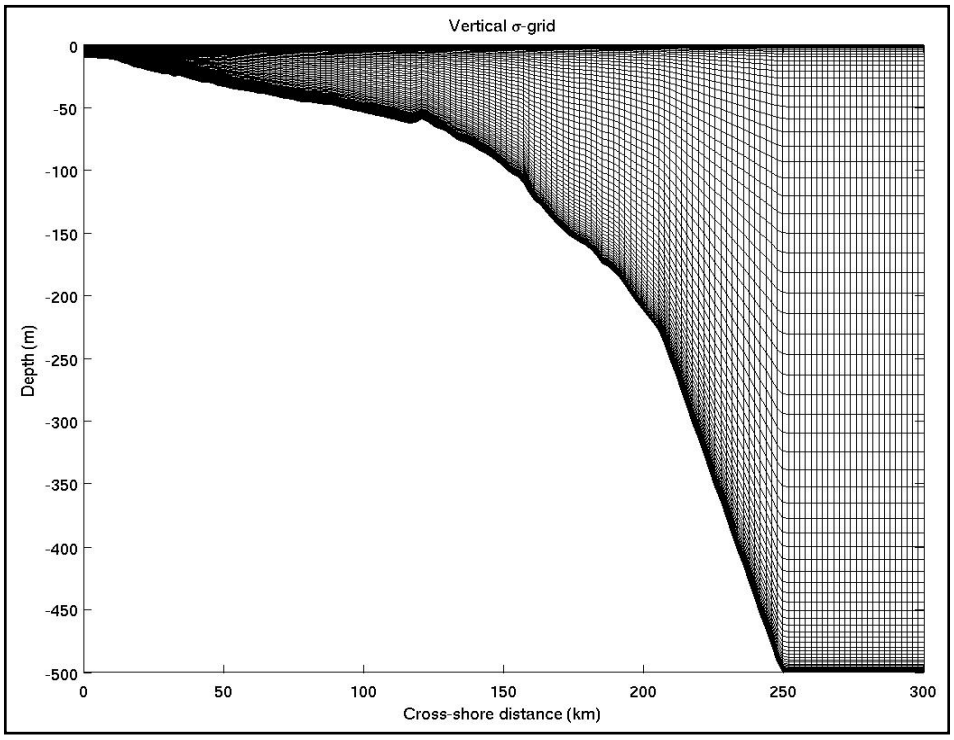

Figure 2. Elevation of σ-grid.

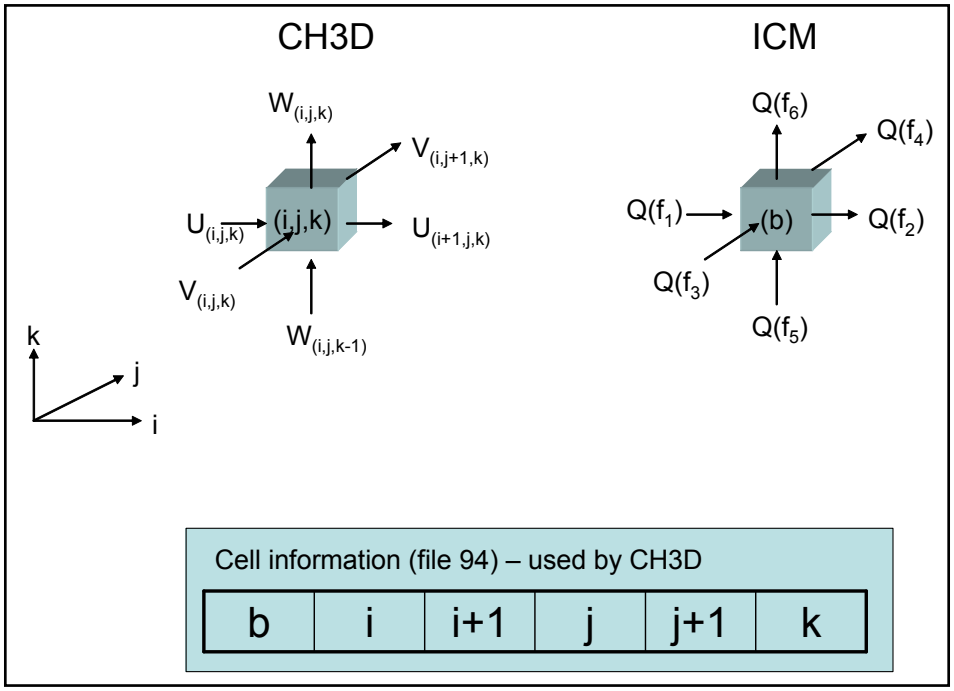

Figure 3. Diagram for cell information file.

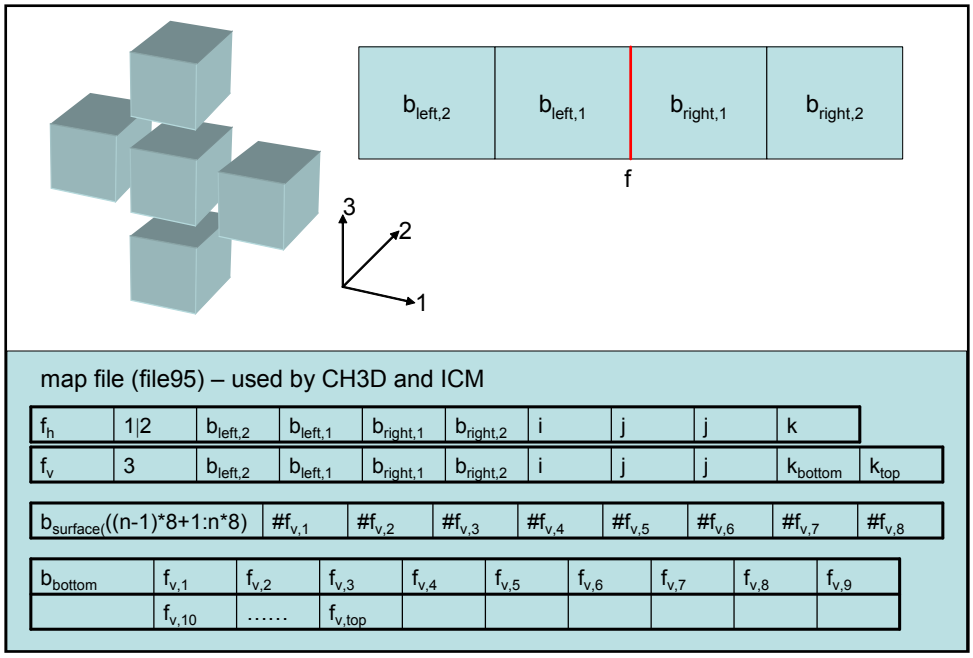

Figure 4. Diagram for face information file.

MATLAB UTILITY PROGRAMS: A program—"read_ch3d_grid_sigma.m"—was created to provide a seamless linkage between a three-dimensional hydrodynamic model (CH3D-σ) and a water quality model (CE-QUAL-ICM). For CH3D-z, a separate program—"read_ch3d_grid_z.m"—was created. Both utilize a function—plot_grid—supplied by a separate program "plot_grid.m."

CH3D is on a structured grid, i.e. on (i,j,k) grid, whereas ICM is on an unstructured grid. A linkage file should relate an ICM cell (or box) to CH3D (i,j,k) cell. The volume change calculated by CH3D is passed to ICM. An ICM cell also needs fluxes around the cell, which should be calculated by CH3D. Previously, a set of utility programs was written to make linkages and other files for pre- and post-processing. It requires an input file that has to be edited whenever there is a modification in the CH3D grid and configuration. It is cumbersome to go into the file for editing every time a change is made and also prone to error.

Development of this program accommodates any change in CH3D to generate linkage files. In addition, it has been proven to be useful to view grid information on GIS. This set of programs will work as a primer for future development of utility programs in CH3D and ICM linkages as well as ICM pre- and post-processing.

CH3D Input Files Used in Linkage Development. Three input files are needed to create linkage files for CH3D-σ and two input files are needed for CH3D-z. CH3D-σ uses separate files for the grid setup and for the model run control, respectively. In CH3D-z, these are combined into one file. Thus, z-version combines file 1 and file 2, as indicated below. Depth data (file 3) is common in both models. An additional input file (file 4) for grid information is required to prepare for a GIS application.

1. CH3D grid configuration file. This file describes the CH3D model grid setup, such as dimension and boundary conditions. Typically, it bears a name like "blk01.inp." An example is given in Appendix A.

    a. Parameters needed for linkage from the CH3D model grid dimension (ICELLS, JCELLS, KCELLS).

    b. River boundaries provide flux boundary conditions:
        (1) NRIVER: number of river boundaries. This should not be confused with the actual number of river boundaries. This number only states how many lines follow to describe boundaries.
        (2) IJDIR: direction of river boundary relative to a cell. For a cell, 1 is set at the left (west) side of the cell. Numbers increase counterclockwise along the cell edges. The bottom is 2, the right side is 3, and the top is 4. All directions are relative to the (i,j) coordinate, i varying in the left-right direction and j varying in the bottom-top direction.
        (3) IJROW: fixed coordinate. If IJDIR=1 or 3, i is fixed. If IJDIR=2 or 4, j is fixed.
        (4) IJRSTR: starting point in varying coordinate. If IJDIR=1 or 3, this is the j location. If IJDIR=2 or 4, this is the I location.
        (5) IJREND: ending point in varying coordinate (similar to IJRSTR)

The program utilizes the names of the rivers (two words recommended).

    c. Bars:
        (1) NBAR: number of bars to block flows across the faces.
        (2) IJBDIR: direction of bar. For a cell, 1 is set at lower (south) side of the cell and 2 is set at the left (west) side of the cell.
        (3) IJBROW: fixed coordinate. If IJBDIR=1, j is fixed. If IJBDIR=2, i is fixed.
        (4) IJBSTR: starting point in varying coordinate. If IJBDIR=1, this is i location. If IJBDIR=2, this is j location.
        (5) IJBEND: ending point in varying coordinate (similar to IJBSTR)

    d. Ocean boundaries provide mass (or volume) boundary conditions:
        (1) TIDBND: number of lines for tide boundary conditions.
        (2) IJTDIR: location of the boundary cell relative to successive water cell(s). It follows the convention of IJRDIR. It should be noted that this is for a cell, not for a face. Thus the flow face directions for open boundary conditions should be orthogonal to those of river boundaries. Thus, IJTDIR=1 is fixed along i=IJTROW+1. IJTDIR=2 is fixed along j=IJTROW+1. Similarly, IJTDIR=3 is fixed along i=IJTROW and IJTDIR=4 is fixed along j=IJTROW.
        (3) IJTROW: fixed coordinate for the boundary cells.
        (4) IJTSTR: starting point on varying coordinate.
        (5) IJTEND: ending point on varying coordinate.

An open boundary should not be placed at a land cell. This has happened occasionally because the CH3D model can be run with this condition, but this is not correct. If this happens, the program will not run.

2. CH3D run control file. This file controls the CH3D run. Typically it is called "main.inp." Appendix B shows an example of this file. Relevant parameters in this file needed for linkage development include the following:
   a. DT: time step (in seconds)
   b. IT1: time step for starting run
   c. IT2: time step for ending run
   d. ITSALT: time step for spin-up. It is recommended this will be the same as the beginning of hydrodynamic information archiving from CH3D.
   e. XMAP: map scale (in centimeters per unit)

In the z-version, there is only one file (default name is "fort.4") corresponding to combined information of block control (file 1) and run control (file 2).

3. Depth data. Depth data (in centimeters) are arranged as (ICELLS×JCELLS) matrix. Typically this is called "fort.23" for the σ-version and "fort.50" for the z-version.

4. Grid data. This file contains locations of grid points and includes information on the matrix size (IMAX×JMAX). This should be the same as ((ICELLS+1)×(JCELLS+1)). Inactive nodes have very big numbers such as 9×10^{19}. This file is typically known as "grid.inp" for the sigma version and "fort.15" for the z version.

Output Files. These programs create six output files. The first two are required by CH3D. The second and third files are required by ICM. The second file is common to both CH3D and ICM. The fourth file is useful in creating boundary conditions and the fifth file is used by post-processing programs. The sixth file is to be used in ArcView GIS.

1. Cell information file. This file is used by CH3D subroutine "WQMOUT.f" to create a hydrodynamic file for ICM. Figure 3 is a diagram of relationships between CH3D cells and ICM boxes. The most commonly known file name is "fort.94". Appendix C shows an example of this file. Important key words are as follows:
   a. NSB: number of surface boxes
   b. NAVG: number of time-steps for averaging (default is set for 1 hr)
   c. ITWQS: time-step to begin hydrodynamic file archiving (default is the same as spin-up time-step)
   d. TBOX: total number of boxes
   e. BOX_NO: box number
   f. IFIRST: i coordinate
   g. ILAST: i+1 coordinate
   h. JFIRST: j coordinate
   i. JLAST: j+1 coordinate
   j. K: k coordinate

2. Face information file. This file is also used in the CH3D subroutine "WQMOUT.f" as well as by ICM. Figure 4 shows the conceptual framework of this map file (commonly known as either "fort.95" or "map.inp"). Figure 4 shows the conceptual diagram for this file. Appendix D shows an example of this file. Key words are as follows:
   a. NHQFT: total number of horizontal faces

b. NQF: total number of faces. Thus, number of vertical faces becomes NQF-NHQFT
c. NHQF: number of horizontal faces in surface layer
d. F: face number
e. QD: flow direction (1: i-direction, 2: j-direction, 3:k-direction)
f. ILB: box number of second upstream cell
g. IB: box number of immediate upstream cell
h. JB: box number of immediate downstream cell
i. JRB: box number of second downstream cell
j. KP: i coordinate if QD=1. j coordinate if QD=2 or 3
k. KF,KL: j coordinate if QD=1. i coordinate if QD=2 or 3
l. LAYER: k coordinate if QD=1 or 2. If Q=3, two numbers show the k koordinates of boxes beneath and on the face.
m. SFC BOX #: number of surface boxes
n. NVF: number of faces orthogonal to vertical direction
o. BOT BOX #: number of bottom box
p. VFN: face numbers orthogonal to vertical direction (upward from bottom)

Boundary faces are identified by either (ILB=0 and IB=0) or (JB=0 and JRB=0).

3. ICM box location file. This file shows locations of ICM boxes in three dimensions (Figure 5). Appendix E shows an example ("wqmgeo.inp"). The left column is a box number and the right column is the box number above the box. At the surface layer, no box exists above a box; thus, the right column becomes 0.

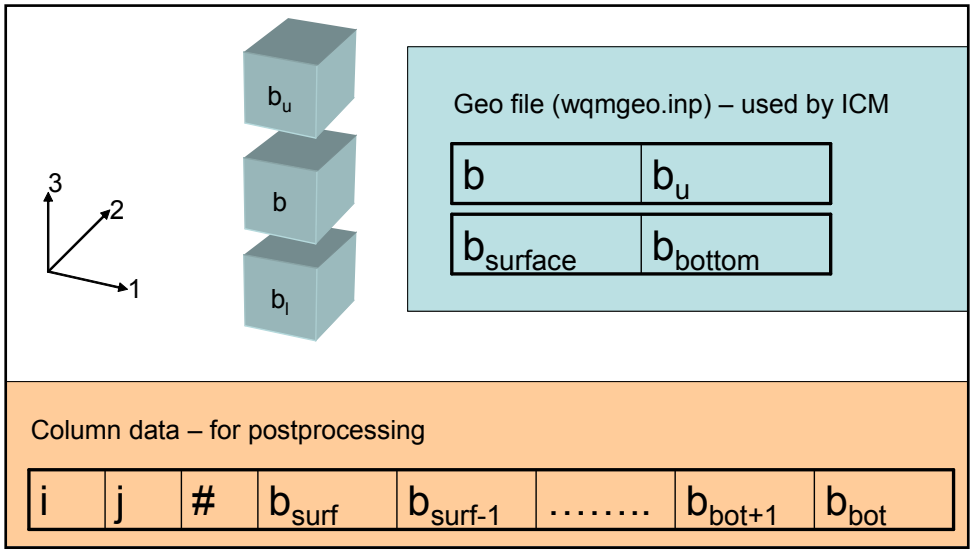

Figure 5. Box location information files.

4. ICM boundary face data file. This is used in preprocessing of ICM boundary conditions. Appendix F shows an example ("bndface.inp"). The first set of the file contains six columns of data.
   a. Column 1: boundary face number count
   b. Column 2: face number

c. Column 3, 4, 5: (i,j,k) coordinate of the face
d. Column 6: name of the boundary (if any)

The second set of the file arranges boundary sequences according to specific boundaries; that is, a specific river boundary or an ocean boundary.

5. File utilized for CH3D-ICM postprocessing. This file ("wqmcol.inp") is used by many post-processing programs. The first two columns contain (i,j) coordinates of corresponding CH3D cells. The next column is the number of boxes in the vertical direction, followed by the box numbers. They are arranged from surface to bottom.

6. File utilized for ET GeoWizards to create a shape file. ET GeoWizards™ is a utility program to be used in ESRI ArcView. It can import text files to create shape files. This Matlab program generates a text file that can be used by ET GeoWizards. The default name is "CH3D_ICM_GIS.txt."

Running the Program. First, one should open the MATLAB program. Next, change the directory to the program location. Then, type in "read_ch3d_grid_sigma" or "read_ch3d_grid_z" (Figure 6). An alternative is to bring in the source code to MATLAB editor and run by invoking "debug."

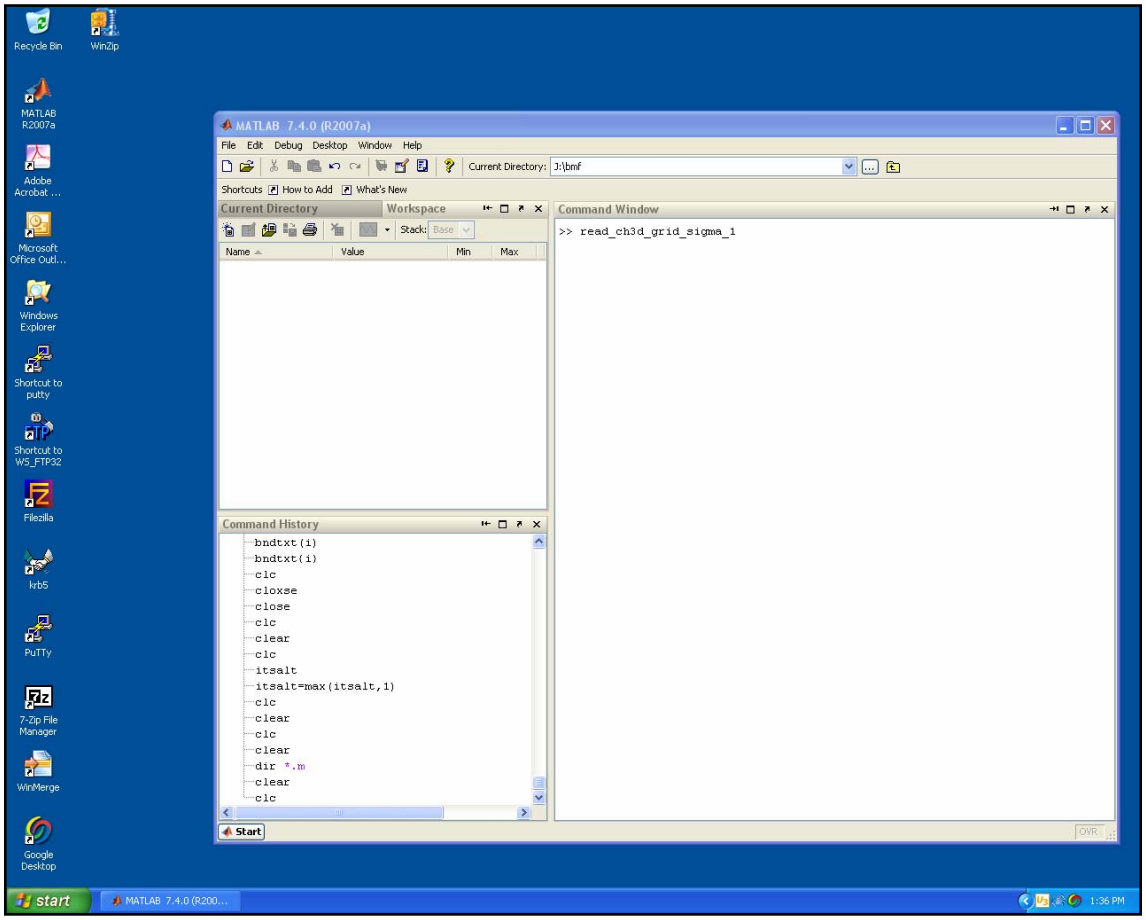

Figure 6. Initializing the program.

The program will ask for the CH3D grid configuration file (Figure 7). In this example, "blk01.inp" is selected. The file is shown in Appendix A. Next, the program will ask for CH3D run control file (Figure 8). In this example, it is "main.inp." For the z-grid version, it will ask for only one file ("fort.4"). This file is shown in Appendix B. The third input file is the depth data (Figure 9). In this example, it is "fort.23."

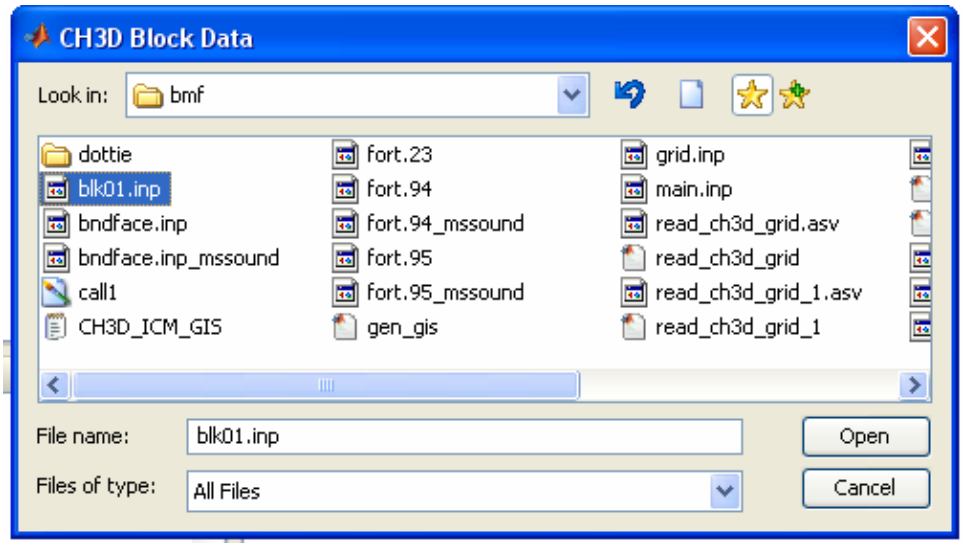

Figure 7. Selecting CH3D grid configuration file.

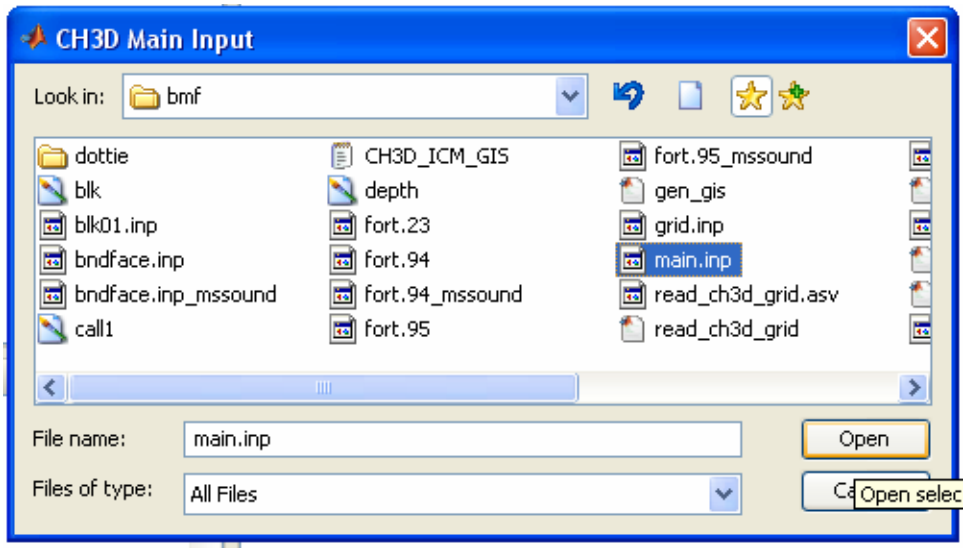

Figure 8. Selecting CH3D run control file.

The program will generate a plot of the grid (Figure 10). All the active cells are represented by blue boxes. This also depicts the locations of river boundaries as red arrows (Figure 11). Note that a flux condition is set by CH3D input file ("fort.13") when an arrow crosses a face of a cell. CH3D also reads in temperature conditions at these faces ("fort.78"). Ocean boundaries are depicted as red squares with a cross inside (Figure 12). The water levels are set at these cells in CH3D ("fort.16"). CH3D also sets salinity and temperature at these boundary cells ("fort.76").

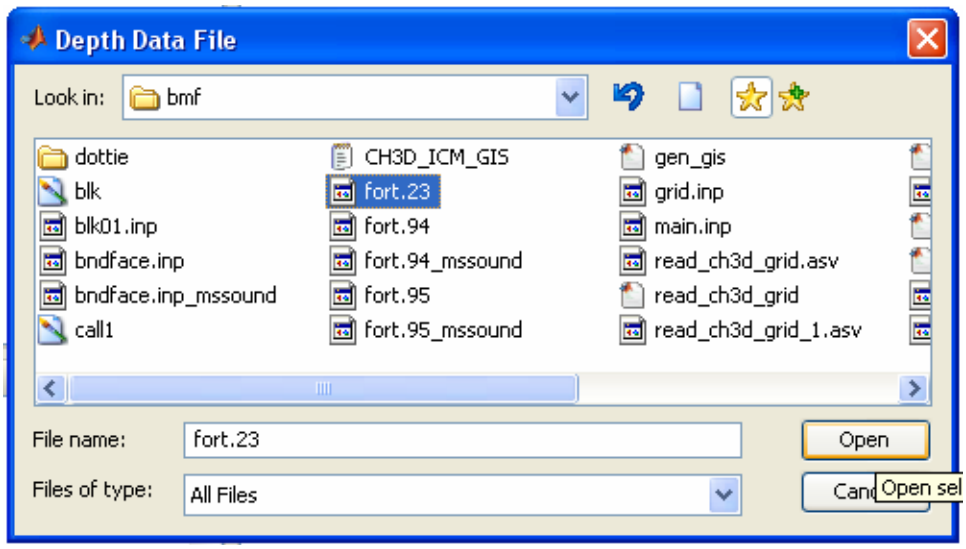

Figure 9. Selecting depth data.

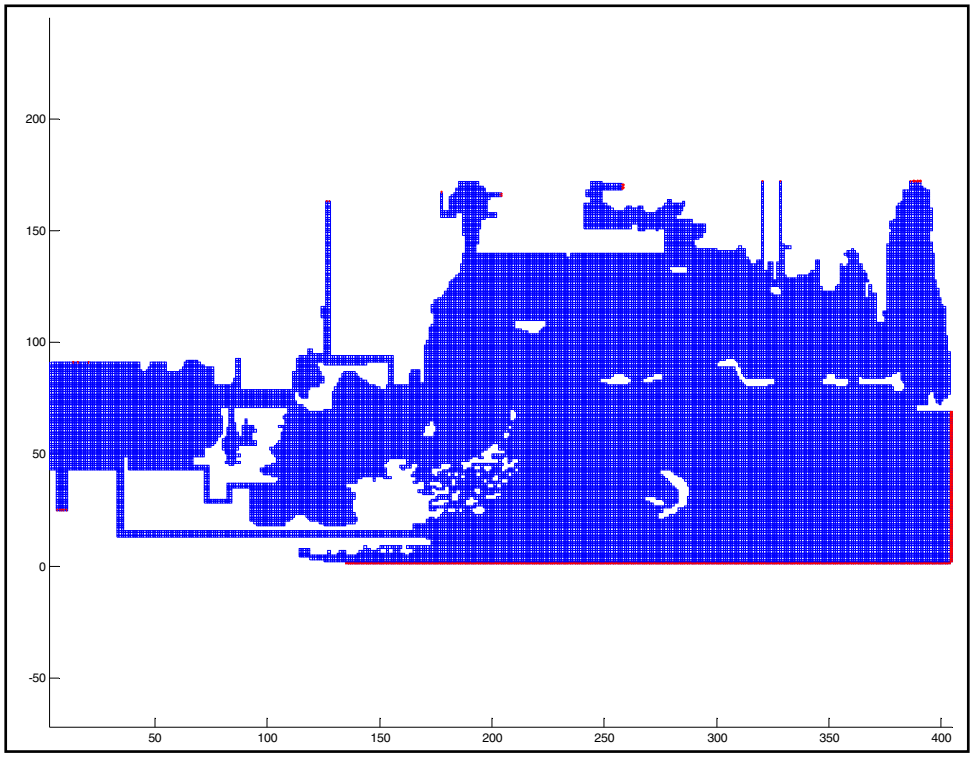

Figure 10. Plot grid with boundary conditions. Blue boxes represent water cell. Red crosses represent open boundaries and red arrows represent river boundaries.

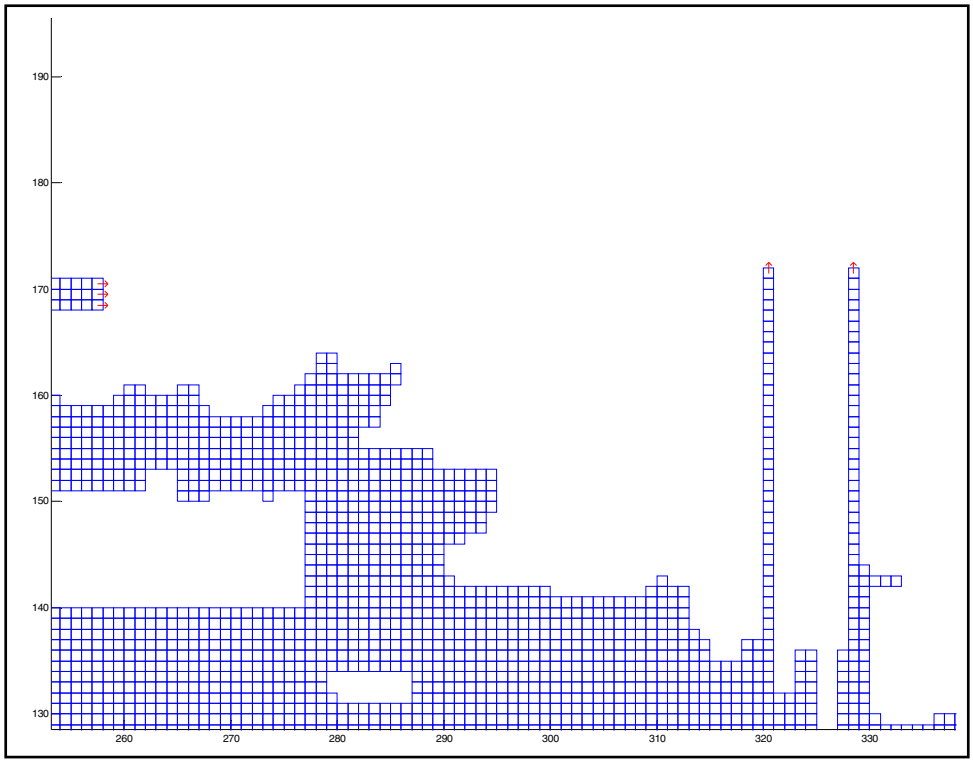

Figure 11. Zoomed in grid to depict river boundaries (red arrows).

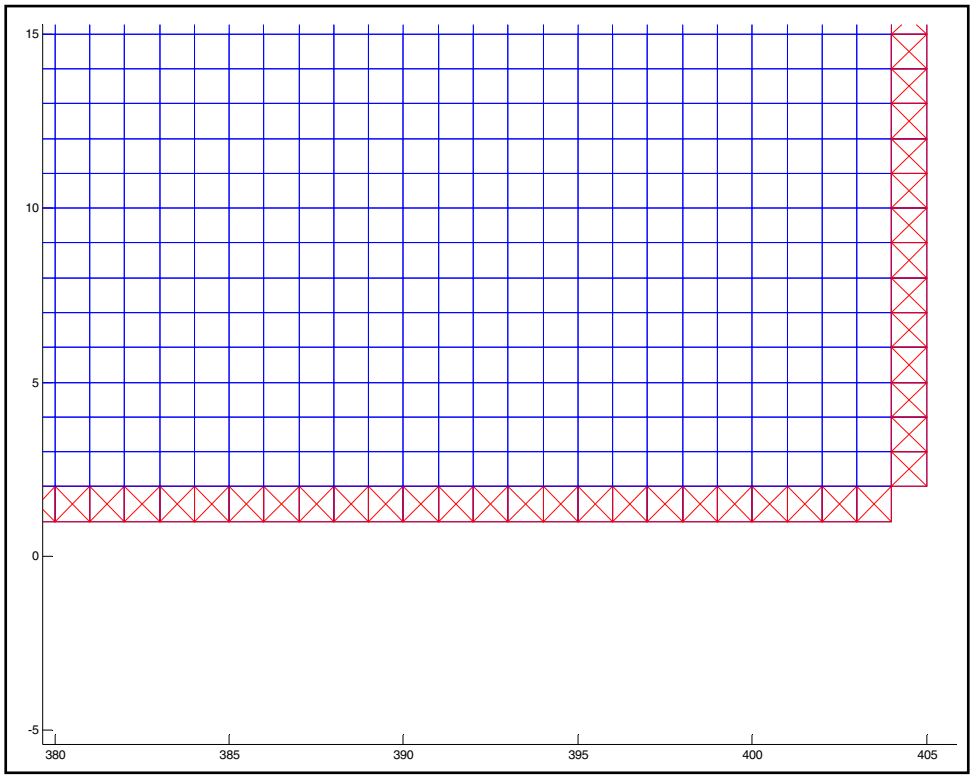

Figure 12. Zoomed in grid to show open boundary cells (red boxes with cross inside).

The following user inputs (see Figures 13 through 17) are to name output files for linkage. To prepare for GIS application, a grid file for input (Figure 18) and a text output file (Figure 19) need to be set.

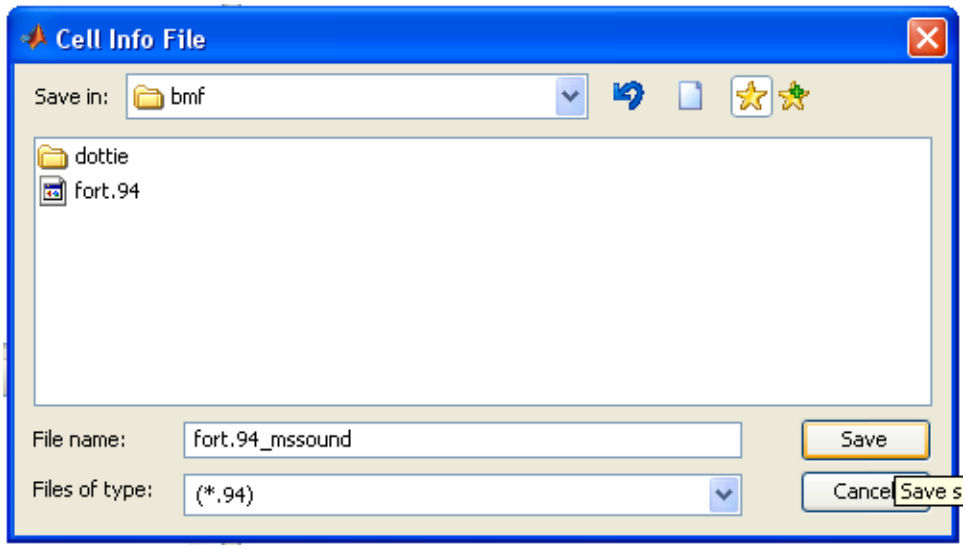

Figure 13. User input for box-cell linkage file (file 94 for CH3D).

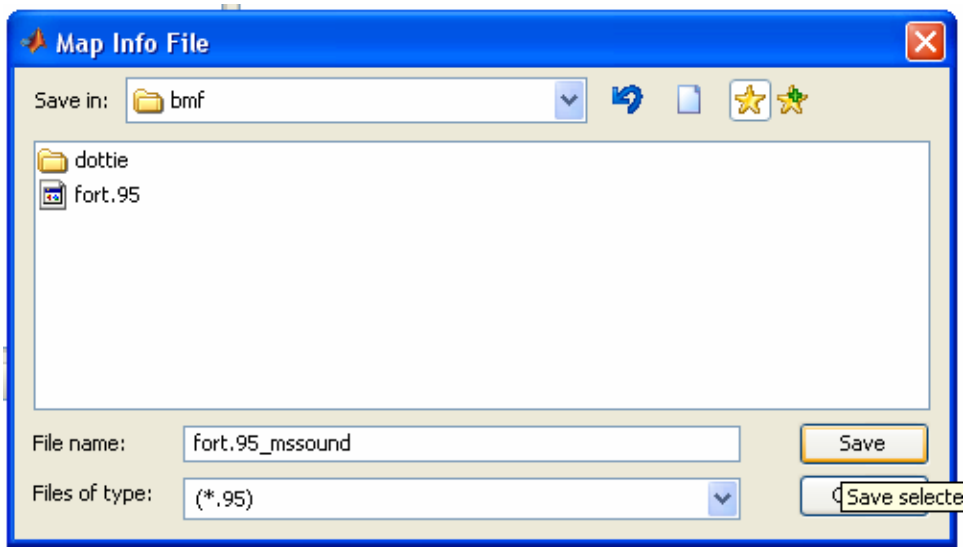

Figure 14. User input for map file (file95 for CH3D).

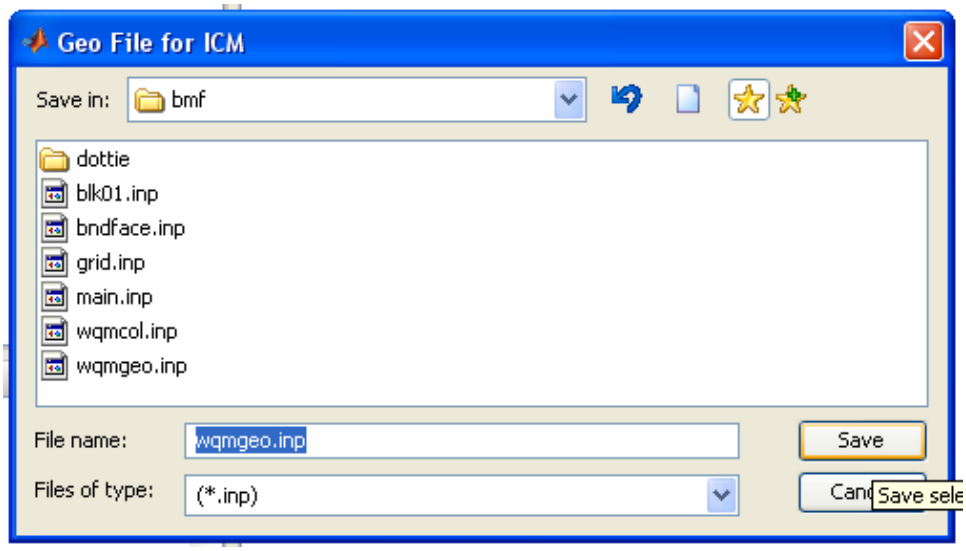

Figure 15. User input for ICM box location file.

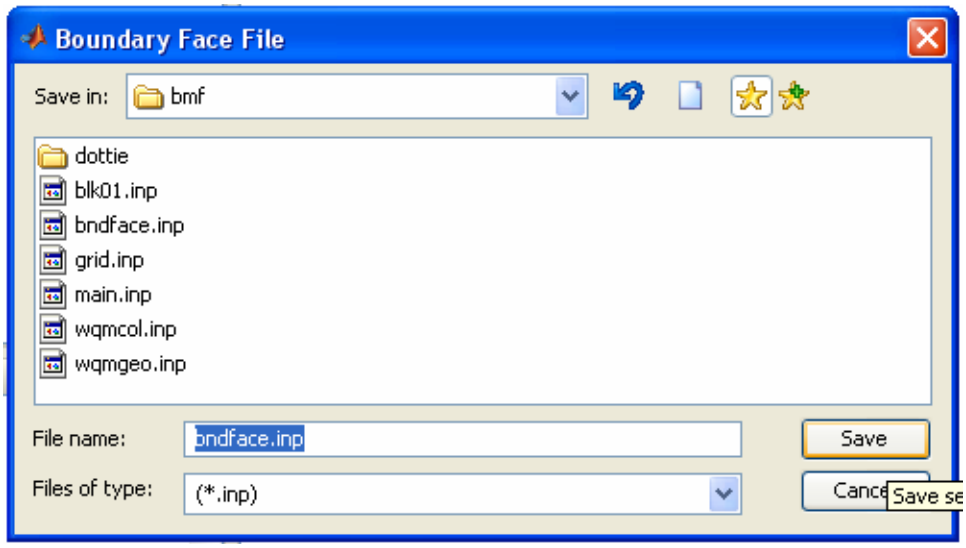

Figure 16. User input for boundary face information file to aid ICM preprocessing.

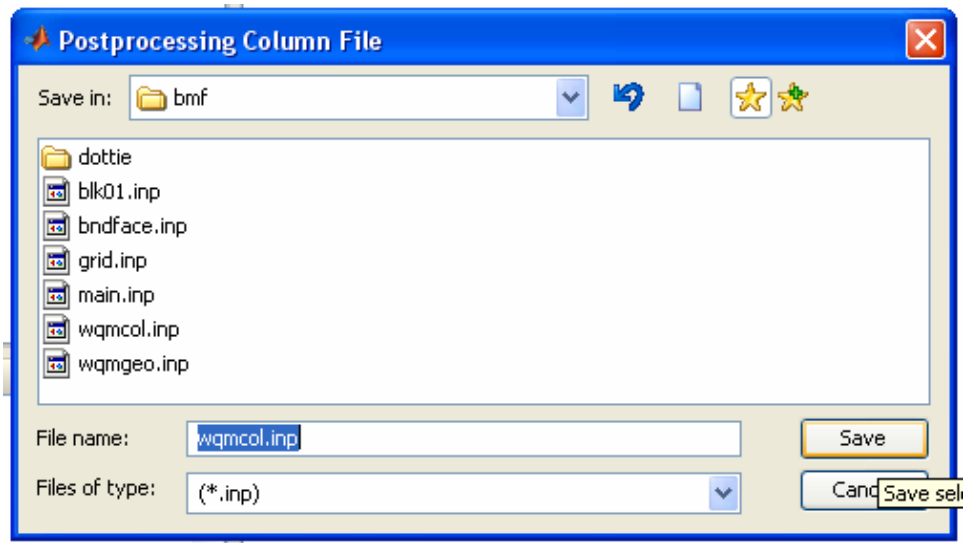

Figure 17. User input for ICM post processing water column information file.

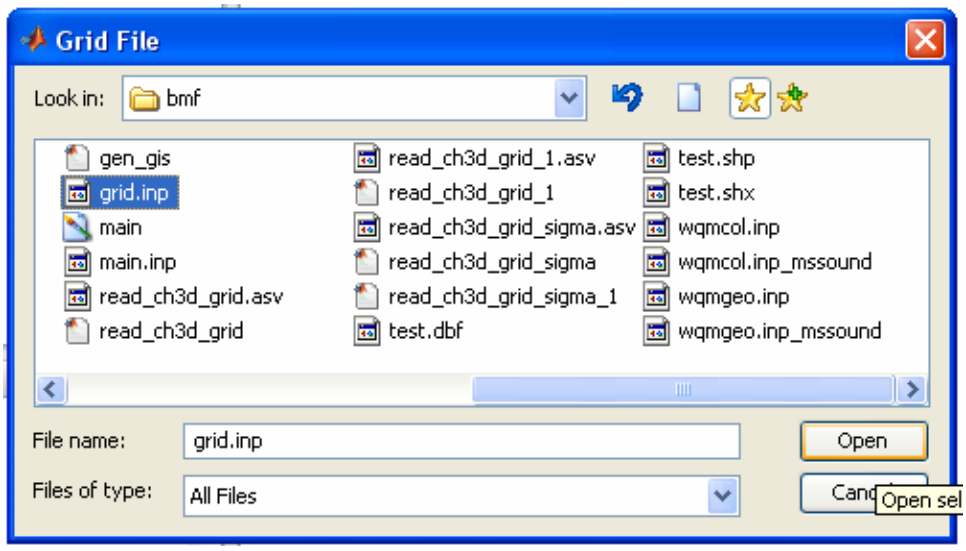

Figure 18. Opening CH3D grid information (“grid.inp”).

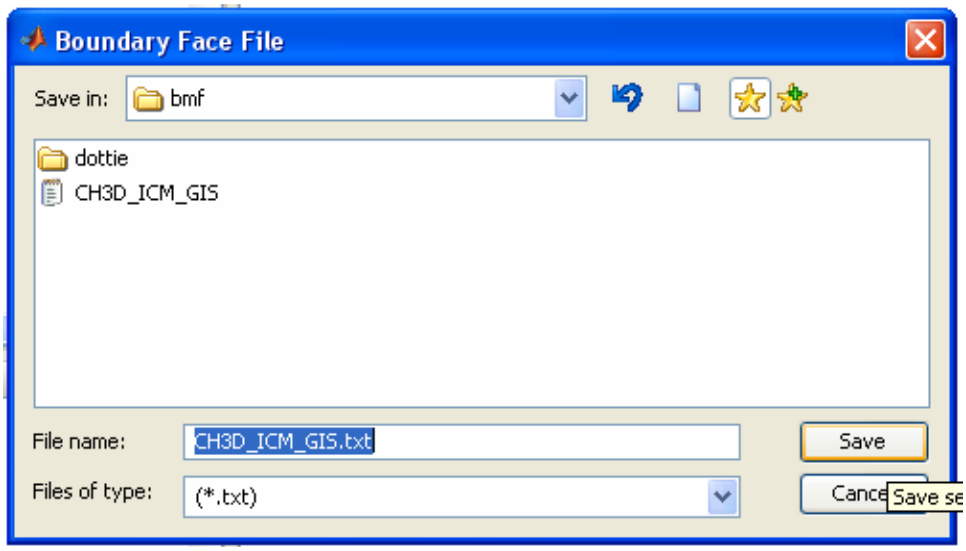

Figure 19. User input for a text file to be utilized by ET GeoWizard to create ArcView shape file.

Figures 20 through 23 show how to create a shape file using ET GeoWizards inside ArcMap from the file generated, "CH3D_ICM_GIS.txt." The "Generate (import from text)" menu is selected (Figure 21). Then, input and output files are specified (Figure 22). Note that "Polygon" type is selected. Check the box for "Data contains attributes" (Figure 23). "CH3D" is (i,j) location of surface cell (=1000*i+j). "Depth" is the water depth of the cell. "L1" is the surface box number for ICM. "Ln" is the ICM box number on n-th layer (surface is 1). Figure 24 shows a created shape file layout.

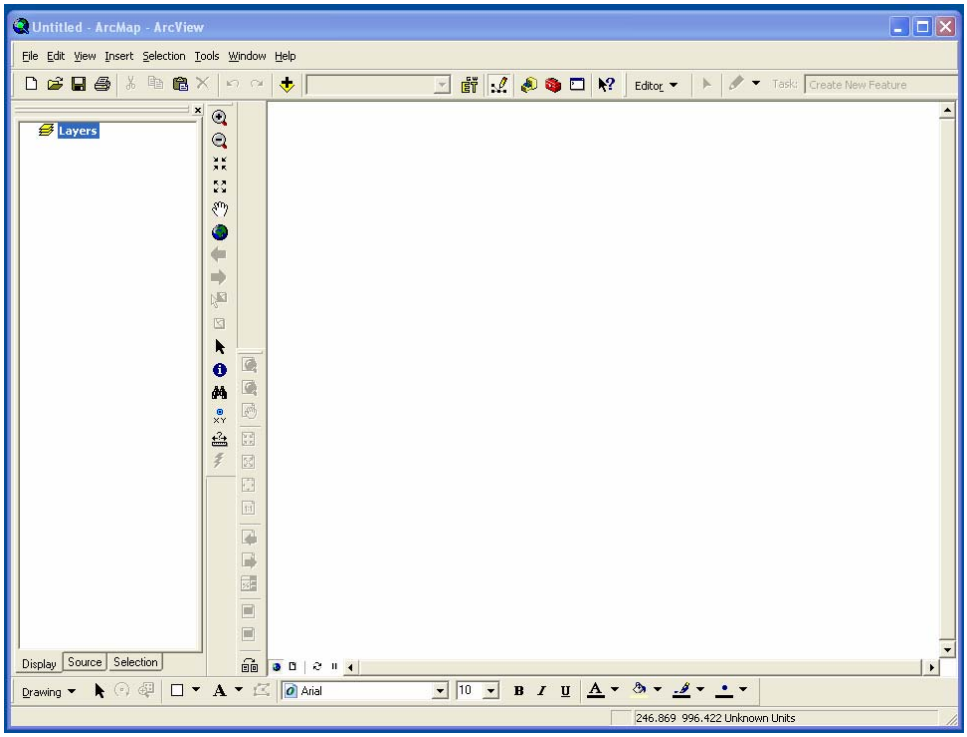

Figure 20. Initial screen for ArcMap.

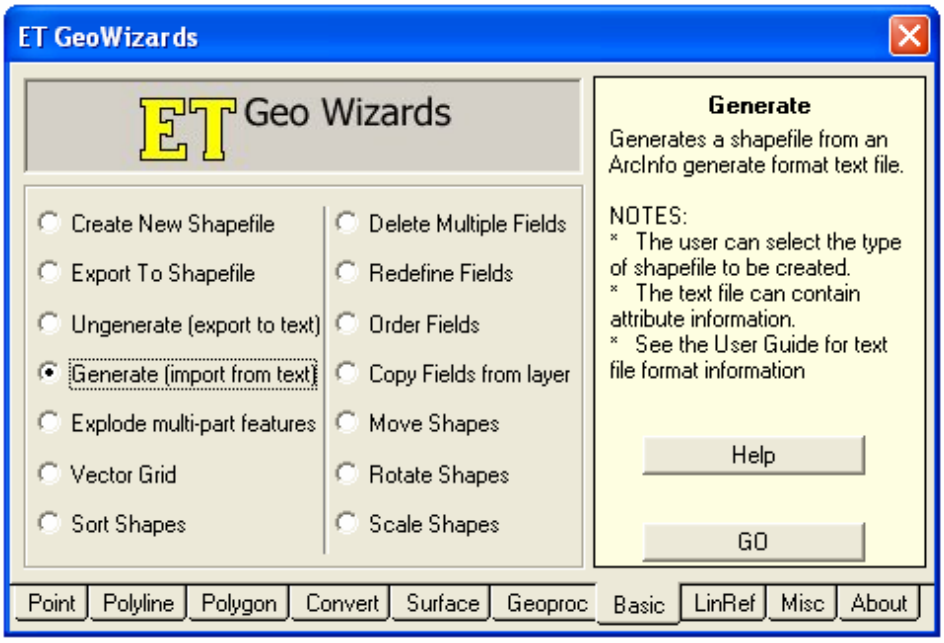

Figure 21. Initialize ET GeoWizard.

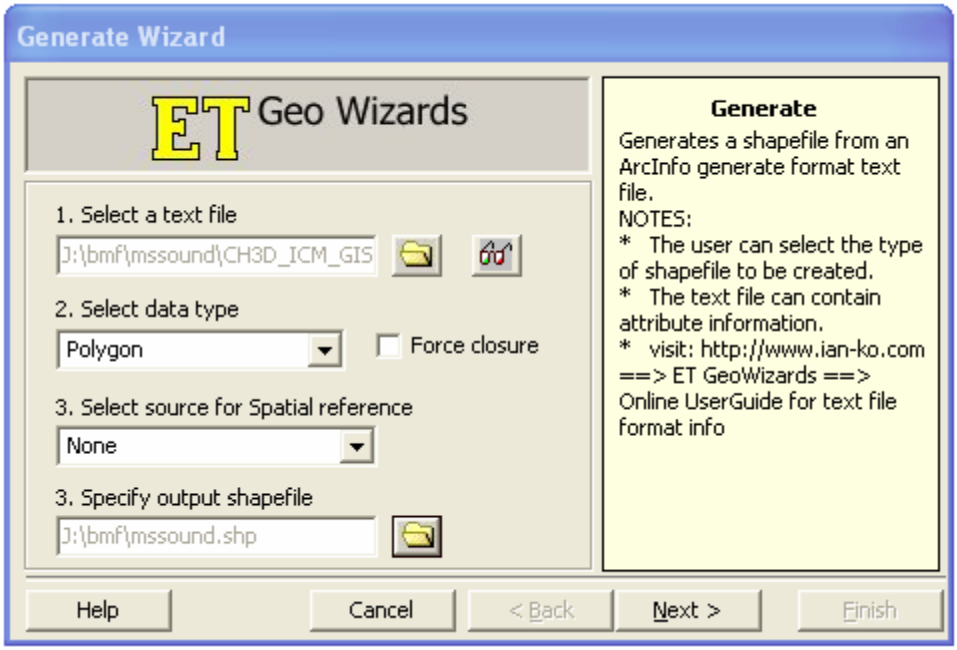

Figure 22. Import prepared text file to make a shape file.

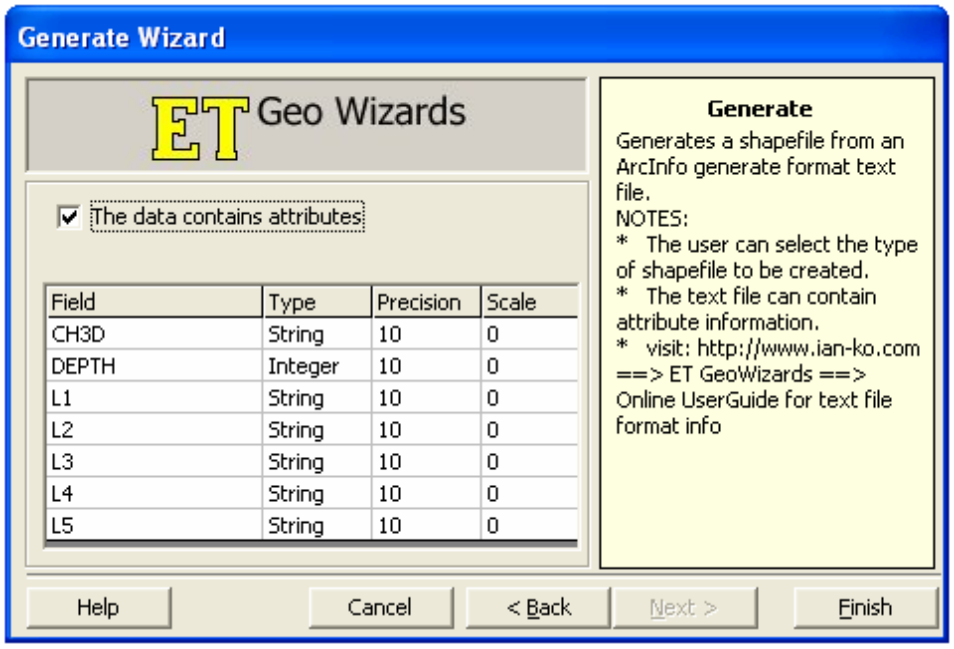

Figure 23. Set attributes.

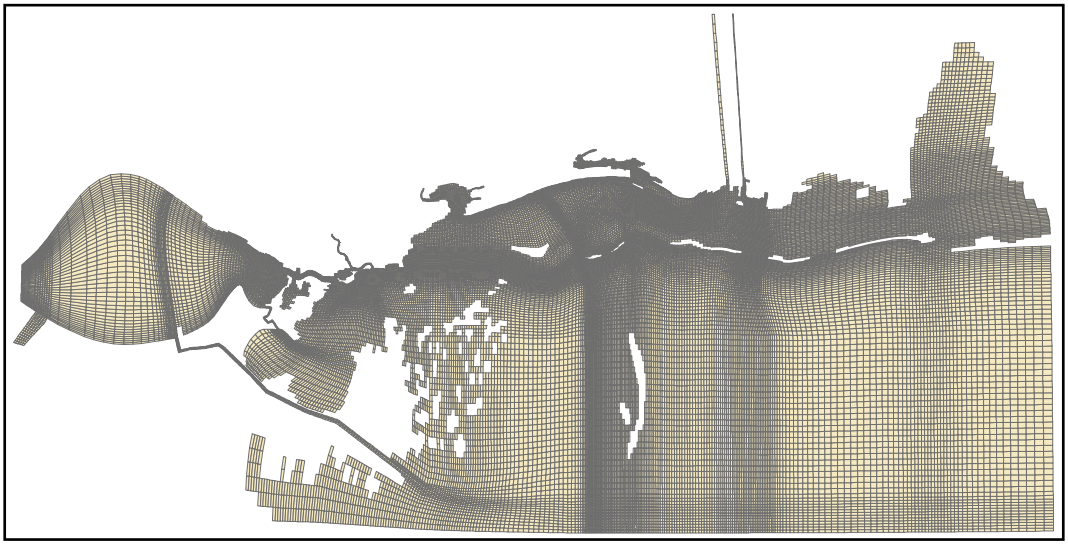

Figure 24. Example of a created GIS shape file.

Appendix G is the source code to deal with CH3D-σ, "read_ch3d_sigma.m." Appendix H is the source code to deal with CH3D-z, "read_ch3d_z.m." Appendix I is the source code for function "plot_grid_1."

SUMMARY: A set of MATLAB programs were developed to generate seamless linkages from CH3D-WES to CE-QUAL-ICM. Extra features such as files to aid in pre- and post-processing and files to be used in creating a GIS file are included. User instructions are also provided.

POINT OF CONTACT: This technical note was written by Dr. Sung-Chan Kim (*sung-chan.kim@erdc.usace.army.mil*, 601-634-3783) of the U.S. Army Engineer Research and Development Center, Environmental Laboratory. Questions about this technical note can be addressed to Dr. Kim or to the manager of the Water Operations Technical Support (WOTS) Program, Robert C. Gunkel (601-634-3722, *Robert.C.Gunkel@erdc.usace.army.mil*). This technical note should be cited as follows:

> Kim, S.-C. 2007. *A Primer for the linkage between unstructured water quality model CE-QUAL-ICM and structured three-dimensional hydrodynamic model CH3D-WES.* Water Quality Technical Notes Collection (ERDC WQTN-AM-15), Vicksburg, MS: U.S. Army Engineer Research and Development Center. An electronic copy of this TN is available at *http://el.erdc.usace.army.mil/publications.cfm?Topic=technote&Code=watqual*.

REFERENCES

Cerco, C. F., and T. Cole. 1993. Three-dimensional eutrophication model of Chesapeake Bay. *J. Environ. Eng.* 119:1006-1025.

Cerco, C. F., and M. Noel. 2004. *The 2002 Chesapeake Bay eutrophication model*. EPA 903-R-04-004. Annapolis, MD: Chesapeake Bay Program Office, U.S. Environmental Protection Agency.

Cerco, C. F., M. Noel. and S.-C. Kim. 2006. Three-dimensional management model for Lake Washington, Part II: Eutrophication modeling and skill assessment. *Lake and Reserv. Manage.* 22:115-131.

Johnson, B. H., K. W. Kim, R. E. Heath, B. B. Hsieh, and H. L. Butler. 1993. Validation of a three-dimensional hydrodynamic model of Chesapeake Bay. *J. Hydraul. Eng.* 119: 2-20.

Kim, S.-C., C. F. Cerco, and B. H. Johnson. 2006. Three-dimensional management model for Lake Washington, Part I: Introduction and hydrodynamic modeling. *Lake and Reserv. Manage.* 22: 103-114.

Appendix A. Example CH3D grid configuration file

```
Grid 405x172 MSSOUND Study
ICELLS  JCELLS   KCELLS
404     171      5
NRANG
  0
RANGDR   RPOS1   RPOS2   RPOS3  RRNAME
NSTA   NFREQ   NSTART    (CURRENT STATIONS)
   0   99929   999991     89.01166,30.22833
IST JST  STATID(K)    (2I4,A48)     ( ONE CARD FOR EACH STATION )
NSTAS  NFREQS  NSTRTS    (TIDE STATIONS)
   0   99929   999991     89.01166,30.22833
IST JST  STATID(K)    (2I4,A48)     ( ONE CARD FOR EACH STATION )
MSTA   MFREQ  MSTART    (SALINITY STATIONS)
   0   99999  999999
IST JST  STATID(K)    (2I4,A48)     ( ONE CARD FOR EACH STATION )
NRIVER
    11
IJRDIR  IJRROW  IJRSTR  IJREND    ( ONE CARD FOR EACH RIVER )
     2      25       6      10     S-W Lake Pont River
     4      90      13      13     Tickfaw River
     4      90      15      15     Tangipahoa River
     4      90      20      20     Tchekfuncta River
     4     162     126     127     Pearl River
     4     166     177     177     Jordan River
     3     203     165     166     Wolf River
     3     257     168     170     Biloxi River
     4     171     320     320     West Pascagula River
     4     171     328     328     East Pascagula River
     4     171     386     390     Mobile Rivers
     I       J  QRIVER            ( ONE CARD FOR EACH CELL )
NBAR    NBARU    KU    NBARV   KV
   0       0       0       0       0
IJBDIR  IJBROW  IJBSTR  IJBEND    ( ONE CARD FOR EACH BAR )
TIDFNO  TIDBND
     6       5
TIDSTR   2    3    4    5    6    7    8    9    10
     1   1    1    1    1    1    1    1    1    1
  IJTDIR  IJTROW  IJTSTR  IJTEND  TIDTYP  TIDFN1  TIDFN2
       2       1     135     231INTERP         2       3
       2       1     231     325CONSTANT       3       3
       2       1     325     403INTERP         3       4
       3     404       2      39INTERP         4       5
       3     404      39      68INTERP         5       6
RESET HS(I,J) TO ZERO AT THE FOLLOWING CELLS
RESET HU(I,J) TO ZERO AT THE FOLLOWING CELLS
RESET HV(I,J) TO ZERO AT THE FOLLOWING CELLS
RESET HS(I,J) TO THE FOLLOWING DEPTHS
   END OF DATA
   END OF FILE
```

Appendix B. Example of CH3D run control file

```
MS Sound Study 01-31 March 1998 DT60=044640
DT                              DT30=089280
60.0
IT1   IT2  ISTART  ITSALT  IHOT  IADI  IFREQP   IWQ
1  044640     0    001440    0    0    999999    1
XREF   ZREF  UREF   COR     GR    RO0   ROR   T0   TR
11000  2000  50.0 0.00004  981.0  1.0   2.0   10   20
THETA
1.0
ITEMP  ISALT   IFI   IFD
  0     -1      0     0
 TWE   TWH   FKB
 10    10     1
IEXP   IAV   AVR   AV1   AV2   AVM     AVM1   AHR
  1     0     1     0     0    50.0   0.001   500
GAMAX   GBMAX
1000      10
IWIND  TAUX    TAUY
  3    0.00    0.00
ISPAC(I),I=1,10
  0    0    0     1     0     0     0     0     0     0
JSPAC(I),I=1,10
  0    0   -1     1     0     0     0     0     0     0
RSPAC(I),I=1,10
0.020  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
HADD    HMIN    H1      H2    SSS0
0.0     1.0     0.0     0.0   0.0
ISMALL   ISF    ITB  ZREFBN   CTB    BZ1  ZREFTN    TZ1
  1       0     2    10.0  0.00125  1.00    1.0     1.0
XMAP    ALXREF  ALYREF
100.00    0       0
```

Appendix C. Example of header lines for cell information ("fort.94")

```
File 94: box info for CH3D
SCK
09-May-2007
     NSB    NAVG   ITWQS    TBOX
   40406      60    1440  202030
  BOX_NO  IFIRST   ILAST  JFIRST   JLAST       K
       1     125     126       2       3       5
       2     126     127       2       3       5
       3     127     128       2       3       5
       4     128     129       2       3       5
       5     129     130       2       3       5
       6     130     131       2       3       5
       7     131     132       2       3       5
       8     132     133       2       3       5
       9     133     134       2       3       5
      10     134     135       2       3       5
      11     135     136       2       3       5
      12     136     137       2       3       5
      13     137     138       2       3       5
      14     138     139       2       3       5
      15     139     140       2       3       5
      16     140     141       2       3       5
      17     141     142       2       3       5
      18     142     143       2       3       5
      19     143     144       2       3       5
```

Appendix D. Example map file (“fort.95”)

```
File 95: Face info for CH3D and ICM
SCK
09-May-2007
:
:
   NHQFT     NQF    NHQF
  395335  556959   79067
       F      QD     ILB      IB      JB     JRB      KP      KF      KL   LAYER
       1       1       0       1       2       3     126       2       2       5
       2       1       1       2       3       4     127       2       2       5
       3       1       2       3       4       5     128       2       2       5
.....................................................................................
   39522       1   40404   40405   40406       0     390     171     171       5
   39523       2       0   11212   11554   11896      44       3       3       5
   39524       2   11212   11554   11896   12234      45       3       3       5
   39525       2   11554   11896   12234   12570      46       3       3       5
   39526       2   11896   12234   12570   12909      47       3       3       5
   39527       2   12234   12570   12909   13266      48       3       3       5
   39528       2   12570   12909   13266   13637      49       3       3       5
   39529       2   12909   13266   13637   14002      50       3       3       5
   39530       2   13266   13637   14002   14368      51       3       3       5
   39531       2   13637   14002   14368   14742      52       3       3       5
   39532       2   14002   14368   14742   15118      53       3       3       5
   39533       2   14368   14742   15118   15503      54       3       3       5
   39534       2   14742   15118   15503   15891      55       3       3       5
   39535       2   15118   15503   15891   16277      56       3       3       5
   39536       2   15503   15891   16277   16660      57       3       3       5
   39537       2   15891   16277   16660   17035      58       3       3       5
   39538       2   16277   16660   17035   17411      59       3       3       5
   39539       2   16660   17035   17411   17790      60       3       3       5
   39540       2   17035   17411   17790   18166      61       3       3       5
   39541       2   17411   17790   18166   18543      62       3       3       5
.....................................................................................
  395335       2  190682  190926  191165  161624      95     403     403       1
  395336       3       0  161625  121219   80813       2     125     125       1       2
  395337       3  161625  121219   80813   40407       2     125     125       2       3
  395338       3  121219   80813   40407       1       2     125     125       3       4
  395339       3   80813   40407       1       0       2     125     125       4       5
  395340       3       0  161626  121220   80814       2     126     126       1       2
  395341       3  161626  121220   80814   40408       2     126     126       2       3
  395342       3  121220   80814   40408       2       2     126     126       3       4
  395343       3   80814   40408       2       0       2     126     126       4       5
  395344       3       0  161627  121221   80815       2     127     127       1       2
  395345       3  161627  121221   80815   40409       2     127     127       2       3
  395346       3  121221   80815   40409       3       2     127     127       3       4
  395347       3   80815   40409       3       0       2     127     127       4       5
.....................................................................................
  556958       3  161624  121218   80812   40406     171     390     390       3       4
  556959       3  121218   80812   40406       0     171     390     390       4       5

  SFC BOX #     (NVF(SB), SB=1,NSB)
    1-    8       4       4       4       4       4       4       4       4
    9-   16       4       4       4       4       4       4       4       4
   17-   24       4       4       4       4       4       4       4       4
   25-   32       4       4       4       4       4       4       4       4
   33-   40       4       4       4       4       4       4       4       4
   41-   48       4       4       4       4       4       4       4       4
.....................................................................................
40393-40400       4       4       4       4       4       4       4       4
40401-40406       4       4       4       4       4       4

  BOT BOX #     (VFN(F,SB), F=1,NVF(SB)
  161625  395336  395337  395338  395339
  161626  395340  395341  395342  395343
  161627  395344  395345  395346  395347
  161628  395348  395349  395350  395351
```

```
161629  395352  395353  395354  395355
161630  395356  395357  395358  395359
```

Appendix E. Example water column information file ("wqmgeo.inp")

```
C: GEO input file for ICM
C: SCK, 09-May-2007
   BOX #  B#_K+1

       1       0
       2       0
       3       0
       4       0
       5       0
       6       0
       7       0
       8       0
       9       0
      10       0
      11       0
      12       0
      13       0
      14       0
      15       0
      16       0
      17       0
      18       0
      19       0
.....
   40390  202014
   40391  202015
   40392  202016
   40393  202017
   40394  202018
   40395  202019
   40396  202020
   40397  202021
   40398  202022
   40399  202023
   40400  202024
   40401  202025
   40402  202026
   40403  202027
   40404  202028
   40405  202029
   40406  202030
```

Appendix F. Example for boundary face information

```
   50 14103 404  51  5 Ocean
   51 14474 404  52  5 Ocean
   52 14847 404  53  5 Ocean
   53 15229 404  54  5 Ocean
   54 15615 404  55  5 Ocean
   55 15999 404  56  5 Ocean
   56 16379 404  57  5 Ocean
   57 16749 404  58  5 Ocean
   58 17120 404  59  5 Ocean
   59 17495 404  60  5 Ocean
   60 17866 404  61  5 Ocean
   61 18238 404  62  5 Ocean
   62 18609 404  63  5 Ocean
   63 18979 404  64  5 Ocean
   64 19349 404  65  5 Ocean
   65 19719 404  66  5 Ocean
   66 20087 404  67  5 Ocean
   67 20454 404  68  5 Ocean
   68 39335 204 165  5 Wolf_River
   69 39373 204 166  5
   70 39445 258 168  5 Biloxi_River
   71 39475 258 169  5
   72 39502 258 170  5
   73 39664   6  25  5 Pont_River
   74 39730   7  25  5
   75 39796   8  25  5
   76 39862   9  25  5
   77 39928  10  25  5
   78 40135  13  91  5 Tickfaw_River
   79 40230  15  91  5 Tangipahoa_River
   80 40466  20  91  5 Tchekfuncta_River
   81 45391 126 163  5 Pearl_River
   82 45521 127 163  5
   83 46012 135   2  5 Ocean
   84 46084 136   2  5 Ocean
   85 46156 137   2  5 Ocean
   86 46226 138   2  5 Ocean
   87 46289 139   2  5 Ocean
   88 46346 140   2  5 Ocean
   89 46400 141   2  5 Ocean
   90 46454 142   2  5 Ocean
Pont_River
   25
      73       74       75       76       77      432      433      434
     435      436      791      792      793      794      795     1150
    1151     1152     1153     1154     1509     1510     1511     1512
    1513
Tickfaw_River
    5
      78      437      796     1155     1514
Tangipahoa_River
```

```
    5
      79     438     797    1156    1515
Tchekfuncta_River
    5
      80     439     798    1157    1516
Pearl_River
   10
      81      82     440     441     799     800    1158    1159
    1517    1518
Jordan_River
    5
     126     485     844    1203    1562
Wolf_River
   10
      68      69     427     428     786     787    1145    1146
    1504    1505
Biloxi_River
   15
      70      71      72     429     430     431     788     789
     790    1147    1148    1149    1506    1507    1508
Pascagula_River
    5
     270     629     988    1347    1706
Pascagula_River
    5
     279     638     997    1356    1715
Mobile_Rivers
   25
     338     340     342     344     346     697     699     701
     703     705    1056    1058    1060    1062    1064    1415
    1417    1419    1421    1423    1774    1776    1778    1780
    1782
Ocean
 1680
       1       2       3       4       5       6       7       8
       9      10      11      12      13      14      15      16
      17      18      19      20      21      22      23      24
```

Appendix G. Program “read_ch3d_grid_sigma.m”

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Matlab program to create ICM linkage files
%   from CH3D-Sigma
%
%   Input files:
%       file15 - grid data
%       file50 - depth data
%       file4 - CH3D block input control file
%       filem - CH3D main input control file
%   Output files:
%       file94 - cell info (used by CH3D)
%       file95 - face info (used by CH3D and ICM)
%       filegeo - cell layout (used by ICM)
%       filecol - cell layeout (used in ICM Postprocessing)
%       filebnd - boundary face data (used by ICM Preprocessing)
%       filegis - input to ArcView ET
%
%   May 2007
%   S. Kim
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%-----------------------------------------------------------------
%
%
%  read in CH3D input control file
%
[file4, pathname]=uigetfile('*.*','CH3D Block Data');
fid4=fopen([pathname file4],'rt');
%fid4=fopen('blk01.inp','rt');
%
%  get CH3D grid setup
%
tline=fgets(fid4);
itest=1;
while (itest>0)
    hdrt=fscanf(fid4,'%s',1);
    icell=str2num(hdrt);
    if(~isempty(icell))
        break;
    end;
end;
jcell=fscanf(fid4,'%g',1);
kcell=fscanf(fid4,'%g',1);
kmax=kcell;
imax=icell+1;
jmax=jcell+1;
%
% set zero for boxes
%
bexist=zeros(imax,jmax,kmax);
```

```
boxnum=bexist;
obox=zeros(imax-1,jmax-1);
%
% set zero for default boundary face
%
river=zeros(imax,jmax,2);
bar=river;
ocean=river;
rbnd=river;
%
%  river boundaries
%
nriver=0;
itest=1;
while (itest>0)
    hdrt=fscanf(fid4,'%s',1);
    nc=str2num(hdrt);
    if(isempty(nc))
        if(strncmp(hdrt,'NRIVER',6))
            break;
        end;
    end;
end;
nriver=fscanf(fid4,'%g',1);
if(nriver>0)
    tline=fgets(fid4);
    tline=fgets(fid4);
    river=zeros(imax,jmax);
    for ir=1:nriver
        tline=fgets(fid4);
        rinfo=sscanf(tline,'%g %g %g %g',[1,4]);
        ttext=strread(tline,'%s');
        [m n]=size(ttext);
        switch rinfo(1)
            case 1
                i=rinfo(2);
                j=rinfo(3);
                je=rinfo(4);
                river(i,j:je,1)=1;
                rtext(i,j:je,1)=strcat(ttext(m-1),'_',ttext(m));
                rbnd(i,j:je,1)=ir;
                nrtext(ir)=rtext(i,j,1);
            case 2
                i=rinfo(3);
                j=rinfo(2);
                je=rinfo(4);
                river(i:je,j,2)=2;
                rtext(i:je,j,2)=strcat(ttext(m-1),'_',ttext(m));
                rbnd(i:je,j,2)=ir;
                nrtext(ir)=rtext(i,j,2);
            case 3
                i=rinfo(2)+1;
                j=rinfo(3);
                je=rinfo(4);
                river(i,j:je,1)=1;
                rtext(i,j:je,1)=strcat(ttext(m-1),'_',ttext(m));
                rbnd(i,j:je,1)=ir;
                nrtext(ir)=rtext(i,j,1);
            case 4
                i=rinfo(3);
                j=rinfo(2)+1;
```

```
                je=rinfo(4);
                river(i:je,j,2)=2;
                rtext(i:je,j,2)=strcat(ttext(m-1),'_',ttext(m));
                rbnd(i:je,j,2)=ir;
                nrtext(ir)=rtext(i,j,2);
        end;
    end;
end;
ir=nriver+1;
nrtext{ir}='Ocean';
%
%  bar
%
itest=1;
while (itest>0)
    hdrt=fscanf(fid4,'%s',1);
    nc=str2num(hdrt);
    if(isempty(nc))
        if(strncmp(hdrt,'NBAR',4))
            break;
        end;
    end;
end;
tline=fgets(fid4);
nbar=fscanf(fid4,'%g',1);
if(nbar>0)
    tline=fgets(fid4);
    tline=fgets(fid4);
    for ibar=1:nbar
        tline=fgets(fid4);
        barinfo=sscanf(tline,'%g %g %g %g',[1,4]);
        switch barinfo(1)
            case 1
                j=barinfo(2);
                i1=barinfo(3);
                i2=barinfo(4);
                bar(i1:i2,j,2)=1;
            case 2
                j1=rinfo(3);
                j2=rinfo(4);
                i=rinfo(2);
                bar(i,j1:i2,1)=2;
        end;
    end;
end;
%
%  ocean boundaries
%
itest=1;
while (itest>0)
    hdrt=fscanf(fid4,'%s',1);
    nc=str2num(hdrt);
    if(isempty(nc))
        if(strncmp(hdrt,'TIDFN2',6))
            itest=2;
            break;
        end;
    end;
end;
if(itest==2)
    tline=fgets(fid4);
```

```
    ioc=0;
    while (itest>0)
        tline=fgets(fid4);
        i1=str2num(tline(1:8));
        if(isempty(i1))
            break;
        end;
        ioc=ioc+1;
        i2=str2num(tline(9:16));
        i3=str2num(tline(17:24));
        i4=str2num(tline(25:32));
        switch i1
            case 1
                ocean(i2+1,i3:i4,1)=1;
                obox(i2,i3:i4)=1;
                for j=i3:i4
                    rtext{i1+1,j,1}='Ocean';
                end;
                rbnd(i2+1,i3:i4,1)=ir;
            case 2
                ocean(i3:i4,i2+1,2)=1;
                obox(i3:i4,i2)=1;
                for j=i3:i4
                    rtext{j,i2+1,2}='Ocean';
                end;
                rbnd(i3:i4,i2+1,2)=ir;
            case 3
                ocean(i2,i3:i4,1)=1;
                obox(i2,i3:i4)=1;
                for j=i3:i4
                    rtext{i2,j,1}='Ocean';
                end;
                rbnd(i2,i3:i4,1)=ir;
            case 4
                ocean(i3:i4,i2,2)=1;
                obox(i3:i4,i2)=1;
                for j=i3:i4
                    rtext{j,i2,2}='Ocean';
                end;
                rbnd(i3:i4,i2,2)=ir;
        end;
    end;
end;
fclose(fid4);
%
%-----------------------------------------------------------------
%
%
%  read in depth data
%
[file50, pathname]=uigetfile('*.*','Depth Data File');
fid_50=fopen([pathname file50],'rt');
%fid_50=fopen('fort.23','rt');
X=fscanf(fid_50,'%g');
depth=reshape(X,[imax-1 jmax-1]);
depth(imax,1:jmax)=0;
depth(1:imax,jmax)=0;
clear X;
fclose(fid_50);
nlayer=kmax*(depth./depth);
km=1;
```

```
nbox=nansum(nansum(nlayer));
%
% plot grid
%
plot_grid_1(imax,jmax,depth,obox,river,bar);
%
%-----------------------------------------------------------------
%
%
%  read in CH3D main input control file
%
[filem, pathname]=uigetfile('*.*','CH3D Main Input');
fidm=fopen([pathname filem],'rt');
%fidm=fopen('main.inp','rt');
%
%
%
tline=fgets(fidm);
tline=fgets(fidm);
itest=1;
while (itest>0)
    hdrt=fscanf(fidm,'%s',1);
    dt=str2num(hdrt);
    if(~isempty(dt))
        break;
    end;
end;
while (itest>0)
    hdrt=fscanf(fidm,'%s',1);
    it1=str2num(hdrt);
    if(~isempty(it1))
        break;
    end;
end;
it2=fscanf(fidm,'%g',1);
istart=fscanf(fidm,'%g',1);
itsalt=fscanf(fidm,'%g',1);
itsalt=max(itsalt,1);
ihot=fscanf(fidm,'%g',1);
iadi=fscanf(fidm,'%g',1);
fclose(fidm);
%
% find boxes
%
%
% find boxes
%
%ibox=zeros(nbox);
%jbox=ibox;
%kbox=ibox;
bexist=zeros(imax,jmax);
nb=0;
k=kmax;
for j=1:jmax-1
    for i=1:imax-1
        if(depth(i,j)>0)&&(obox(i,j)==0)
            nb=nb+1;
            ibox(nb)=i;
            jbox(nb)=j;
            kbox(nb)=k;
            bexist(i,j)=1;
```

```
        end;
    end;
end;
nsb=nb;
for k=kmax-1:-1:1
    ibeg=nb+1;
    iend=ibeg+nsb-1;
    ibox(ibeg:iend)=ibox(1:nsb);
    jbox(ibeg:iend)=jbox(1:nsb);
    kbox(ibeg:iend)=k;
    nb=iend;
end;
boxnum=zeros(imax,jmax,kmax);
for ib=1:nb
    boxnum(ibox(ib),jbox(ib),kbox(ib))=ib;
end;
%
%-----------------------------------------------------------------
%
%
%  write to file94
%
[file94, pathname]=uiputfile('*.94','Cell Info File');
fid94=fopen([pathname file94],'wt');
%fid94=fopen('fort.94','wt');
%
%
%
fprintf(fid94,'File 94: box info for CH3D\n');
fprintf(fid94,'SCK\n');
fprintf(fid94,date);
fprintf(fid94,'\n');
%
%
%
fprintf(fid94,'     NSB    NAVG   ITWQS    TBOX\n');
navg=3600/dt;
fprintf(fid94,'%8d',nsb);
fprintf(fid94,'%8d',navg);
fprintf(fid94,'%8d',itsalt);
fprintf(fid94,'%8d\n',nb);
%
%
%
fprintf(fid94,'  BOX_NO  IFIRST   ILAST  JFIRST   JLAST       K\n');
for i=1:nb
    fprintf(fid94,'%8d',i);
    fprintf(fid94,'%8d',ibox(i));
    fprintf(fid94,'%8d',ibox(i)+1);
    fprintf(fid94,'%8d',jbox(i));
    fprintf(fid94,'%8d',jbox(i)+1);
    fprintf(fid94,'%8d\n',kbox(i));
end;
fclose(fid94);
%
%-----------------------------------------------------------------
%
%
%  find faces
%
nhqft=0;
```

```
bndfn=0;
k=kmax;
%
%  x-sweep
%
for j=1:jmax-1
    for i=1:imax
        if(bexist(i,j)>0)
            if(i>1)
                if(bexist(i-1,j)>0)
                    if(bar(i,j,1)==0)
                        nhqft=nhqft+1;
                        qd(nhqft)=1;
                        ib(nhqft)=boxnum(i-1,j,k);
                        jb(nhqft)=boxnum(i,j,k);
                        jrb(nhqft)=boxnum(i+1,j,k);
                        if(i>2)
                            ilb(nhqft)=boxnum(i-2,j,k);
                        else
                            ilb(nhqft)=0;
                        end;
                        iface(nhqft)=i;
                        jface(nhqft)=j;
                        kface(nhqft)=k;
                    end;
                elseif(river(i,j,1)>0||ocean(i,j,1)>0)
                    nhqft=nhqft+1;
                    qd(nhqft)=1;
                    ib(nhqft)=boxnum(i-1,j,k);
                    jb(nhqft)=boxnum(i,j,k);
                    jrb(nhqft)=boxnum(i+1,j,k);
                    if(i>2)
                        ilb(nhqft)=boxnum(i-2,j,k);
                    else
                        ilb(nhqft)=0;
                    end;
                    iface(nhqft)=i;
                    jface(nhqft)=j;
                    kface(nhqft)=k;
                    bndfn=bndfn+1;
                    bndfce(bndfn)=nhqft;
                    bndtxt(bndfn)=rtext(i,j,1);
                    bndid(bndfn)=rbnd(i,j,1);
                end;
            elseif(river(i,j,1)>0||ocean(i,j,1)>0)
                nhqft=nhqft+1;
                qd(nhqft)=1;
                ib(nhqft)=0;
                ilb(nhqft)=0;
                jb(nhqft)=boxnum(i,j,k);
                jrb(nhqft)=boxnum(i+1,j,k);
                iface(nhqft)=i;
                jface(nhqft)=j;
                kface(nhqft)=k;
                bndfn=bndfn+1;
                bndfce(bndfn)=nhqft;
                bndtxt(bndfn)=rtext(i,j,1);
                bndid(bndfn)=rbnd(i,j,1);
            end;
        elseif(river(i,j,1)>0||ocean(i,j,1)>0)
            nhqft=nhqft+1;
```

```
            qd(nhqft)=1;
            ib(nhqft)=boxnum(i-1,j,k);
            if(i==2)
                ilb(nhqft)=0;
            else
                ilb(nhqft)=boxnum(i-2,j,k);
            end;
            jb(nhqft)=0;
            jrb(nhqft)=0;
            iface(nhqft)=i;
            jface(nhqft)=j;
            kface(nhqft)=k;
            bndfn=bndfn+1;
            bndfce(bndfn)=nhqft;
            bndtxt(bndfn)=rtext(i,j,1);
            bndid(bndfn)=rbnd(i,j,1);
        end;
    end;
end;
%
%  y-sweep
%
for i=1:imax-1
    for j=1:jmax
        if(bexist(i,j)>0)
            if(j>1)
                if(bexist(i,j-1)>0)
                    if(bar(i,j,2)==0)
                        nhqft=nhqft+1;
                        qd(nhqft)=2;
                        ib(nhqft)=boxnum(i,j-1,k);
                        jb(nhqft)=boxnum(i,j,k);
                        jrb(nhqft)=boxnum(i,j+1,k);
                        if(j>2)
                            ilb(nhqft)=boxnum(i,j-2,k);
                        else
                            ilb(nhqft)=0;
                        end;
                        iface(nhqft)=i;
                        jface(nhqft)=j;
                        kface(nhqft)=k;
                    end;
                elseif(river(i,j,2)>0||ocean(i,j,2)>0)
                    nhqft=nhqft+1;
                    qd(nhqft)=2;
                    ib(nhqft)=boxnum(i,j-1,k);
                    jb(nhqft)=boxnum(i,j,k);
                    jrb(nhqft)=boxnum(i,j+1,k);
                    if(j>2)
                        ilb(nhqft)=boxnum(i,j-2,k);
                    else
                        ilb(nhqft)=0;
                    end;
                    iface(nhqft)=i;
                    jface(nhqft)=j;
                    kface(nhqft)=k;
                    bndfn=bndfn+1;
                    bndfce(bndfn)=nhqft;
                    bndtxt(bndfn)=rtext(i,j,2);
                    bndid(bndfn)=rbnd(i,j,2);
                end;
```

```
            elseif(river(i,j,2)>0||ocean(i,j,2)>0)
                nhqft=nhqft+1;
                qd(nhqft)=2;
                ib(nhqft)=0;
                ilb(nhqft)=0;
                jb(nhqft)=boxnum(i,j,k);
                jrb(nhqft)=boxnum(i,j+1,k);
                iface(nhqft)=i;
                jface(nhqft)=j;
                kface(nhqft)=k;
                bndfn=bndfn+1;
                bndfce(bndfn)=nhqft;
                bndtxt(bndfn)=rtext(i,j,2);
                bndid(bndfn)=rbnd(i,j,2);
            end;
        elseif(river(i,j,2)>0||ocean(i,j,2)>0)
            nhqft=nhqft+1;
            qd(nhqft)=2;
            ib(nhqft)=boxnum(i,j-1,k);
            if(j==2)
                ilb(nhqft)=0;
            else
                ilb(nhqft)=boxnum(i,j-2,k);
            end;
            jb(nhqft)=0;
            jrb(nhqft)=0;
            iface(nhqft)=i;
            jface(nhqft)=j;
            kface(nhqft)=k;
            bndfn=bndfn+1;
            bndfce(bndfn)=nhqft;
            bndtxt(bndfn)=rtext(i,j,2);
            bndid(bndfn)=rbnd(i,j,2);
        end;
    end;
end;
bndsfn=bndfn;
nhqf=nhqft;
for k=kmax-1:-1:1
    ibeg=nhqft+1;
    iend=ibeg+nhqf-1;
    qd(ibeg:iend)=qd(1:nhqf);
    ilb(ibeg:iend)=ilb(1:nhqf)+nsb*(kmax-k);
    ib(ibeg:iend)=ib(1:nhqf)+nsb*(kmax-k);
    jb(ibeg:iend)=jb(1:nhqf)+nsb*(kmax-k);
    jrb(ibeg:iend)=jrb(1:nhqf)+nsb*(kmax-k);
    iface(ibeg:iend)=iface(1:nhqf);
    jface(ibeg:iend)=jface(1:nhqf);
    kface(ibeg:iend)=k;
    nhqft=iend;
    jbeg=bndfn+1;
    jend=jbeg+bndsfn-1;
    bndfce(jbeg:jend)=bndfce(1:bndsfn)+nhqf*(kmax-k);
    bndtxt(jbeg:jend)=bndtxt(1:bndsfn);
    bndid(jbeg:jend)=bndid(1:bndsfn);
    bndfn=jend;
end;
nqf=nhqft+nsb*(kmax-1);
%
%-----------------------------------------------------------------
%
```

```
%
%  write to file95
%
[file95, pathname]=uiputfile('*.95','Map Info File');
fid95=fopen([pathname file95],'wt');
%fid95=fopen('fort.95','wt');
%
%
%
fprintf(fid95,'File 95: Face info for CH3D and ICM\n');
fprintf(fid95,'SCK\n');
fprintf(fid95,date);
fprintf(fid95,'\n');
fprintf(fid95,':\n');
fprintf(fid95,':\n');
%
%
%
fprintf(fid95,'   NHQFT     NQF    NHQF\n');
fprintf(fid95,'%8d',nhqft);
fprintf(fid95,'%8d',nqf);
fprintf(fid95,'%8d\n',nhqf);
%
%
%
fprintf(fid95,'       F      QD     ILB      IB      JB     JRB');
fprintf(fid95,'      KP      KF      KL   LAYER\n');
for i=1:nhqft
    fprintf(fid95,'%8d',i);
    fprintf(fid95,'%8d',qd(i));
    fprintf(fid95,'%8d',ilb(i));
    fprintf(fid95,'%8d',ib(i));
    fprintf(fid95,'%8d',jb(i));
    fprintf(fid95,'%8d',jrb(i));
    switch qd(i)
        case 1
            if(jb(i)==0)
                inum=ibox(ib(i))+1;
                jnum=jbox(ib(i));
                knum=kbox(ib(i));
            else
                inum=ibox(jb(i));
                jnum=jbox(jb(i));
                knum=kbox(jb(i));
            end;
            fprintf(fid95,'%8d',inum);
            fprintf(fid95,'%8d',jnum);
            fprintf(fid95,'%8d',jnum);
            fprintf(fid95,'%8d',knum);
            fprintf(fid95,'\n');
        case 2
            if(jb(i)==0)
                inum=ibox(ib(i));
                jnum=jbox(ib(i))+1;
                knum=kbox(ib(i));
            else
                inum=ibox(jb(i));
                jnum=jbox(jb(i));
                knum=kbox(jb(i));
            end;
            fprintf(fid95,'%8d',jnum);
```

```
            fprintf(fid95,'%8d',inum);
            fprintf(fid95,'%8d',inum);
            fprintf(fid95,'%8d',knum);
            fprintf(fid95,'\n');
    end;
end;
i=nhqft;
for ibx=1:nsb
    for k=2:kmax
        i=i+1;
        vfn(ibx,k)=i;
        fprintf(fid95,'%8d',i);
        fprintf(fid95,'%8d',3);
        if(k==2)
            ib1=0;
        else
            ib1=(kmax-k+2)*nsb+ibx;
        end;
        ib2=(kmax-k+1)*nsb+ibx;
        ib3=(kmax-k)*nsb+ibx;
        if(k==kmax)
            ib4=0;
        else
            ib4=(kmax-k-1)*nsb+ibx;
        end;
        fprintf(fid95,'%8d',ib1);
        fprintf(fid95,'%8d',ib2);
        fprintf(fid95,'%8d',ib3);
        fprintf(fid95,'%8d',ib4);
        fprintf(fid95,'%8d',jbox(ibx));
        fprintf(fid95,'%8d',ibox(ibx));
        fprintf(fid95,'%8d',ibox(ibx));
        fprintf(fid95,'%8d',k-1);
        fprintf(fid95,'%8d',k);
        fprintf(fid95,'\n');
    end;
end;
%
%
%
nvf=kmax-1;
fprintf(fid95,'\n');
fprintf(fid95,'  SFC BOX #     (NVF(SB), SB=1,NSB)\n');
nline=floor(nsb/8);
nextra=mod(nsb,8);
for il=1:nline
    ibeg=(il-1)*8+1;
    iend=ibeg+7;
    fprintf(fid95,'%5d',ibeg);
    fprintf(fid95,'-');
    fprintf(fid95,'%5d',iend);
    for k=ibeg:iend
        fprintf(fid95,'%8d',nvf);
    end;
    fprintf(fid95,'\n');
end;
if (nextra>0)
    ibeg=iend+1;
    iend=ibeg+nextra-1;
    fprintf(fid95,'%5d',ibeg);
    fprintf(fid95,'-');
```

```
    fprintf(fid95,'%5d',iend);
    for k=ibeg:iend
        fprintf(fid95,'%8d',nvf);
    end;
    fprintf(fid95,'\n');
end;
%
%
%
fprintf(fid95,'\n');
fprintf(fid95,'  BOT BOX #     (VFN(F,SB), F=1,NVF(SB)\n');
bbstart=nsb*(kmax-1);
for i=1:nsb
    fprintf(fid95,'%8d',bbstart+i);
    iprcnt=0;
    for k=nvf:-1:1
        iprcnt=iprcnt+1;
        if(iprcnt>1&&mod(iprcnt,9)==1)
            fprintf(fid95,'\n        ');
        end;
        fprintf(fid95,'%8d',vfn(i,kmax-k+1));
    end;
    fprintf(fid95,'\n');
end;
%
%
%
fclose(fid95);
%
%----------------------------------------------------------------
%
%
%  write to filegeo
%
[filegeo, pathname]=uiputfile('wqmgeo.*','Geo File for ICM');
fidgeo=fopen([pathname filegeo],'wt');
%fidgeo=fopen('wqmgeo.inp','wt');
%
%
%
fprintf(fidgeo,'C: GEO input file for ICM\n');
fprintf(fidgeo,'C: SCK, ');
fprintf(fidgeo,date);
fprintf(fidgeo,'\n');
fprintf(fidgeo,'   BOX #  B#_K+1\n');
fprintf(fidgeo,'\n');
for i=1:nsb
    fprintf(fidgeo,'%8d%8d\n',i,0);
end;
for i=nsb+1:nb
    fprintf(fidgeo,'%8d%8d\n',i,...
        boxnum(ibox(i),jbox(i),kbox(i)+1));
end;
fprintf(fidgeo,'\n');
fprintf(fidgeo,'    SBOX    BBOX\n');
for i=1:nsb
    inum=ibox(i);
    jnum=jbox(i);
    knum=km;
    fprintf(fidgeo,'%8d%8d\n',i,boxnum(inum,jnum,knum));
end;
```

```
fclose(fidgeo);
%
%-----------------------------------------------------------------
%
%
%  write to filecol
%
[filecol, pathname]=uiputfile('wqmcol.*','Postprocessing Column File');
fidcol=fopen([pathname filecol],'wt');
%fidcol=fopen('wqmcol.inp','wt');
%
%
%
for i=1:nsb
    inum=ibox(i);
    jnum=jbox(i);
    fprintf(fidcol,'%3d %3d',inum,jnum);
    fprintf(fidcol,' %2d',nlayer(inum,jnum));
    for k=kmax:-1:km
        fprintf(fidcol,'%7d',boxnum(inum,jnum,k));
    end;
    fprintf(fidcol,'\n');
end;
fclose(fidcol);
%
%-----------------------------------------------------------------
%
%
%  write to filebnd
%
[filebnd, pathname]=uiputfile('bndface.*','Boundary Face File');
fidbnd=fopen([pathname filebnd],'wt');
%fidbnd=fopen('bndface.inp','wt');
%
%
%
ncbnd=zeros(ir,1);
for i=1:bndfn
    fnum=bndfce(i);
    fprintf(fidbnd,'%5d %5d %3d %3d %2d',i,fnum,...
        iface(fnum),jface(fnum),kface(fnum));
    fprintf(fidbnd,' %s\n',bndtxt{i});
    j=bndid(i);
    ncbnd(j)=ncbnd(j)+1;
    k=ncbnd(j);
    bndfid(j,k)=i;
end;
for j=1:ir
    fprintf(fidbnd,'%s\n',nrtext{j});
    fprintf(fidbnd,'%5d\n',ncbnd(j));
    nline=floor(ncbnd(j)/8);
    nextra=mod(ncbnd(j),8);
    ibeg=0;
    iend=0;
    if(nline>0)
        for il=1:nline
            ibeg=(il-1)*8+1;
            iend=ibeg+7;
            for k=ibeg:iend
                fprintf(fidbnd,'%8d',bndfid(j,k));
            end;
```

```
            fprintf(fidbnd,'\n');
        end;
    end;
    if (nextra>0)
        ibeg=iend+1;
        iend=ibeg+nextra-1;
        for k=ibeg:iend
            fprintf(fidbnd,'%8d',bndfid(j,k));
        end;
        fprintf(fidbnd,'\n');
    end;
end;
fclose(fidbnd);
%
%
%
%
%-----------------------------------------------------------------
%
%
% read in grid data
%
[file15, pathname]=uigetfile('*.*','Grid File');
fid_15=fopen([pathname file15],'rt');
%fid_15=fopen('grid.inp','rt');
%
%
%
tline=fgets(fid_15);
itest=1;
while (itest>0)
    hdrt=fscanf(fid_15,'%s',1);
    imax=str2num(hdrt);
    if(~isempty(imax))
        break;
    end;
end;
jmax=fscanf(fid_15,'%g',1);
tline=fgets(fid_15);
for j=1:jmax
    for i=1:imax
        tline=fgets(fid_15);
        X=sscanf(tline,'%g %g',[1,2]);
        x(i,j)=X(1);
        y(i,j)=X(2);
    end;
end;
clear X;
fclose(fid_15);
%
%-----------------------------------------------------------------
%
%
%  write to filebnd
%
[filegis, pathname]=uiputfile('CH3D_ICM_GIS.*','Boundary Face File');
fidgis=fopen([pathname filegis],'wt');
%fidgis=fopen('CH3D_ICM_GIS.txt','wt');
%
%
%
```

```
fprintf(fidgis,'%s,','CH3D');
fprintf(fidgis,'%s','DEPTH');
for k=1:kmax
    fprintf(fidgis,',%s',strcat('L',num2str(k)));
end;
fprintf(fidgis,'\n');
%hold on;
for ib=1:nsb
    i=ibox(ib);
    j=jbox(ib);
    ij=i*1000+j;
    fprintf(fidgis,'%6.6d,',ij);
    fprintf(fidgis,'%g',depth(i,j));
    for k=kmax:-1:1
        fprintf(fidgis,',%g',boxnum(i,j,k));
    end;
    fprintf(fidgis,'\n');
    xx(1)=x(i,j);yy(1)=y(i,j);
    xx(2)=x(i+1,j);yy(2)=y(i+1,j);
    xx(3)=x(i+1,j+1);yy(3)=y(i+1,j+1);
    xx(4)=x(i,j+1);yy(4)=y(i,j+1);
    xx(5)=xx(1);yy(5)=yy(1);
    for l=1:5
        fprintf(fidgis,'%g,%g\n',xx(l),yy(l));
    end;
    fprintf(fidgis,'%s\n','END');
end;
fprintf(fidgis,'END\n');
fclose(fidgis);
```

Appendix H. Program “read_ch3d_z.m”

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Matlab program to create ICM linkage files
%   from CH3D-Z
%
%   Input files:
%       file15 - grid data
%       file50 - depth data
%       file4 - CH3D input control file
%   Output files:
%       file94 - cell info (used by CH3D)
%       file95 - face info (used by CH3D and ICM)
%       filegeo - cell layout (used by ICM)
%       filecol - cell layeout (used in ICM Postprocessing)
%       filebnd - boundary face data (used by ICM Preprocessing)
%       filegis - input to ArcView ET
%
%   May 2007
%   S. Kim
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clear;
%
%
%  read in CH3D input control file
%
[file4, pathname]=uigetfile('*.*','CH3D Input Control File');
fid4=fopen([pathname file4],'rt');
%
%  get CH3D run info
%
tline=fgets(fid4);
tline=fgets(fid4);
itest=1;
while (itest>0)
    hdrt=fscanf(fid4,'%s',1);
    it1=str2num(hdrt);
    if(~isempty(it1))
        break;
    end;
end;
it2=fscanf(fid4,'%g',1);
dt=fscanf(fid4,'%g',1);
istart=fscanf(fid4,'%g',1);
itest=fscanf(fid4,'%g',1);
itsalt=fscanf(fid4,'%g',1);
littorl=fscanf(fid4,'%g',1);
laterl=fscanf(fid4,'%g',1);
%
%  get CH3D grid setup
%
```

```
itest=1;
while (itest>0)
    hdrt=fscanf(fid4,'%s',1);
    nc=str2num(hdrt);
    if(isempty(nc))
        if(strncmp(hdrt,'KD',2))
            break;
        end;
    end;
end;
ipa=fscanf(fid4,'%g',1);
ipb=fscanf(fid4,'%g',1);
id=fscanf(fid4,'%g',1);
jpa=fscanf(fid4,'%g',1);
jpb=fscanf(fid4,'%g',1);
jd=fscanf(fid4,'%g',1);
kpa=fscanf(fid4,'%g',1);
kpb=fscanf(fid4,'%g',1);
kd=fscanf(fid4,'%g',1);
%
% set grid dimension
%
imax=ipb+1;
jmax=jpb+1;
kmax=kpb;
%
% set zero for boxes
%
bexist=zeros(imax,jmax,kmax);
boxnum=bexist;
obox=zeros(imax-1,jmax-1);
%
% set zero for default boundary face
%
river=zeros(imax,jmax,2);
bar=river;
ocean=river;
%
%  read in deltaz
%
itest=1;
while (itest>0)
    hdrt=fscanf(fid4,'%s',1);
    nc=str2num(hdrt);
    if(isempty(nc))
        if(strncmp(hdrt,'DELTAZM',7))
            break;
        end;
    end;
end;
deltaz=fscanf(fid4,'%g',1);
%
%  read in map scale
%
itest=1;
while (itest>0)
    hdrt=fscanf(fid4,'%s',1);
    nc=str2num(hdrt);
    if(isempty(nc))
        if(strncmp(hdrt,'YMAP',4))
            break;
```

```
        end;
    end;
end;
xscale=fscanf(fid4,'%g',1);
deltaz=int32(deltaz/xscale);
xscale=.01*xscale;
%
%  check river boundaries
%
itest=1;
while (itest>0)
    hdrt=fscanf(fid4,'%s',1);
    nc=str2num(hdrt);
    if(isempty(nc))
        if(strncmp(hdrt,'NRIVER',6))
            break;
        end;
    end;
end;
nriver=fscanf(fid4,'%g',1);
if(nriver>0)
    tline=fgets(fid4);
    tline=fgets(fid4);
    for ir=1:nriver
        tline=fgets(fid4);
        rinfo=sscanf(tline,'%g %g %g %g',[1,4]);
        ttext=strread(tline,'%s');
        switch rinfo(1)
            case 1
                i=rinfo(2);
                j1=rinfo(3);
                j2=rinfo(4);
                river(i,j1:j2,1)=1;
                [m n]=size(ttext);
                rtext(i,j1:j2,1:2)=strcat(ttext(m-1),'_',ttext(m));
                rbnd(i,j1:j2,1:2)=ir;
                nrtext(ir)=rtext(i,j1,1);
            case 2
                i1=rinfo(3);
                i2=rinfo(4);
                j=rinfo(2);
                river(i1:i2,j,2)=2;
                [m n]=size(ttext);
                rtext(i1:i2,j,1:2)=strcat(ttext(m-1),'_',ttext(m));
                rbnd(i1:i2,j,1:2)=ir;
                nrtext(ir)=rtext(i1,j,1);
            case 3
                i=rinfo(2)+1;
                j1=rinfo(3);
                j2=rinfo(4);
                river(i,j1:j2,1)=1;
                [m n]=size(ttext);
                rtext(i,j1:j2,1:2)=strcat(ttext(m-1),'_',ttext(m));
                rbnd(i,j1:j2,1:2)=ir;
                nrtext(ir)=rtext(i,j1,1);
            case 4
                i1=rinfo(3);
                i2=rinfo(4);
                j=rinfo(2)+1;
                river(i1:i2,j,2)=2;
                [m n]=size(ttext);
```

```
                rtext(i1:i2,j,1:2)=strcat(ttext(m-1),'_',ttext(m));
                rbnd(i1:i2,j,1:2)=ir;
                nrtext(ir)=rtext(i1,j,1);
        end;
    end;
end;
%ir=nriver+1;
%nrtext{ir}='Ocean';
%
%  bar
%
itest=1;
while (itest>0)
    hdrt=fscanf(fid4,'%s',1);
    nc=str2num(hdrt);
    if(isempty(nc))
        if(strncmp(hdrt,'NBAR',4))
            break;
        end;
    end;
end;
nbar=fscanf(fid4,'%g',1);
if(nbar>0)
    tline=fgets(fid4);
    tline=fgets(fid4);
    for ibar=1:nbar
        tline=fgets(fid4);
        barinfo=sscanf(tline,'%g %g %g %g',[1,4]);
        switch barinfo(1)
            case 1
                j=barinfo(2);
                i1=barinfo(3);
                i2=barinfo(4);
                bar(i1:i2,j,2)=1;
            case 2
                j1=rinfo(3);
                j2=rinfo(4);
                i=rinfo(2);
                bar(i,j1:i2,1)=2;
        end;
    end;
end;
%
%  ocean boundaries
%
ir=nriver;
itest=1;
while (itest>0)
    hdrt=fscanf(fid4,'%s',1);
    nc=str2num(hdrt);
    if(isempty(nc))
        if(strncmp(hdrt,'TIDFN2',6))
            break;
        end;
    end;
end;
tline=fgets(fid4);
ioc=0;
while (itest>0)
    tline=fgets(fid4);
    i1=str2num(tline(1:8));
```

```
    if(isempty(i1))
        break;
    end;
    ioc=ioc+1;
    ir=ir+1;
    nrtext{ir}=strcat('Ocean',num2str(ioc));
    i2=str2num(tline(9:16));
    i3=str2num(tline(17:24));
    i4=str2num(tline(25:32));
    switch i1
        case 1
            ocean(i2+1,i3:i4,1)=1;
            obox(i2,i3:i4)=1;
            for j=i3:i4
                rtext{i2+1,j,1}=strcat('Ocean',num2str(ioc));
            end;
            rbnd(i2+1,i3:i4,1)=ir;
        case 2
            ocean(i3:i4,i2+1,2)=1;
            obox(i3:i4,i2)=1;
            for j=i3:i4
                rtext{j,i2+1,2}=strcat('Ocean',num2str(ioc));
            end;
            rbnd(i3:i4,i2+1,2)=ir;
        case 3
            ocean(i2,i3:i4,1)=1;
            obox(i2,i3:i4)=1;
            for j=i3:i4
                rtext{i2,j,1}=strcat('Ocean',num2str(ioc));
            end;
            rbnd(i2,i3:i4,1)=ir;
        case 4
            ocean(i3:i4,i2,2)=1;
            obox(i3:i4,i2)=1;
            for j=i3:i4
                rtext{j,i2,2}=strcat('Ocean',num2str(ioc));
            end;
            rbnd(i3:i4,i2,2)=ir;
    end;
end;
fclose(fid4);
%
%-----------------------------------------------------------------
%
%  read in depth data
%
[file50, pathname]=uigetfile('*.*','Depth Data File');
fid_50=fopen([pathname file50],'rt');
%fid_50=fopen('..\fort.50','rt');
X=fscanf(fid_50,'%g');
depth=reshape(X,[imax-1 jmax-1]);
depth(imax,1:jmax)=0;
depth(1:imax,jmax)=0;
nlayer=depth/5; % each layer is 5 ft thick
km=kmax-nlayer+1;
clear X;
fclose(fid_50);
%
%  plot grid
%
plot_grid_1(imax,jmax,depth,obox,river,bar);
```

```
%
% find boxes
%
nb=0;
for k=kmax:-1:1
    if(k==kmax)
        nsb=0;
    end;
    for j=1:jmax-1
        for i=1:imax-1
            if(depth(i,j)>0)&&(obox(i,j)==0)
                if (k>=km(i,j))
                    nb=nb+1;
                    if(k==kmax)
                        nsb=nsb+1;
                    end;
                    ibox(nb)=i;
                    jbox(nb)=j;
                    kbox(nb)=k;
                    bexist(i,j,k)=1;
                    boxnum(i,j,k)=nb;
                end;
            end;
        end;
    end;
end;
save ch3d_input imax jmax kmax deltaz xscale dt it1 it2 itsalt;
save ch3d_depth depth nlayer km;
save ch3d_bc ir obox ocean river rtext rbnd nrtext bar;
save ch3d_box nb nsb ibox jbox kbox bexist boxnum;
%
%-----------------------------------------------------------------
%
%
%  write to file94
%
[file94, pathname]=uiputfile('*.94','Cell Info File');
fid94=fopen([pathname file94],'wt');
%
%
%
fprintf(fid94,'File 94: box info for CH3D\n');
fprintf(fid94,'SCK\n');
fprintf(fid94,date);
fprintf(fid94,'\n');
%
%
%
fprintf(fid94,'     NSB    NAVG   ITWQS    TBOX\n');
navg=3600/dt;
fprintf(fid94,'%8d',nsb);
fprintf(fid94,'%8d',navg);
fprintf(fid94,'%8d',itsalt);
fprintf(fid94,'%8d\n',nb);
%
%
%
fprintf(fid94,'  BOX_NO  IFIRST   ILAST  JFIRST   JLAST       K\n');
for i=1:nb
    fprintf(fid94,'%8d',i);
    fprintf(fid94,'%8d',ibox(i));
```

```
    fprintf(fid94,'%8d',ibox(i)+1);
    fprintf(fid94,'%8d',jbox(i));
    fprintf(fid94,'%8d',jbox(i)+1);
    fprintf(fid94,'%8d\n',kbox(i));
end;
fclose(fid94);
%
%-----------------------------------------------------------------
%
%
%  find faces
%
nhqft=0;
bndfn=0;
for k=kmax:-1:1
    if(k==kmax)
        nhqf=0;
    end;
%
%  x-sweep
%
    for j=1:jmax-1
        for i=1:imax
            if(bexist(i,j,k)>0)
                if(i>1)
                    if(bexist(i-1,j,k)>0)
                        if(bar(i,j,1)==0)
                            nhqft=nhqft+1;
                            if(k==kmax)
                                nhqf=nhqf+1;
                            end;
                            qd(nhqft)=1;
                            ib(nhqft)=boxnum(i-1,j,k);
                            jb(nhqft)=boxnum(i,j,k);
                            jrb(nhqft)=boxnum(i+1,j,k);
                            if(i>2)
                                ilb(nhqft)=boxnum(i-2,j,k);
                            else
                                ilb(nhqft)=0;
                            end;
                            iface(nhqft)=i;
                            jface(nhqft)=j;
                            kface(nhqft)=k;
                        end;
                    elseif(river(i,j,1)>0||ocean(i,j,1)>0)
                        if(k>=km(i,j))
                            nhqft=nhqft+1;
                            if(k==kmax)
                                nhqf=nhqf+1;
                            end;
                            qd(nhqft)=1;
                            ib(nhqft)=boxnum(i-1,j,k);
                            jb(nhqft)=boxnum(i,j,k);
                            jrb(nhqft)=boxnum(i+1,j,k);
                            if(i>2)
                                ilb(nhqft)=boxnum(i-2,j,k);
                            else
                                ilb(nhqft)=0;
                            end;
                            iface(nhqft)=i;
                            jface(nhqft)=j;
```

```
                            kface(nhqft)=k;
                            bndfn=bndfn+1;
                            bndfce(bndfn)=nhqft;
                            bndtxt(bndfn)=rtext(i,j,1);
                            bndid(bndfn)=rbnd(i,j,1);
                        end;
                    end;
                elseif(river(i,j,1)>0||ocean(i,j,1)>0)
                    if(k>=km(i,j))
                        nhqft=nhqft+1;
                        if(k==kmax)
                            nhqf=nhqf+1;
                        end;
                        qd(nhqft)=1;
                        ib(nhqft)=0;
                        ilb(nhqft)=0;
                        jb(nhqft)=boxnum(i,j,k);
                        jrb(nhqft)=boxnum(i+1,j,k);
                        iface(nhqft)=i;
                        jface(nhqft)=j;
                        kface(nhqft)=k;
                        bndfn=bndfn+1;
                        bndfce(bndfn)=nhqft;
                        bndtxt(bndfn)=rtext(i,j,1);
                        bndid(bndfn)=rbnd(i,j,1);
                    end;
                end;
            elseif(river(i,j,1)>0||ocean(i,j,1)>0)
                if(k>=km(i-1,j))
                    nhqft=nhqft+1;
                    if(k==kmax)
                        nhqf=nhqf+1;
                    end;
                    qd(nhqft)=1;
                    ib(nhqft)=boxnum(i-1,j,k);
                    ilb(nhqft)=boxnum(i-2,j,k);
                    jb(nhqft)=0;
                    jrb(nhqft)=0;
                    iface(nhqft)=i;
                    jface(nhqft)=j;
                    kface(nhqft)=k;
                    bndfn=bndfn+1;
                    bndfce(bndfn)=nhqft;
                    bndtxt(bndfn)=rtext(i,j,1);
                    bndid(bndfn)=rbnd(i,j,1);
                end;
            end;
        end;
    end;
%
%  y-sweep
%
    for i=1:imax-1
        for j=1:jmax
            if(bexist(i,j,k)>0)
                if(j>1)
                    if(bexist(i,j-1,k)>0)
                        if(bar(i,j,2)==0)
                            nhqft=nhqft+1;
                            if(k==kmax)
                                nhqf=nhqf+1;
```

```
                    end;
                    qd(nhqft)=2;
                    ib(nhqft)=boxnum(i,j-1,k);
                    jb(nhqft)=boxnum(i,j,k);
                    jrb(nhqft)=boxnum(i,j+1,k);
                    if(j>2)
                        ilb(nhqft)=boxnum(i,j-2,k);
                    else
                        ilb(nhqft)=0;
                    end;
                    iface(nhqft)=i;
                    jface(nhqft)=j;
                    kface(nhqft)=k;
                end;
            elseif(river(i,j,2)>0||ocean(i,j,2)>0)
                if(k>=km(i,j))
                    nhqft=nhqft+1;
                    if(k==kmax)
                        nhqf=nhqf+1;
                    end;
                    qd(nhqft)=2;
                    ib(nhqft)=boxnum(i,j-1,k);
                    jb(nhqft)=boxnum(i,j,k);
                    jrb(nhqft)=boxnum(i,j+1,k);
                    if(j>2)
                        ilb(nhqft)=boxnum(i,j-2,k);
                    else
                        ilb(nhqft)=0;
                    end;
                    iface(nhqft)=i;
                    jface(nhqft)=j;
                    kface(nhqft)=k;
                    bndfn=bndfn+1;
                    bndfce(bndfn)=nhqft;
                    bndtxt(bndfn)=rtext(i,j,2);
                    bndid(bndfn)=rbnd(i,j,2);
                end;
            end;
        elseif(river(i,j,2)>0||ocean(i,j,2)>0)
            if(k>=km(i,j))
                nhqft=nhqft+1;
                if(k==kmax)
                    nhqf=nhqf+1;
                end;
                qd(nhqft)=2;
                ib(nhqft)=0;
                ilb(nhqft)=0;
                jb(nhqft)=boxnum(i,j,k);
                jrb(nhqft)=boxnum(i,j+1,k);
                iface(nhqft)=i;
                jface(nhqft)=j;
                kface(nhqft)=k;
                bndfn=bndfn+1;
                bndfce(bndfn)=nhqft;
                bndtxt(bndfn)=rtext(i,j,2);
                bndid(bndfn)=rbnd(i,j,2);
            end;
        end;
    elseif(river(i,j,2)>0||ocean(i,j,2)>0)
        if(j>1&&k>=km(i,j-1))
            nhqft=nhqft+1;
```

```
                    if(k==kmax)
                        nhqf=nhqf+1;
                    end;
                    qd(nhqft)=2;
                    ib(nhqft)=boxnum(i,j-1,k);
                    ilb(nhqft)=boxnum(i,j-2,k);
                    jb(nhqft)=0;
                    jrb(nhqft)=0;
                    iface(nhqft)=i;
                    jface(nhqft)=j;
                    kface(nhqft)=k;
                    bndfn=bndfn+1;
                    bndfce(bndfn)=nhqft;
                    bndtxt(bndfn)=rtext(i,j,2);
                    bndid(bndfn)=rbnd(i,j,2);
                end;
            end;
        end;
    end;
end;
%
%  vertical faces
%
nqf=nhqft;
for j=1:jmax-1
    for i=1:imax-1
        if(bexist(i,j,kmax)>0)
            sfbox=boxnum(i,j,kmax);
            bbox(sfbox)=boxnum(i,j,km(i,j));
            nvf(sfbox)=kmax-km(i,j);
            if(km(i,j)<kmax)
                for k=km(i,j)+1:kmax
                    nqf=nqf+1;
                    if(k==km(i,j)+1)
                        ilb(nqf)=0;
                    else
                        ilb(nqf)=boxnum(i,j,k-2);
                    end;
                    ib(nqf)=boxnum(i,j,k-1);
                    jb(nqf)=boxnum(i,j,k);
                    if(k==kmax)
                        jrb(nqf)=0;
                    else
                        jrb(nqf)=boxnum(i,j,k+1);
                    end;
                    qd(nqf)=3;
                    vfn(sfbox,k)=nqf;
                end;
            end;
        end;
    end;
end;
%
%-----------------------------------------------------------------
%
%
%  write to file95
%
[file95, pathname]=uiputfile('*.95','Map Info File');
fid95=fopen([pathname file95],'wt');
%
```

```
%
%
fprintf(fid95,'File 95: Face info for CH3D and ICM\n');
fprintf(fid95,'SCK\n');
fprintf(fid95,date);
fprintf(fid95,'\n');
fprintf(fid95,':\n');
fprintf(fid95,':\n');
%
%
%
fprintf(fid95,'   NHQFT     NQF    NHQF\n');
fprintf(fid95,'%8d',nhqft);
fprintf(fid95,'%8d',nqf);
fprintf(fid95,'%8d\n',nhqf);
%
%
%
fprintf(fid95,'       F      QD     ILB      IB      JB     JRB');
fprintf(fid95,'      KP      KF      KL   LAYER\n');
%for i=1:nhqft
for i=1:nqf
    fprintf(fid95,'%8d',i);
    fprintf(fid95,'%8d',qd(i));
    fprintf(fid95,'%8d',ilb(i));
    fprintf(fid95,'%8d',ib(i));
    fprintf(fid95,'%8d',jb(i));
    fprintf(fid95,'%8d',jrb(i));
    switch qd(i)
        case 1
            if(jb(i)==0)
                inum=ibox(ib(i))+1;
                jnum=jbox(ib(i));
                knum=kbox(ib(i));
            else
                inum=ibox(jb(i));
                jnum=jbox(jb(i));
                knum=kbox(jb(i));
            end;
            fprintf(fid95,'%8d',inum);
            fprintf(fid95,'%8d',jnum);
            fprintf(fid95,'%8d',jnum);
            fprintf(fid95,'%8d',knum);
            fprintf(fid95,'\n');
        case 2
            if(jb(i)==0)
                inum=ibox(ib(i));
                jnum=jbox(ib(i))+1;
                knum=kbox(ib(i));
            else
                inum=ibox(jb(i));
                jnum=jbox(jb(i));
                knum=kbox(jb(i));
            end;
            fprintf(fid95,'%8d',jnum);
            fprintf(fid95,'%8d',inum);
            fprintf(fid95,'%8d',inum);
            fprintf(fid95,'%8d',knum);
            fprintf(fid95,'\n');
        case 3
            fprintf(fid95,'%8d',jbox(jb(i)));
```

```
            fprintf(fid95,'%8d',ibox(jb(i)));
            fprintf(fid95,'%8d',ibox(jb(i)));
            fprintf(fid95,'%8d',kbox(jb(i))-1);
            fprintf(fid95,'%8d',kbox(jb(i)));
            fprintf(fid95,'\n');
    end;
end;
%
%
%
fprintf(fid95,'\n');
fprintf(fid95,'  SFC BOX #     (NVF(SB), SB=1,NSB)\n');
nline=floor(nsb/8);
nextra=mod(nsb,8);
for il=1:nline
    ibeg=(il-1)*8+1;
    iend=ibeg+7;
    fprintf(fid95,'%5d',ibeg);
    fprintf(fid95,'-');
    fprintf(fid95,'%5d',iend);
    for k=ibeg:iend
        fprintf(fid95,'%8d',nvf(k));
    end;
    fprintf(fid95,'\n');
end;
if (nextra>0)
    ibeg=iend+1;
    iend=ibeg+nextra-1;
    fprintf(fid95,'%5d',ibeg);
    fprintf(fid95,'-');
    fprintf(fid95,'%5d',iend);
    for k=ibeg:iend
        fprintf(fid95,'%8d',nvf(k));
    end;
    fprintf(fid95,'\n');
end;
%
%
%
fprintf(fid95,'\n');
fprintf(fid95,'  BOT BOX #     (VFN(F,SB), F=1,NVF(SB)\n');
for i=1:nsb
    fprintf(fid95,'%8d',bbox(i));
    if(nvf(i)>0)
        iprcnt=0;
        for k=nvf(i):-1:1
            iprcnt=iprcnt+1;
            if(iprcnt>1&&mod(iprcnt,9)==1)
                fprintf(fid95,'\n        ');
            end;
            fprintf(fid95,'%8d',vfn(i,kmax-k+1));
        end;
    end;
    fprintf(fid95,'\n');
end;
%
%
%
fclose(fid95);
%
%---------------------------------------------------------------
```

```
%
%
%  write to filegeo
%
[filegeo, pathname]=uiputfile('wqmgeo.*','Geo File for ICM');
fidgeo=fopen([pathname filegeo],'wt');
%
%
%
fprintf(fidgeo,'C: GEO input file for ICM\n');
fprintf(fidgeo,'C: SCK, ');
fprintf(fidgeo,date);
fprintf(fidgeo,'\n');
fprintf(fidgeo,'   BOX #  B#_K+1\n');
fprintf(fidgeo,'\n');
for i=1:nsb
    fprintf(fidgeo,'%8d%8d\n',i,0);
end;
for i=nsb+1:nb
    fprintf(fidgeo,'%8d%8d\n',i,...
        boxnum(ibox(i),jbox(i),kbox(i)+1));
end;
fprintf(fidgeo,'\n');
fprintf(fidgeo,'    SBOX    BBOX\n');
for i=1:nsb
    inum=ibox(i);
    jnum=jbox(i);
    knum=km(inum,jnum);
    fprintf(fidgeo,'%8d%8d\n',i,boxnum(inum,jnum,knum));
end;
fclose(fidgeo);
%
%-----------------------------------------------------------------
%
%
%  write to filecol
%
[filecol, pathname]=uiputfile('wqmcol.*','Postprocessing Column File');
fidcol=fopen([pathname filecol],'wt');
%
%
%
for i=1:nsb
    inum=ibox(i);
    jnum=jbox(i);
    fprintf(fidcol,'%3d %3d',inum,jnum);
    fprintf(fidcol,' %2d',nlayer(inum,jnum));
    for k=kmax:-1:km(inum,jnum)
        fprintf(fidcol,'%6d',boxnum(inum,jnum,k));
    end;
    fprintf(fidcol,'\n');
end;
fclose(fidcol);
%
%-----------------------------------------------------------------
%
%
%  write to filebnd
%
[filebnd, pathname]=uiputfile('bndface.*','Boundary Face File');
fidbnd=fopen([pathname filebnd],'wt');
```

```
%
%
%
ncbnd=zeros(ir,1);
for i=1:bndfn
    fnum=bndfce(i);
    fprintf(fidbnd,'%5d %6d %3d %3d %2d',i,fnum,...
        iface(fnum),jface(fnum),kface(fnum));
    fprintf(fidbnd,' %s\n',bndtxt{i});
    j=bndid(i);
    ncbnd(j)=ncbnd(j)+1;
    k=ncbnd(j);
    bndfid(j,k)=i;
end;
for j=1:ir
    fprintf(fidbnd,'%s\n',nrtext{j});
    fprintf(fidbnd,'%5d\n',ncbnd(j));
    nline=floor(ncbnd(j)/8);
    nextra=mod(ncbnd(j),8);
    ibeg=0;
    iend=0;
    if(nline>0)
        for il=1:nline
            ibeg=(il-1)*8+1;
            iend=ibeg+7;
            for k=ibeg:iend
                fprintf(fidbnd,'%8d',bndfid(j,k));
            end;
            fprintf(fidbnd,'\n');
        end;
    end;
    if (nextra>0)
        ibeg=iend+1;
        iend=ibeg+nextra-1;
        for k=ibeg:iend
            fprintf(fidbnd,'%8d',bndfid(j,k));
        end;
        fprintf(fidbnd,'\n');
    end;
end;
fclose(fidbnd);
%
%
%
%
%-----------------------------------------------------------------
%
%
% read in grid data
%
[file15, pathname]=uigetfile('*.*','Grid File');
fid_15=fopen([pathname file15],'rt');
%
%
%
itest=1;
while (itest>0)
    hdrt=fscanf(fid_15,'%s',1);
    imax=str2num(hdrt);
    if(~isempty(imax))
        break;
```

```
    end;
end;
jmax=fscanf(fid_15,'%g',1);
tline=fgets(fid_15);
for j=1:jmax
    for i=1:imax
        tline=fgets(fid_15);
        X=sscanf(tline,'%g %g',[1,2]);
        x(i,j)=xscale*X(1);
        y(i,j)=xscale*X(2);
    end;
end;
clear X;
fclose(fid_15);
%
%-----------------------------------------------------------------
%
%
%
%  write to GIS
%
[filegis, pathname]=uiputfile('CH3D_ICM_GIS.*','Boundary Face File');
fidgis=fopen([pathname filegis],'wt');
%
%
%
fprintf(fidgis,'%s,','CH3D');
fprintf(fidgis,'%s','DEPTH');
for k=1:kmax
    fprintf(fidgis,',%s',strcat('L',num2str(k)));
end;
fprintf(fidgis,'\n');
%hold on;
for ib=1:nsb
    i=ibox(ib);
    j=jbox(ib);
    ij=i*1000+j;
    fprintf(fidgis,'%6.6d,',ij);
    fprintf(fidgis,'%g',depth(i,j));
    for k=kmax:-1:1
        fprintf(fidgis,',%g',boxnum(i,j,k));
    end;
    fprintf(fidgis,'\n');
    xx(1)=x(i,j);yy(1)=y(i,j);
    xx(2)=x(i+1,j);yy(2)=y(i+1,j);
    xx(3)=x(i+1,j+1);yy(3)=y(i+1,j+1);
    xx(4)=x(i,j+1);yy(4)=y(i,j+1);
    xx(5)=xx(1);yy(5)=yy(1);
    for l=1:5
        fprintf(fidgis,'%g,%g\n',xx(l),yy(l));
    end;
    fprintf(fidgis,'%s\n','END');
end;
fprintf(fidgis,'END\n');
fclose(fidgis);
```

Appendix I. Function “plot_grid”

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% plot grid 1
%
function plot_grid_1(imax,jmax,depth,obox,river,bar)

    h1=figure;
    hold on;
    for j=1:jmax-1
        for i=1:imax-1
            if(depth(i,j)>0)
                xl(1)=i;
                yl(1)=j;
                xl(2)=i+1;
                yl(2)=j;
                xl(3)=i+1;
                yl(3)=j+1;
                xl(4)=i;
                yl(4)=j+1;
                xl(5)=xl(1);
                yl(5)=yl(1);
                plot(xl,yl);
            end;
            if(obox(i,j)>0)
                xl(1)=i;
                yl(1)=j;
                xl(2)=i+1;
                yl(2)=j;
                xl(3)=i+1;
                yl(3)=j+1;
                xl(4)=i;
                yl(4)=j+1;
                xl(5)=xl(1);
                yl(5)=yl(1);
                plot(xl,yl,'r');
                plot([xl(1) xl(3)],[yl(1) yl(3)],'r');
                plot([xl(2) xl(4)],[yl(2) yl(4)],'r');
            end;
        end;
    end;
    clear xl yl;
    for j=1:jmax
        for i=1:imax
            if(river(i,j,1)>0)
                xl(1)=i-0.5;
                yl(1)=j+0.5;
                xl(2)=i+0.5;
                yl(2)=yl(1);
                plot(xl,yl,'r');
                xl(1)=xl(2)-0.3;
                yl(1)=yl(2)+0.3;
                plot(xl,yl,'r');
                yl(1)=yl(2)-0.3;
```

```
                plot(xl,yl,'r');
            end;
            if(bar(i,j,1)>0)
                xl(1)=i;
                yl(1)=j;
                xl(2)=i;
                yl(2)=j+1;
                plot(xl(1:2),yl(1:2),'g');
            end;
        end;
    end;
    for i=1:imax
        for j=1:jmax
            if(river(i,j,2)>0)
                xl(1)=i+0.5;
                yl(1)=j-0.5;
                xl(2)=xl(1);
                yl(2)=j+0.5;
                plot(xl,yl,'r');
                xl(1)=xl(2)-0.3;
                yl(1)=yl(2)-0.3;
                plot(xl,yl,'r');
                xl(1)=xl(2)+0.3;
                plot(xl,yl,'r');
            end;
            if(bar(i,j,2)>0)
                xl(1)=i;
                yl(1)=j;
                xl(2)=i+1;
                yl(2)=j;
                plot(xl(1:2),yl(1:2),'g');
            end;
        end;
    end;
    axis equal;
    drawnow;
```