

Documentation of Programs Used to Determine a Wetlands Hydroperiod from Model-Simulated Water-Surface Elevations

By Roy S. Sonenshein

U.S. GEOLOGICAL SURVEY

Open-File Report 96-455

Prepared in cooperation with the
South Florida Water Management District

Tallahassee, Florida
1996



U.S. DEPARTMENT OF THE INTERIOR
BRUCE BABBITT, Secretary

U.S. GEOLOGICAL SURVEY
Gordon P. Eaton, Director

Any use of trade, product, or firm names in this publication is for descriptive purposes only and does not imply endorsement by the U.S. Geological Survey

For additional information, write to:

District Chief
U.S. Geological Survey
Suite 3015
227 N. Bronough Street
Tallahassee, Florida 32301

Copies of this report can be purchased from:

U.S. Geological Survey
Branch of Information Services
Box 25286
Denver, CO 80225-0286

CONTENTS

Abstract.....	1
Introduction	1
Documentation of Programs Used to Determine a Wetlands Hydroperiod.....	2
Technique for Determining a Wetlands Hydroperiod.....	2
MODFLOW Post-Processor.....	2
Geographic Information System Programs and Menus.....	4
Sample Output	5
Program Code and Menu Listings	6
modbin2arc.c	11
hyper.aml	19
modbin2arc.aml	23
land.aml	25
hydroperiod.aml.....	31
modbin2arc.menu	39
hyper.menu	43
References Cited.....	47

FIGURES

1. Program flow chart for the technique to determine a wetlands hydroperiod.....	3
2. Flow chart for the land.aml program	6
3. Flow chart for the hydroperiod.aml program.....	7
4. Schematics showing lines of equal water level, lines of equal land-surface elevation, and zonal representations of hydroperiods for stress periods 1 and 2.....	8

TABLES

1. Description of computer programs	3
2. Description of the input data set for the ANSI compatible C program, modbin2arc.c	4
3. Description of the output data set for the ANSI compatible C program, modbin2arc.c	5
4. Command line arguments for the land.aml program	9
5. Description of the input data set for the land.aml program	9
6. Command line arguments for the hydroperiod.aml program.....	9

Documentation of Programs Used to Determine a Wetlands Hydroperiod from Model-Simulated Water-Surface Elevations

By Roy S. Sonenshein

Abstract

A technique has been developed to determine a wetlands hydroperiod by comparing simulated water levels from a ground-water flow model and land-surface elevation data through a geographic information system. The simulated water levels are compared with the land-surface elevation data to determine the height of the water surface above or below land surface for the area of interest. Finally, the hydroperiod is determined for established time periods using criteria specified by the user.

The program application requires the use of geographic information system software (ARC/INFO), including the TIN and GRID subsystems of the software. The application consists of an ANSI compatible C program to translate ground-water data output from the U.S. Geological Survey modular three-dimensional, finite-difference, ground-water flow model (MODFLOW) into a format that can be used as input for the geographic information system programs (AML's). The application uses ARC/INFO AML programs and ARC/INFO menu interface programs to create digital spatial data layers of the land surface and water surface and to determine the hydroperiod. The technique can be used to evaluate and manage wetlands hydrology.

INTRODUCTION

Wetlands serve a variety of functions, including providing area for floodwater storage and, in some places, aquifer recharge and as highly productive and diverse wildlife and floral habitats. Until recently, the

preservation of wetlands had been of minimal concern because of the relative abundance of freshwater, fish and other wildlife, and the desire to drain and fill these areas for development. Now the value of wetlands in sustaining the living standards of humanity has become widely recognized (Maltby and Dugan, 1994, p. 30); in order to preserve these wetlands, the remaining wetlands must be delineated.

A major factor in wetlands delineation is the hydroperiod, which is defined as the length of time of seasonal flooding. The hydroperiod varies seasonally and from year to year, and is the major determination of wetlands functions (Mitsch and Gosselink (1993, p. 67). Because hydrologic data for wetlands areas are often sparse or nonexistent, hydroperiods are rarely known with a high degree of certainty.

Specific vegetation types require certain hydroperiods to survive. In some cases, upper and lower bounds of hydroperiods can be inferred from local vegetation; however, this method does not provide any predictive capability. In other cases, hydrologic models can be used to predict the water-surface elevation, but many of the hydrologic models do not contain accurate land-surface data at the scale required to define the hydroperiod. A geographic information system (GIS) has the capability to store and manipulate land-surface and water-level data, but lacks comprehensive hydrologic modeling capability. By combining the capabilities of a GIS and an existing ground-water flow model, it is possible to determine the hydroperiod of any region within the model grid where land-surface data exist. The relation between other data, such as vegetation type and hydroperiod, can then be evaluated using the results. In 1993, the U.S. Geological Survey, in cooperation with the South Florida Water Management District, began a study to develop a technique to estimate the wetlands hydroperiod by comparing simulated water levels to land-surface elevation data using a GIS.

The purpose of this report is to describe a technique and document computer program applications for determining the hydroperiod of an area. The technique requires a data set of land-surface elevations and water-surface elevations simulated by a ground-water flow model. The technique is intended to be used as a predictive tool for evaluating and managing water resources in wetlands for any area within the grid of a ground-water flow model.

The computer program description and documentation include data requirements and sample output. Ground-water data, which are output by the modular three-dimensional, finite-difference, ground-water flow model, MODFLOW (McDonald and Harbaugh, 1988), are used for input into a program. The GIS software package, ARC/INFO (Environmental Systems Research Institute, 1990), is used to create digital spatial data layers of the land surface and water surface and to determine the hydroperiod. The TIN and GRID packages of the ARC/INFO software are required for this application.

DOCUMENTATION OF PROGRAMS USED TO DETERMINE A WETLANDS HYDROPERIOD

The technique to determine a wetlands hydroperiod from model-simulated water-surface elevations was automated using four ARC/INFO AML programs, two ARC/INFO menu interface programs, and an ANSI compatible C program (fig. 1 and table 1). The main AML program (`hyper.aml`) executes the two menu programs (`modbin2arc.menu` and `hyper.menu`) and the three remaining AML programs (`modbin2arc.aml`, `land.aml`, and `hydroperiod.aml`). The `modbin2arc.menu` program is an input form to enter the data required to execute the ANSI compatible C program, `modbin2arc.c`. The `hyper.menu` program is an input form to enter the data required to execute two AML programs, `land.aml` and `hydroperiod.aml`.

Technique for Determining a Wetlands Hydroperiod

The technique to determine a wetlands hydroperiod uses the results of a ground-water flow model simulation as input to a GIS. To calculate the hydroperiod, a water surface must be compared to a land-surface model of the same area through time. In this technique,

MODFLOW binary-format head output files for a transient simulation are used to define the water surface at the end of each time step in the simulation. The technique also allows water-level data output from other models or water-level data interpolated from measurements to be used if the data are in the required format. Land-elevation data are supplied by the user in a text file. These two data layers, water surface and land elevation, are compared, and a data layer defining the hydroperiod for each time step in the simulation is generated. The hydroperiods for each time step are summed to generate the hydroperiod over the entire simulation.

Care must be taken in evaluating the output from the computer program applications described in this report. Ground-water flow models are an approximation of reality. Several limiting factors must be considered when using the output from a ground-water flow model. The uncertainty in hydrogeologic parameters and boundary conditions is among the most important limitations in a model of a ground-water system. Another limitation is the use of a discrete network of finite-difference cells to represent the ground-water flow system. As the cell size increases, the accuracy of the model in representing the ground-water flow system decreases. Finally, differences in data handling between MODFLOW and the ARC/INFO GIS system must be considered. MODFLOW calculates conditions at the center of each grid cell at a specified time. ARC/INFO is used to calculate the water level at any point within the area of the ground-water flow model by interpolating between the values determined for the center of each grid cell. If the area of interest is much smaller than the grid cell, the model output may then not be truly representative of the water levels in the area.

MODFLOW Post-Processor

The ANSI compatible C program, `modbin2arc.c` (table 1), extracts the simulated head data from the binary-format file output by the MODFLOW Basic model package (version 1, 9/1/87). The program requires a data file as a command line argument. The format for this file is defined in table 2.

The `modbin2arc.c` program outputs a series of data files, one for every time step in each stress period for the model layer of interest. Files contain reformatting simulated head values for each grid cell in the MODFLOW simulation. For example, a simulation

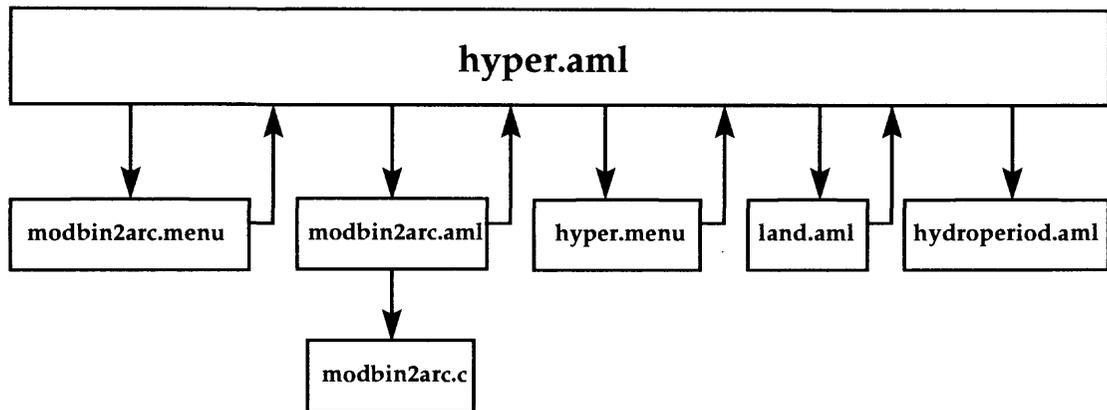


Figure 1. Program flow chart for the technique to determine a wetlands hydroperiod.

Table 1. Description of computer programs

Size of file (bytes)	Name	Description
15340	modbin2arc.c	C code to convert MODFLOW binary data to ARC/INFO input data set
2640	hyper.aml	Main ARC/INFO AML - runs menus and other AML's
1610	modbin2arc.aml	ARC/INFO AML to create input data set for modbin2arc.c and execute program
5756	land.aml	ARC/INFO AML to create land elevation GRID
11854	hydroperiod.aml	ARC/INFO AML to determine hydroperiods
3063	modbin2arc.menu	ARC/INFO forms menu to enter data for modbin2arc.c
4480	hyper.menu	ARC/INFO forms menu to enter data for land.aml and hydroperiod.aml

Table 2. Description of the input data set for the ANSI compatible C program, modbin2arc.c

Line number	Sample input	Format	Description of input field
1	hydro	Character	Keyword
2	7	Integer	Number of time steps in each stress period
3	1	Integer	Model layer number to be evaluated
4	88.88	Real	Head value in MODFLOW output for dry cells
5	550188	Real	Minimum x-coordinate value of the MODFLOW grid
6	2849191	Real	Maximum y-coordinate value of the MODFLOW grid these coordinates reference the upper left corner of the grid and not the center of the upper left cell
7	hydro.delrc	Character	The MODFLOW horizontal grid spacing (.delrc) file. This program requires the row and column arrays to be in a separate file from the Basic package file. See McDonald and Harbaugh (1988, p. 14-35 to 14-38) for a description of the format of this file.
8	hydro.hbin	Character	The MODFLOW binary head output (.hbin) file. See McDonald and Harbaugh (1988, p 4-14 to 4-17) for a description of how to create this file using the MODFLOW variable IHEDUN.
9	4	Integer	Number of layers in the MODFLOW simulation
10	31	Integer	Number of rows in the MODFLOW simulation
11	25	Integer	Number of columns in the MODFLOW simulation
12	46	Integer	Number of stress periods in the MODFLOW simulation
13	99.99	Real	Head value in MODFLOW output for no-flow cells

with three stress periods each having seven time steps generates a total of 21 files. The files are named using the convention:

keyword_layer_stress-period_time-step_total-time

where the total-time is the elapsed time from the beginning of the MODFLOW simulation to the end of the current time step. An example file name for keyword hydro, model layer 1, stress period 30, time step 1, and elapsed time of 204 is:

hydro_1_30_1_204.0

These files are formatted for input to the AML program, hydroperiod.aml, generated by GIS which determines the hydroperiod and contains one line of data for each grid cell. Other model output may also be used as input as long as the data files are formatted as required by the AML program. The format for the

output data set is presented in table 3. Fields are comma delimited, and the files must end with a line containing the word "END."

Geographic Information System Programs and Menus

Two ARC/INFO GIS macro programs (AML's), land.aml and hydroperiod.aml (table 1), generate the land-elevation data layer and determine the hydroperiod and require the TIN and GRID packages of ARC/INFO. The TIN package is used to generate surface maps (called a TIN) of land-elevation and water-level data. The GRID package is used to create uniformly spaced raster data sets (called a GRID) of land-elevation and water-level elevation data from the TIN surface maps. Raster data sets are compared and summed

Table 3. Description of the output data set for the ANSI compatible C program, modbin2arc.c

[The last line of the file is the word "END"]

Field number	Sample output	Format	Description of output field
1	1	Integer	Identification number (does not have to be unique)
2	552130	Real	x-coordinate for the center of grid cell
3	2849141	Real	y-coordinate for the center of grid cell
4	2.4966	Real	Water-surface elevation of the specified grid cell. No-flow and dry cells have the elevation set to -99.99.

using the GRID package to determine the hydroperiod. Flow diagrams outlining the steps used by the land.aml and hydroperiod.aml programs are shown in figures 2 and 3, respectively.

The land.aml program uses the TIN and GRID packages to create a uniformly spaced point data set of land-surface elevations from user-provided land-surface data (fig. 2). The program-required command-line arguments are given in table 4, and the required input data set format is given in table 5.

The hydroperiod.aml program makes a copy of the land-surface elevation point data set created by the land.aml program for each stress period (fig. 3). The program calculates two values at each point in the data set for every time step in the stress period and stores these values in the GIS point attribute table. The values are represented by the: (1) water level interpolated from the MODFLOW simulation using the TIN package, and (2) time (hydroperiod) that the water level is at or above the user-specified depth below land surface. The user-specified depth is the distance of water below land surface that defines a wetlands. The time is always equal to the length of the time step.

The hydroperiod.aml program also creates an ARC/INFO grid for each stress period. The spacing of each grid is identical to the spacing of the land-surface point data set, with each data point located at the center of a grid cell and assigned a value (h_s) equal to the sum of all the hydroperiods determined for every time step (h_n), as expressed in the following equation:

$$h_s = \sum_{n=1}^{\text{Number of timesteps}} h_n \quad (1)$$

Another grid is created and assigned point values (h_t) equal to the sum of the hydroperiods (h_s) determined for each stress period, as expressed in the following equation:

$$h_t = \sum_{n=1}^{\text{Number of stress periods}} h_s \quad (2)$$

The grid represents the hydroperiod for the total time period. Related INFO tables are created for each grid, giving the total area for each hydroperiod within the grid. The hydroperiod.aml program requires the command line arguments presented in table 6.

The two ARC/INFO menu interface programs, modbin2arc.menu and hyper.menu (table 1), were developed to automate the running of the MODFLOW post-processor and GIS interface programs. The user enters the required data using these screen menus and runs the program interactively.

Sample Output

Water-level output for two stress periods from a hypothetical MODFLOW simulation was compared with hypothetical land-surface elevations to determine the hydroperiod for individual stress periods and for the overall total simulation. The results are illustrated in figure 4, which shows lines of equal water level, lines of equal land-surface elevation, and zonal representations of hydroperiods. Wetlands hydroperiods were determined using a water level of 0.25 foot below land surface.

Program Code and Menu Listings

As previously stated, the technique to determine a wetlands hydroperiod from model-simulated water-surface elevations was automated using four ARC/INFO AML programs, two ARC/INFO interface menu programs, and an ANSI compatible C program (table 1). The code for the AML, menu, and C programs are presented on subsequent pages of this report.

The AML and C programs have been previously used by the U.S. Geological Survey (USGS). However, it should be noted that no warranty, expressed or implied, is made by the USGS as to the accuracy and functioning of the programs and related program material, nor shall distribution of this report constitute any such warranty. No responsibility is assumed by the USGS in connection herewith.

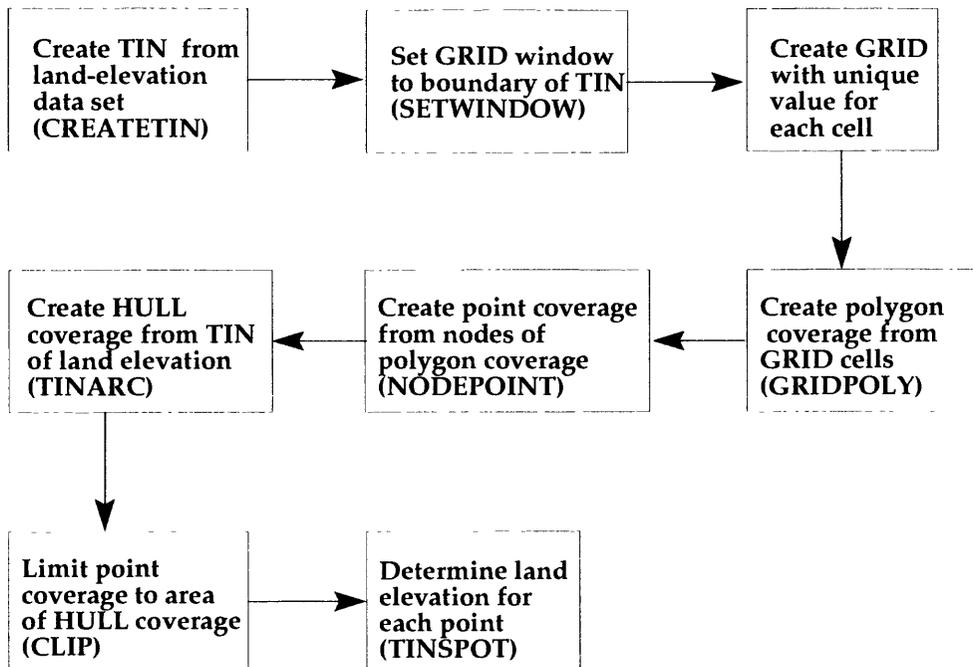


Figure 2. Flow chart for the land.aml program.

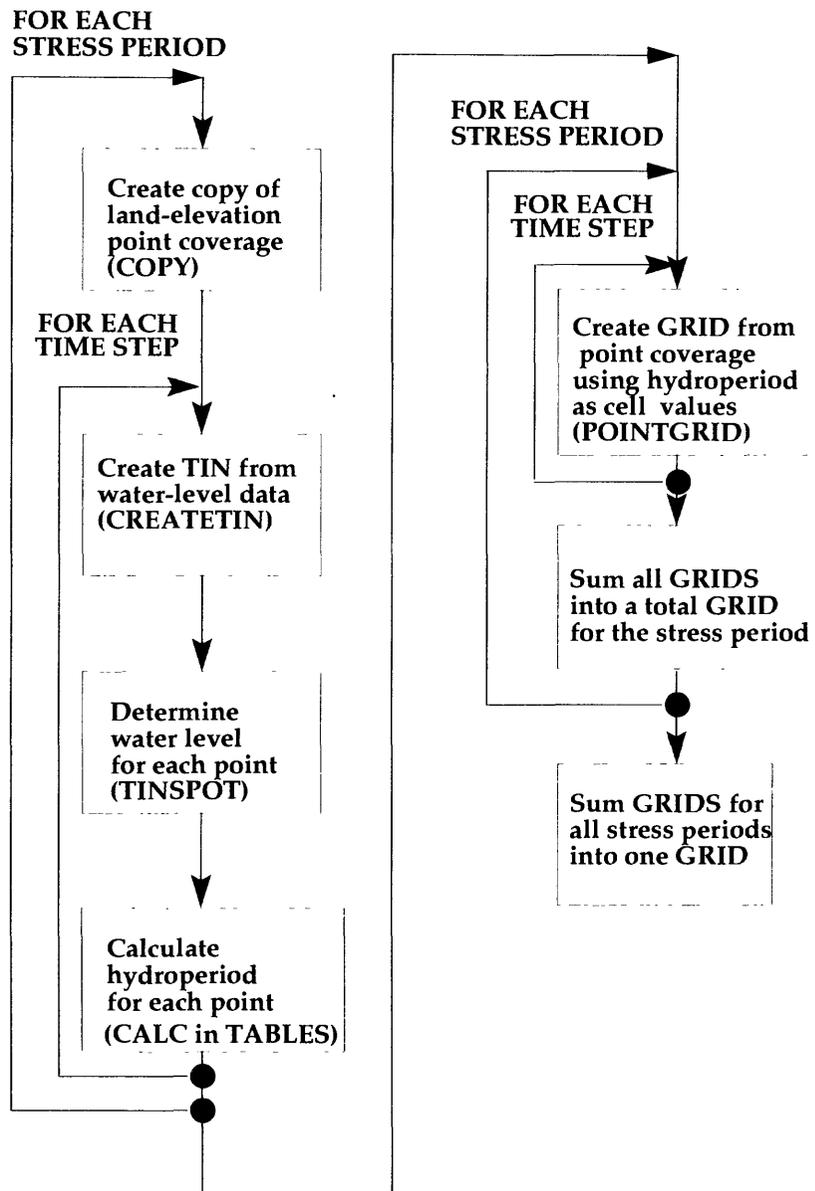
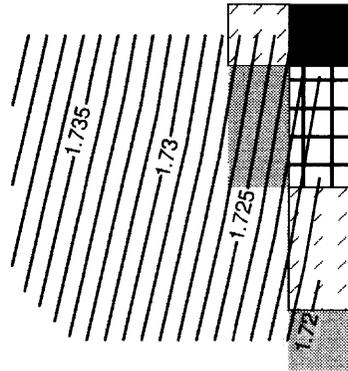
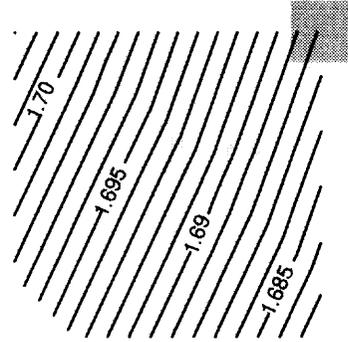


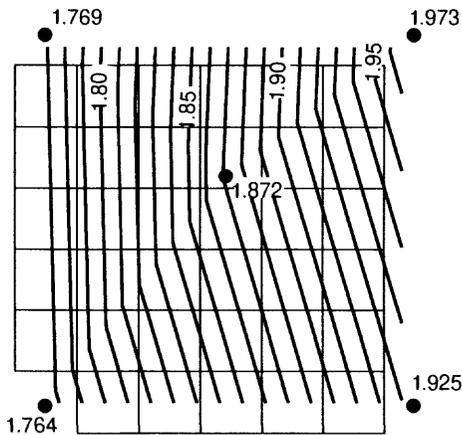
Figure 3. Flow chart for the hydroperiod.aml program.



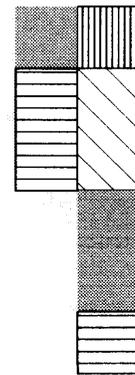
Lines of equal water-level at end of stress period 1 and hydroperiods for stress period 1



Lines of equal water-level at end of stress period 2 and hydroperiods for stress period 2



Data points, lines of equal land-surface elevation, and ARC/INFO grid cells



Total hydroperiods for stress periods 1 and 2

EXPLANATION

Percent of time water level at or above 0.25 ft below land surface

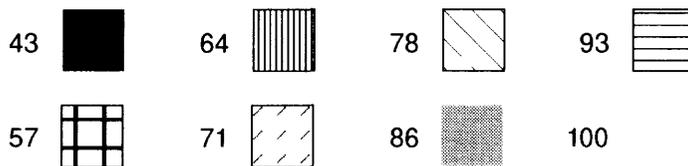


Figure 4. Schematics showing lines of equal water level, lines of equal land-surface elevation, and zonal representations of hydroperiods for stress periods 1 and 2.

Table 4. Command line arguments for the land.aml program

Sample argument	Format	Description of argument
land.dat	Character	Name of file containing point elevation data in format described in table 5.
50	Integer	Specified dimension of the square cells to be used to model land surface. The choice of this dimension is critical. Changing this dimension can significantly affect the results of the program.
land	Character	Output directory name to contain the land-surface data layer. This must be a directory that DOES NOT exist in the directory where the program is being executed.

Table 5. Description of the input data set for the land.aml program

Field number	Sample input	Format	Description of input field
1	1	Integer	Identification number (does not have to be unique)
2	553330	Real	x-coordinate
3	2846797	Real	y-coordinate
4	1.744	Real	Land-surface elevation

Table 6. Command line arguments for the hydroperiod.aml program

Sample argument	Format	Description of argument
hydro	Character	Keyword used for the MODFLOW interface program
1	Integer	MODFLOW layer number
50	Integer	Dimension of the square cells used to model land surface in the land.aml program
land	Character	Output point spatial data layer containing the land surface data layer. This data layer was created by the land.aml program
1	Integer	First stress period to use for simulation
3	Integer	Last stress period to use for simulation (greater than or equal to first stress period)
204	Real	Starting time for first stress period
0.25	Real	Specified depth of water table below land surface considered to be a wetlands

modbin2arc.c

```

/*****
/*  C Program modbin2arc
/*  Written by Kevin B Kotun, US Geological Survey, Miami, Florida 4/95
/*  Modified by Roy S Sonenshein, US Geological Survey, Miami, Florida 2/96
/*****
/*****
/*  This code will read xmin, ymax of a model grid and along with the
/*  delrc file will locate the model grid in consistent geographic coords
/*  and read head arrays per timestep from the .hbin file. No flow and dry
/*  cells are assigned a value of -99.99 in the output files.
/*****
/*****
/*  The arrays are set to handle a model grid of a maximum size of
/*  60000 grid cells, with a maximum of either 1000 rows or 1000 columns
/*  To increase these sizes, modify the following array sizes
/*  delr[1000]    xcell[1000]  (delr and xcell must be the same size)
/*  delc[1000]    ycell[1000]  (delc and ycell must be the same size)
/*  head[60000]  rbuff[60000] (head and rbuff must be the same size)
/*****
/*****
/*LIBRARIES*/
/*****
#include <stdio.h>
#include <stdlib.h>
/*****
/*  GLOBAL VARIABLES */
/*****
/*
/*  nlay: No. of layers in model grid
/*  nper: No. of stress periods in simulation
/*  nstp: No. of time steps in each stress period
/*  nrow: No. of rows in model grid
/*  ncol: No. of columns in model grid
/*  klay: layer counter
/*  kper: stress period counter
/*  kstp: time step counter
/*  xmin: x-coordinate of upper left corner of model grid (not cell center)
/*  ymax: y-coordinate of upper left corner of model grid (not cell center)
/*  totim: running duration of simulation
/*
/*****
/*  LOCAL VARIABLES */
/*****
/*  *delrc: pointer to file containing row and column discretization
/*  *data: pointer to file with data needed to run this program
/*  *outfile: pointer to file where output is desired to be located
/*  *hbin: pointer to file with binary MODFLOW output
/*  datafile: char string that holds the name of the file where the program
/*            reads in necessary information. This string is used as an
/*            argument to the program
/*  rowcol: char string that holds the name of the MODFLOW delrc file
/*  keywrd: char string used as prefix to all output files
/*  outstrg: char string that holds the name of the output file for head
/*  values

```

```

/* bcf_fle: char string that holds the name of the block centered flow      **
/*      package file                                                         **
/* hbin_fle:char string that holds the name of the binary output file      **
/* token:  char string that is used to read in lines from MODFLOW input    **
/*      files                                                                 **
/* status:  integer function return                                         **
/* binstatus: integer function return                                       **
/* i,j:     loop counters                                                   **
/* keylay:  the layer of interest i.e. the layer that head values will     **
/*      be read and printed                                                 **
/* strleng: integer value that holds the length of a read string           **
/* delc:    discretization along a column (distance between rows)         **
/* delr:    discretization along a row   (distance between columns)       **
/* xcell:   Array that holds X-coordinates of cells                       **
/* ycell:   Array that holds Y-coordinates of cells                       **
/* head:    Head value                                                      **
/* hnoflo:  MODFLOW value for no-flow cells                               **
/* hdry:    MODFLOW value for dry cells                                    **
/*                                                                 **
/*****
/* GLOBAL VARIABLE DECLARATION */
/*****
int nlay, nper, nstp, nrow, ncol;
int klay, kper, kstp;
float xmin, ymax, totim;
/*****
/* FUNCTION PROTOTYPES */
/*****
/* int gethead(float, FILE*, int, float, float); */
/*****
/* MAIN LISTING */
/*****
void main(int argc, char **argv)
{
/*****
/* LOCAL DECLARATIONS */
/*****
FILE *delrc, *data, *outfile;
FILE *hbin;          /* I/O files */
char datafle[81], rowcol[81];
char keywrđ[41], outstrg[101];
char bcf_fle[41], hbin_fle[41];
char token[81];
int status, binstatus;
int i,j, keylay, strleng;
float delc[1000],delr[1000], xcell[1000], ycell[1000], head[60000];
float hnoflo, hdry;
/*****
/* SET FILE POINTER FOR DATA FILE */
/*****
if (argc < 2)
{
printf("\nThis program requires a filename as an argument.\n\
The file must contain the program parameters which come from\n\

```

```

describe the MODFLOW model (one per line):\n\n\
keyword (char string)\n\
number of timesteps (integer)\n\
the layer of interest (integer)\n\
the dry cell head value (real)\n\
minumum x-coordinate (real) upper left corner of grid\n\
maximum y-coordinate (real) not the cell center \n\
the .delrc file (char string)\n\
MODFLOW file of row and column spacing\n\
the .hbin file (char string)\n\
MODFLOW binary output head data set\n\
the number of layers (integer)\n\
the number of rows (integer)\n\
the number of columns (integer)\n\
the number of stress periods (integer)\n\
the no flow cell head value (real)\n\n");
    exit(0);
}
if ((data=fopen(argv[1],"r")) == NULL)
{
    printf("I cannot find the PROGRAM DATA file %s\n",argv[1]);
    exit(0);
}
/*****
/* SCAN DATA FILE FOR THIS PROGRAM */
/*****
fscanf(data,"%s%d%d%f%f%f%s%s%d%d%d%d%f",&keywrld,&nstp,&keylay,&hdry,&xmin,\
        &ymax,&rowcol,&hbin_file,&nlay,&nrow,&ncol,&nper,&hnoflo);
/*****
/* SET FILE POINTERS FOR I/O */
/*****
if ((hbin=fopen(hbin_file,"rb")) == NULL)
{
    printf("I cannot find the binary file %s\n",hbin_file);
    exit(0);
}

if ((delrc=fopen(rowcol,"r")) == NULL)
{
    printf("I cannot find the row column discretization file %s\n",rowcol);
    exit(0);
}
rewind(hbin);
rewind(delrc);
/*****
/* ECHO DATA FILE TO SCREEN */
/*****
printf("\n The binary file is in %s\n",hbin_file);
printf("\n The row and column file is %s\n",rowcol);
printf("\n The keyword is %s\n",keywrld);
printf("\n The layer of interest is %d\n",keylay);
printf("\n There are %d rows\n",nrow);
printf("\n There are %d columns \n",ncol);
printf("\n There are %d layers \n",nlay);

```

```

printf("\n There are %d stress periods \n",nper);
printf("\n No flow head is %f\n",hnoflo);
printf("\n Head for a dry cell is %f\n",hdry);
printf("\n Xmin is %f\n",xmin);
printf("\n Ymax is %f\n",ymax);

/*****
/* GET DISCRETIZATION FROM .DELRC FILE */
/*****
for(i=0;i<ncol;i++) fscanf(delrc,"%f",&delr[i]);
for(i=0;i<nrow;i++) fscanf(delrc,"%f",&delc[i]);
/*****
/* FUNCTION CALL FOR CELL CENTER LOCATIONS */
/*****
status= cellxys(&delr,&delc,&xcell,&ycell);
/*****
/* FUNCTION CALL FOR HEAD VALUES AT EACH CELL LAYER 1 */
/*****
/* INITIALIZE VARIABLES */
/*****
binstatus= 1;
kper= 0;
kstp= 0;
klay= 0;

while (kper != nper || kstp != nstp)
{
    binstatus = gethead(&head, hbin, keylay, &hnoflo, &hdry);
/*    printf("stress period= %d \ntime step= %d\n",kper,kstp); */
/*****
/*    OUTPUT */
/*****
    strlen= sprintf(outstrg,"%s_%d_%d_%d%.1f",keywr, klay,kper,kstp,totim);
    outfle=fopen(outstrg,"w");

    for(i=0;i<31;i++)
    {
        for(j=0;j<25;j++)
        {
            fprintf(outfle,"%d,%f,%f,%f\n",1,xcell[j],ycell[i],\
                head[((i*ncol)+j)]);
        }
    }
    fprintf(outfle,"END\n");
    fclose(outfle);
}

/*****
/* END MAIN */
/*****
/* FUNCTION TO CALCULATE CELL LOCATION */
/*****
int cellxys(float *delrptr,float *delcptr, float *xcellptr,\
    float *ycellptr)

```

```

{
int i;
        xcellptr[0]=xmin+0.5*delrp[0];
        ycellptr[0]=ymax-0.5*delcp[0];

for(i=1;i<=ncol;i++)
    {
        xcellptr[i]=xcellptr[i-1]+(0.5*(delrp[i-1]+delrp[i]));
    }

for(i=1;i<=nrow;i++)
    {
        ycellptr[i]=ycellptr[i-1]-(0.5*(delcp[i-1]+delcp[i]));
    }
return (-1);
}

/*****
/* FUNCTION gethead
** This function will read the .hbin file from MODFLOW and prepare nrow by
** ncol ASCII files for input to ARC to determine wether a cell is wet or not
** written by Kevin Kotun August 22 1994 */
/*****
int gethead(float *headptr, FILE *hbin, int keylay, float *hnoflo, float *hdry)

{
/*****
/* DECLARATIONS */
/*****
int k;          /* loop counters */
int pertim,text; /* "header values from .hbin for */
                /* each timestep */
int numread;    /* holds the number of values read with binary read function */
char chbuff[16]; /* buffer for char. variables */
int intbuff[3]; /* buffer for int. variables */
float rbuff[60000]; /* buffer for float variables */
/*****
/* READ FIRST TWO INTERGERS FROM HEADER */
/*****
while(feof(hbin)==0)
    {
numread=fread((void*)chbuff,4,1,hbin); /* read 1 cntrl character */
numread=fread((void*)intbuff,4,2,hbin);

        kstp= intbuff[0];
        kper= intbuff[1];
/*****
/* READ FIRST TWO REALS FROM HEADER */
/*****
numread=fread((void*)rbuff,4,2,hbin); /* read 2 cntrl characters */

```

```

pertim= rbuff[0];
totim= rbuff[1];
/*****
/* READ TEXT STRING FROM HEADER */
/*****
numread=fread((void*)chbuff,16,1,hbin);
/*****
/* READ NEXT THREE INTEGERS FROM HEADER */
/*****
numread=fread((void*)intbuff,4,3,hbin);
klay= intbuff[2];
/*****
/* READ HEAD VALUES */
/*****
numread=fread((void*)chbuff,4,2,hbin); /* read 2 characters prob. cntrl
chars */
numread=fread((void*)rbuff,4,(nrow*ncol),hbin); /* read head array */
numread=fread((void*)chbuff,4,1,hbin); /* read 1 characters prob. cntrl
chars */

if (klay==keylay)
{
for(k=0;k<(nrow*ncol);k++)
{
if (rbuff[k]== *hnoflo || rbuff[k]==*hdry) {rbuff[k]=-99.99;}
*headptr=rbuff[k];
headptr++;
}
return (0);
}
}
return(0);

}
/*****
/* END FUNCTION gethead */
/*****

```

hyper.aml

```

/* hyper.aml
/* Roy S. Sonenshein
/* US Geological Survey, Miami, Florida
/* coded 3/96

&type
&type HYDROPERIOD MODEL
&type
&type version 1.0 (3/96)
&type
&type This program uses the output from a MODFLOW model
&type to determine the HYDROPERIOD for an area within
&type the model using land elevation data.

&type
&type *****NOTE*****           To run this program,
&type the ARC/INFO TIN and GRID packages be available.
&type
&type This program uses menus to define the input requirements
&type needed to run the program. This requires a windows environment.
&type If you do not have windows, the programs will need to be run
&type from the command line. There are three programs that can be
&type run. These are:
&type
&type 1) modbin2arc - A C program that converts the MODFLOW binary
&type           head file into files that can be read into ARC/INFO.
&type 2) land - An ARC/INFO aml that converts land elevation
&type           point data into an ARC/INFO grid.
&type 3) hydroperiod - An ARC/INFO aml which uses the files created
&type           by the first two programs to create ARC/INFO grids
&type           and coverages which define the hydroperiod for
&type           the area of interest.
&type
&type Do you wish to run the programs using the menus?
&s answer [query '(Y/N) CR = No' .FALSE.]

&if %answer% = .FALSE. &then &stop
&else &do
  &type
  &s termtype [query 'Enter terminal type, default = 9999 for xterm' 9999]
  &term %termtype%
&end

&type
&type Do you wish to run the modbin2arc program
&type to create the water level data files?
&type (If NO, these files must already exist
&type to run the rest of the program)
&s answer [query '(Y/N) CR = No' .FALSE.]

&if %answer% = .TRUE. &then;
&do
  &menu modbin2arc.menu &stripe 'MODBIN2ARC MENU'
  &if %run% = 1 &then;

```

```

&do
  &r modbin2arc.aml %ROOT% %TIMESTEPS% %LAYER% %DRY% %XMIN% %YMAX%~
    [entryname %DELRC%] [entryname %HEADBIN%] %NUMLAYERS% %ROWS% %COLUMNS%~
    %STRESS% %NOFLOW%
  &type
  &type Data files have been created by modbin2arc
&end
&else; &type Program modbin2arc not executed
&type
&end

&type
&type Do you wish to run the AMLS to determine hydroperiod?
&s answer [query '(Y/N) CR = No' .FALSE.]

&if %answer% = .TRUE. &then;
&do
  &menu hyper.menu &stripe HYDROPERIOD_MENU
  &if %run% = 1 &then;
  &do
    &r land [entryname %pointfile%] %cellsize% %outpoint%
    &r hydroperiod %root% %layer% %cellsize% %outpoint% %first%~
      %last% %prevertime% %lsdist%
    &type
    &type HYDROPERIOD coverages and grids have been created
    &type
  &end
  &else; &type Hydroperiod amls not executed
  &type
&end

&type Program completed
&type

```

modbin2arc.aml

```

&args root tstps layer dry xmin ymax delrc hbin numlyrs rows cols stress noflow

/* modbin2arc.aml
/* This program is run by hyper.aml
/* to create the input file required for
/* the C program modbin2arc

/* Roy S. Sonenshein
/* US Geological Survey, Miami, Florida
/* coded 3/96

&if [filelist modbin2arc xx.files -file] < 1 &then
  &do
    &mess &popup
    &type The executable C file modbin2arc, required for this program
    &type is not in the current directory
    &type
    &mess &on
    &return
  &end
&s dele      = [delete xx.files -file]
&s fil       = %root%.%layer%.infile
&s dele      = [delete %fil% -file]
&s openfil   = [open %fil% openstat -WRITE]
&s writefil  = [write %openfil% %root%]
&s writefil  = [write %openfil% %tstps%]
&s writefil  = [write %openfil% %layer%]
&s writefil  = [write %openfil% %dry%]
&s writefil  = [write %openfil% %xmin%]
&s writefil  = [write %openfil% %ymax%]
&s writefil  = [write %openfil% [entryname %delrc%]]
&s writefil  = [write %openfil% [entryname %hbin%]]
&s writefil  = [write %openfil% %numlyrs%]
&s writefil  = [write %openfil% %rows%]
&s writefil  = [write %openfil% %cols%]
&s writefil  = [write %openfil% %stress%]
&s writefil  = [write %openfil% %noflow%]
&s closefil  = [close %openfil%]
&system ./modbin2arc %fil%
&mess &popup
&type The program modbin2arc has been run using the specified parameters
&type creating files in ARC/INFO createtin format.
&type The parameters you have entered are in file %fil%
&type This file can be used as the command line argument to run modbin2arc.
&type Run modbin2arc with no argument to see the format of the file.
&mess &on

```

land.aml

```

&args pointfile cellsize outpoint
&message &off
&type

/*          land.aml          */
/* last modified 2/24/95 by Roy S. Sonenshein */
/*          sunshine@usgs.gov */
/*          US Geological Survey */
/*          Miami, Fl          */

/* This program requires both the TIN and the GRID packages
/* of ARC/INFO

/* Creates a uniformly spaced point coverage from
/* a point data set of land elevation data. Also
/* creates of TIN data set of the land elevation data.
/* The point data set (pointfile) needs to be in the
/* createtin generate format
/* ID,x,y,z      ---- one line for each point
/* END          ---- last line in file

/* REQUIRED ARGUMENTS
/* pointfile = point data set file of land elevation data
/* cellsize = x and y dimension of cellsize to be used to create GRID
/* outpoint = output point coverage to contain land elevation data

/* outpoint_tn = output point tin created from raw land elevation data

/* check for arguments
&if [null %pointfile%] or [null %cellsize%] or [null %outpoint%] &then
    &goto usage

&s %outpoint% = [translate %outpoint%]

/* check for correct argument types
&if [exists %pointfile% -file] &then; &s dummy = 1
    &else &do
        &type
        &type FILE %pointfile% does not exist
        &goto usage
    &end

&if [type %cellsize%] ne -1 &then;
    &do
        &type
        &type CELL SIZE %cellsize% must be an integer
        &goto usage
    &end

&if [exists %outpoint% -cover] &then;
    &do
        &type
        &type COVER %outpoint% already exists

```

```

    &goto usage
&end

&if [exists %outpoint% -file] or [exists %outpoint% -directory] &then;
    &do
        &type
            &type %outpoint% already exists as a file or directory
            &goto usage
    &end

/* set working geodataset variables and delete existing working datasets
&s bndgrid = xxbnd_grd
&s bndpoly = xxbnd_poly
&s landdraw = %pointfile%_cov
&s landtin = %outpoint%_tn
&s landhull = xxland_hull
&S landtemp = xxland_cov

&if [exists %bndgrid% -grid] or [exists %bndgrid% -cover] &then kill %bndgrid%
&if [exists %bndpoly% -cover] or [exists %bndpoly% -grid] &then kill %bndpoly%
&if [exists %landdraw% -cover] &then kill %landdraw%
&if [exists %landtin% -tin] or [exists %landtin% -cover] &then kill %landtin%
&if [exists %landhull% -cover] &then kill %landhull%
&if [exists %landtemp% -cover] &then kill %landtemp%

&if [exists %bndgrid% -directory] or [exists %bndpoly% -directory] ~
    or [exists %landtin% -directory] or [exists %landhull% -directory]~
    or [exists %landtemp% -directory] or [exists %bndgrid% -file]~
    or [exists %bndpoly% -file] or [exists %landtin% -file]~
    or [exists %landdraw% -directory] or [exists %landdraw% -file]~
    or [exists %landhull% -file] or [exists %landtemp% -file] &then
    &do
        &type
            &type One or more of the following:
            &type %bndgrid%, %bndpoly%, %landtin%, or %landhull%
            &type exist as a system directory or file
            &type These are required as temporary directories
            &type or as output ARC/INFO data sets for this program.
            &type Please move, rename, or delete (KILL) these directories or files.
            &goto usage
    &end

/* create TIN from land elevation data points
/* need error routine to state that POINTFILE is in wrong format
&type Creating TIN [translate %landtin%] from %pointfile% data set
&type
&severity &error &routine tinfail
createtin %landtin%
generate %pointfile% point mass
end
&severity &error &fail
&describe %landtin%

/*create point data set from POINTFILE

```

```

&type Creating raw POINT coverage [translate %landraw%]
&type from %pointfile% data set
&type
generate %landraw%
input %pointfile%
points
quit

/* enter GRID subsystem and set window and cellsize environment variables
&type Creating point coverage [translate %outpoint%]
&type with cell size of %cellsize%
&type
GRID
SETCELL %cellsize%
SETWINDOW %tin$xmin% %tin$ymin% %tin$xmax% %tin$ymax%

/* create boundary grid with a unique value for each grid cell
&if [exists xxgrid1_grd -grid] &then kill xxgrid1_grd
xxgrid1_grd = 1
&describe xxgrid1_grd
%bndgrid% = $$ROWMAP * %grd$ncols% + $$COLMAP
&if [exists xxgrid1_grd -grid] &then kill xxgrid1_grd

/* create polygon coverage of grid cells
%bndpoly% = GRIDPOLY (%bndgrid%)

QUIT

&if [exists %bndgrid% -grid] &then kill %bndgrid%

/* create point coverage from nodes of boundary polygon coverage
/* NOTE: The four corners will not have a point, all other cell
/*      corners will be converted to a point
NODEPOINT %bndpoly% %landtemp%
BUILD %landtemp% POINT

/* limit final land elevation coverage to those points
/* that fall within the boundaries of the land elevation TIN
TINARC %landtin% %landhull% HULL
CLIP %landtemp% %landhull% %outpoint% POINT

/* determine land elevation for each point
TINSPOT %landtin% %outpoint% LAND

/*cleanup
&label cleanup
&if [exists %bndgrid% -grid] &then kill %bndgrid%
&if [exists %bndpoly% -cover] &then kill %bndpoly%
&if [exists %landhull% -cover] &then kill %landhull%
&if [exists %landtemp% -cover] &then kill %landtemp%
&severity &error &fail
&message &on
&return &inform

```

```

&label usage
&type
&type Usage: LAND <POINT FILE> <CELL SIZE> <OUTPUT POINT COVER>
&type           Requires both the TIN and GRID packages
&type

&messages &on
&return

/* error routine for problem with createtin - most likely
/* due to a bad point file
&routine tinfail

&type
&type An error has occurred in the createtin subroutine
&type The error is most likely due to an improper
&type input point file. Please check the file to verify
&type the file is in the correct format
&type
&type ID,x,y,z      ---- one line for each point
&type END          ---- last line in file
&type
quit
&goto cleanup

&routine fail
&type
&type An unspecified error has occurred in the program
&goto cleanup

```

hydroperiod.aml

```
&args root layer cellsize outpoint first last prevtime lsdist
&severity &error &routine fail
```

```
/*          hydroperiod.aml          */
/* last modified 2/27/96 by Roy S. Sonenshein */
/*          sunshine@usgs.gov          */
/*          US Geological Survey        */
/*          Miami, Fl                   */
```

```
/* This aml is run after land.aml - which was used to create a
/* point coverage (outpoint) of land elevation data in the area of interest.
/* This aml creates water elevation surfaces (TIN's) from point
/* water level data. These surfaces, along with a land elevation surface,
/* are then converted to gridded data which are subtracted to determine
/* the hydroperiod for each grid cell.
```

```
/* This program requires both the TIN and the GRID packages
/* of ARC/INFO
```

```
/* The point data sets need to be named in the format
```

```
/* root_layer_stress_timestep_totaltime
/* (where totaltime is the elapsed time from the start of the model run)
```

```
/* and need to be in the createtin generate format
```

```
/* ID,x,y,z      ---- one line for each point
/* END           ---- last line in file
```

```
/* REQUIRED INPUT ARGUMENTS
```

```
/*
/* root = root filename of point data set files containing water level data
/* layer = model layer number
/* cellsize = x and y dimension of cellsize used to create GRID in land.aml
/* outpoint = output point coverage created in land.aml
/* first = first stress period to use
/* last = last stress period to use
/* lsdist = specified distance below land surface considered wetland
/* prevtime = starting time for first stress period
```

```
/* check for all required arguments entered
&message &off
```

```
&if [null %prevtime%] &then &goto usage
```

```
/* check for correct argument types
&if [filelist %root%_* xx.files -file] < 1 &then;
&do
```

```
    &type
    &type There are no files with the root name %root%_
    &type These files, in createtin generate point format,
    &type are needed to run this program. These files are point
    &type data sets of water levels for different time steps
    &type and different stress periods for a given MODFLOW (or
```

```

    &type similar model) layer and need to be named in the format
    &type rootname_modellayer_stressperiod_timestep_totalltime
    &goto usage
&end

&if [filelist %root%_%layer%_* xx.files -file] < 1 &then;
&do
    &type
    &type There are no files with the root name %root%_%layer%_
    &type Verify that the correct model layer was entered
    &goto usage
&end

&if [type %cellsize%] ne -1 &then;
&do
    &type
    &type CELL SIZE %cellsize% must be an integer
    &goto usage
&end

&if [exists %outpoint% -cover] &then; &s dummy = 1
&else; &do
    &type
    &type COVER %outpoint% does not exist
    &goto usage
&end

&if [type %first%] ne -1 or [type %last%] ne -1 &then;
&do
    &type
    &type STRESS PERIODS %first% and %last% must be integers
    &goto usage
&end

&if [type %prevtime%] > 0 &then;
&do
    &type
    &type START TIME OF FIRST STRESS PERIOD %prevtime%
    &type must be numeric (real or integer)
    &goto usage
&end

&if [null %lsdist%] &then &s lsdist = 0
&if [type %lsdist%] > 0 &then;
&do
    &type
    &type LAND SURFACE SENSITIVITY TO HYDROPERIOD %lsdist%
    &type must be numeric (real or integer)
    &goto usage
&end

&if %last% - %first% < 0 &then;
&do
    &type

```

```

    &type The last stress period (%last%) must be less than or
    &type equal to the first stress period (%first%).
    &goto usage
&end

&if [filelist %root%_%layer%_%first%_* xx.files -file] < 1 &then;
&do
    &type
    &type There are no files with the root name %root%_%layer%_%first%_
    &type Verify that the correct starting stress period was entered
    &goto usage
&end

&if [filelist %root%_%layer%_%last%_* xx.files -file] < 1 &then;
&do
    &type
    &type There are no files with the root name %root%_%layer%_%last%_
    &type Verify that the correct ending stress period was entered
    &goto usage
&end

&if [filelist %root%_%layer%_%first%_*_* xx.files -file] < 1 &then;
&do
    &type
    &type Verify that the correct file name format was used.
    &type
    &type rootname_modellayer_stressperiod_timestep_totalltime
    &type
    &type These input files, in createtin generate point format,
    &type are needed to run this program. These files are point
    &type data sets of water levels for different time steps
    &type and different stress periods for a given MODFLOW (or
    &type similar model) layer and need to be named in the format
    &type specified above.
    &type
    &goto usage
&end

/* Create one coverage (%wlcov%) for each stress period.
/* The coverage is a copy of the final land elevation "grid" coverage.
/* The PAT file in each coverage will contain an attribute
/* of the water level for each time step created by TINSPOT.
/* An attribute is calculated by comparing the water level
/* to the land surface elevation to determine the hydroperiod
/* for that point.

/* first DO LOOP cycles through stress periods from %first% to %last%
/* second DO LOOP cycles through all timesteps within each stress period

&do stress = %first% &to %last% &by 1
    &type PROCESSING STRESS PERIOD %stress%
    &s rootname = %root%_%layer%_%stress%
    &s wlcov = %root%%stress%_cov
    &if [exists %wlcov% -cover] &then kill %wlcov%

```

```

copy %outpoint% %wlcov%
&s filenum = [filelist %rootname%_* xx.files -file]
&s opennum = [open xx.files openstat -READ]
&if %openstat% ne 0 &then
  &do
    &type Problem opening temporary file of list of timestep files
    &type
    &s maxtimestep = 0
    &goto cleanup
  &end
&do timestep = 1 &to %filenum% &by 1
  &s readfil = [read %opennum% readstat]
  &type PROCESSING %readfil% for TIMESTEP %timestep%
  &s currtime = [after %readfil% %rootname%_%timestep%_]
  &if [type %currtime%] >= 0 &then;
    &do
      &type The file %readfil% is not in the proper format.
      &type The last part (timestep length) must be a number.
      &goto cleanup
    &end
    &s timelength%timestep% = %currtime% - %prevtime%
    &s prevtime = %currtime%
    &type Length of timestep %timestep% = [value timelength%timestep%]

/* create a TIN for each timestep
/* determine for each timestep the water level at each grid cell (TINSPOT)
/* and then determine whether the water level is above or below a
/* specified distance of landsurface

/* create TIN from water level elevation data sets
/* error routine (TINFAIL) states that POINTFILE is in wrong format

  &severity &error &routine tinfail
  &s wltin = xx%timestep%_tn
  &if [exists %wltin% -tin] &then kill %wltin%
  createtin %wltin%
  generate %readfil% point mass
  end
  &severity &error &routine fail

/* determine land elevation for each point
  TINSPOT %wltin% %wlcov% timestep%timestep% linear
  &if [exists %wltin% -tin] &then kill %wltin%
&end

/* use TABLES to determine a value for each timestep
/* 0 for water level greater than the user specified distance
/*      below land surface
/* length of timestep for water level <= the specified distance
/*      below land surface
/* dry and no flow cells need to be set to a low enough value
/* (-99.99 was used in the modbin2arc program) resulting in a
/* value of 0 being assigned to those cells

```

```

&s wlinfo = [translate %wlcov%.PAT]
&type Calculating land surface - water level differences
TABLES
&do timestep = 1 &to %filenum% &by 1
  ADDITEM %wlinfo% diff%timestep% 8 8 F 2
  SELECT %wlinfo%
  ASELECT
  RESEL TIMESTEP%timestep% ge LAND - %lsdist%
  CALC diff%timestep% = [value timelength%timestep%]
  SELECT
&end
Q STOP
&end

/* Define coordinates of window for GRID operations
/* when setting window, offset the origin to the left and down
/* by 1/2 the cell size so the data points fall in the center
/* of each cell

&describe %root%%first%_cov
&s corr = %cellsize% / 2
&s xmin = %dsc$xmin% - %corr%
&s ymin = %dsc$ymin% - %corr%
&s xmax = %dsc$xmax% + %corr%
&s ymax = %dsc$ymax% + %corr%

/* enter GRID subsystem and set window and cellsize environment variables

GRID
SETCELL %cellsize%
SETWINDOW %xmin% %ymin% %xmax% %ymax%

/* first DO LOOP cycles through stress periods from %first% to %last%
/* imbedded DO LOOPS cycle through all timesteps within each stress period

&type CREATING GRID FOR EACH STRESS PERIOD
&do stress = %first% &to %last% &by 1
  &s rootname = %root%_%layer%_%stress%
  &s filenum = [filelist %rootname%_* xx.files -file]
  &type Processing STRESS PERIOD %stress%
  &type to create [translate %root%%stress%_GRD]
  &s wlcov = %root%%stress%_cov

/* Create temporary grid for each timestep using the hydroperiod
/* value to set the value for each grid cell

&do timestep = 1 &to %filenum% &by 1
  &type Processing TIME STEP %timestep%
  &if [exists xxgrid%timestep%_grid -grid] &then kill xxgrid%timestep%_grid
  xxgrid%timestep%_grid = pointgrid (%wlcov%, diff%timestep%)
&end

/* Combine all grids for each stress period into one grid
/* summing the hydroperiods for each cell

```

```

/* xx1 is a working grid stored in memory only
/* %root%%stress%_grd is the final grid for each stress period
/* and is stored as an integer grid. The int() function concatenates
/* the final grid hydroperiod values for each cell.

&if [exists %root%%stress%_grd -grid] &then kill %root%%stress%_grd
docell
xx1 := 0
&do timestep = 1 &to %filenum% &by 1
  xx1 := xx1 + xxgrid%timestep%_grd
&end
&type Creating [translate %root%%stress%_grd]
%root%%stress%_grd = int(xx1)
end

/* Kill temporary grids

&do timestep = 1 &to %filenum% &by 1
  &if [exists xxgrid%timestep%_grd -grid] &then kill xxgrid%timestep%_grd
&end

/* Determine the total area for each "HYDROPERIOD" for the stress period
/* and store in an INFO file

&type Total area for each "HYDROPERIOD" is being stored
&type in INFO file [translate %root%%stress%_GEO]
&if [exists %root%%stress%_GEO -info] &then
  &s dele [delete %root%%stress%_GEO -info]
  %root%%stress%_GEO = zonalgeometry(%root%%stress%_grd,AREA)
&end

/* if more than one stress period, combine all stress periods for a
/* total grid

&if %last% > %first% &then
  &do
    &if [exists %root%_grd -grid] &then kill %root%_grd
    &type Creating [translate %root%_GRD]
    &type combined grid for stress periods %first% to %last%
    docell
    xx1 := 0
    &do stress = %first% &to %last% &by 1
      xx1 := xx1 + %root%%stress%_grd
    &end
    %root%_grd = int(xx1)
  end

  &type Total area for each "HYDROPERIOD" is being stored
  &type in INFO file %root%_GEO
  &if [exists %root%_GEO -info] &then
    &s dele [delete %root%_GEO -info]
    %root%_GEO = zonalgeometry(%root%_grd,AREA)
  &end

```

```

/* quit GRID subsystem

QUIT

/*cleanup
&label cleanup
&do timestep = 1 &to %filenum% &by 1
  &s wltin = xx%timestep%_tn
  &if [exists %wltin% -tin] &then kill %wltin%
  &s dele = [delete xx.files -file]
&end
&severity &error &fail
&messages &on
&return &inform

&label usage
&type
&type Usage: HYDROPERIOD <ROOT NAME> <LAYER> <CELL SIZE> <INPUT POINT COVER>
&type           <FIRST STRESS PERIOD> <LAST STRESS PERIOD>
&type           <START TIME OF FIRST STRESS PERIOD>
&type           <LAND SURFACE SENSITIVITY TO HYDROPERIOD>
&type           Requires both the TIN and GRID packages
&type

&messages &on
&return

/* error routine for problem with createtin - most likely
/* due to a bad point file
&routine tinfail

&type
&type An error has occurred in the createtin subroutine
&type The error is most likely due to an improper
&type input point file. Please check the file to verify
&type the file is in the correct format
&type
&type ID,x,y,z      ---- one line for each point
&type END          ---- last line in file
&type
quit
&messages &on
&goto cleanup

&routine fail
&type
&type An unspecified error has occurred in the program
&goto cleanup

```

modbin2arc.menu

7 modbin2arc.menu

PROGRAM (MODBIN2ARC) TO CREATE FILES IN ARC/INFO CREATETIN FORMAT
ONE FILE IS CREATED FOR EACH TIME-STEP FOR THE MODEL LAYER SPECIFIED
WITH THE NAMING CONVENTION

ROOT-NAME_LAYER_STRESS-PERIOD_TIME-STEP_TIME-AT-END-OF-TIME-STEP

(Use right mouse button for help on each entry field)

Select MODFLOW binary head data file ^Select MODFLOW file of row and column
spacings
by clicking on file list below ^by clicking on file list below
%datalist1 %datalist2

Enter ROOT file name for %input1 Enter the number of:
output files ^layers %input6
Enter the head value used for: ^rows %input7
Dry cells %input2 columns %input8
No flow cells %input3 stress periods %input9

Enter coordinates of upper left ^Enter number of time steps%input10
corner of model grid: ^per stress period
minimum x-coordinate %input4 Enter model layer number %input11
maximum y-coordinate %input5 to process

%button1 %button2

%datalist1 INPUT HEADBIN 32 TYPEIN NO SCROLL YES ROWS 4 ~
REQUIRED ~
NEXT %input1 ~
QUERY '&mess &popup;&type MODFLOW binary head file;&mess &on' ~
FILE ~
* -FILE -SORT

%datalist2 INPUT DELRC 32 TYPEIN NO SCROLL YES ROWS 4 ~
REQUIRED ~
NEXT %input1 ~
QUERY '&mess &popup;&type MODFLOW row and column spacings file;&mess &on' ~
FILE ~
* -FILE -SORT

%input1 INPUT ROOT 11 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input2 ~
CHARACTER

%input2 INPUT DRY 11 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input3 ~
REAL

%input3 INPUT NOFLOW 11 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input4 ~

```

REAL
%input4 INPUT XMIN 11 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input5 ~
REAL
%input5 INPUT YMAX 11 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input6 ~
REAL
%input6 INPUT NUMLAYERS 10 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input7 ~
RANGE 1 1000 ~
INTEGER
%input7 INPUT ROWS 10 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input8 ~
RANGE 1 1000 ~
INTEGER
%input8 INPUT COLUMNS 10 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input9 ~
RANGE 1 1000 ~
INTEGER
%input9 INPUT STRESS 10 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input10 ~
RANGE 1 1000 ~
INTEGER
%input10 INPUT TIMESTEPS 11 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input11 ~
RANGE 1 1000 ~
INTEGER
%input11 INPUT LAYER 11 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input1 ~
RANGE 1 1000 ~
INTEGER
%button1 BUTTON KEEP 'Execute modbin2arc program' ~
&s run = 1; &return
%button2 BUTTON KEEP RETURN 'Exit menu without executing program'~
&s run = 0;&return
%FORMOPT STARTFIELD %input1 SETVARIABLES RETURN MESSAGEVARIABLE msg

```

hyper.menu

7 hyper.menu

HYDROPERIOD MODEL (HYPER) USING ARC/INFO TIN and GRID PACKAGES
 (Use right mouse button for help on each entry field)

Select raw elevation data file %datalist1
 by clicking on file list----->

(Use this file list to search
 ^for MODFLOW water level files
 named in the following manner:
 ROOT_LAYER_STRESS_TIMESTEP_TIME

Enter cellsize %input1

Enter ROOT file name for %input4
 MODFLOW water level data files

Enter output cover %input2
 for land elevation %datalist2
 Do not use existing directory
 See list at right----->

Enter model layer number %input5
 ^Enter stress periods ^FIRST ^LAST
 to process %inp7 %input8

Enter hydroperiod %input3
 sensitivity to
 land surface
 Down is positive

Enter starting model time %input9
 ^for first stress period
 ^being processed

%button1

%button2

%datalist1 INPUT POINTFILE 32 TYPEIN NO SCROLL YES ROWS 4 ~

REQUIRED ~

NEXT %input1 ~

QUERY '&mess &popup;&type File in createtin input format (1,x,y,z) - one per line;~
 &type Followed by END on last line;~
 &mess &on' ~

FILE ~

* -FILE -SORT

%datalist2 INPUT DATALIST2 14 TYPEIN NO SCROLL YES ROWS 4 ~

QUERY '&mess &popup;~

&type No data entry is associated with this list;~

&mess &on' ~

FILE ~

* -DIRECTORY -SORT

%input1 INPUT CELLSIZE 11 TYPEIN YES SCROLL NO ~

REQUIRED ~

NEXT %input2 ~

QUERY '&mess &popup;~

&type Integer - Spacing of data points to use in creating GRID of land elevation data;~

&mess &on' ~

RANGE 1 10000 ~

INTEGER

%input2 INPUT OUTPOINT 11 TYPEIN YES SCROLL NO SIZE 12 ~

REQUIRED ~

NEXT %input3 ~

QUERY '&mess &popup;~

&type Name of ARC/INFO point cover to be created from land elevation data;~

&mess &on' ~

CHARACTER

%input3 INPUT LSDIST 11 TYPEIN YES SCROLL NO ~

REQUIRED ~

NEXT %input4 ~

QUERY '&mess &popup;&type Water level distance below land surface considered a wetland;~

&type Depth below land surface is positive, above land surface is negative;~

&type This value is the critical value used in determining hydroperiod;~

&mess &on' ~

RANGE -1 50 ~

REAL

%input4 INPUT ROOT 11 TYPEIN YES SCROLL NO ~

REQUIRED ~

NEXT %input5 ~

```

QUERY '&mess &popup;~
    &type Water level data file names in the format ROOT_LAYER_STRESS_TIMESTEP_TOTALTIME;~
    &type Water level data files in createtin input format (1,x,y,z);~
    &type - one per line followed by END on last line.;~
    &type These files can be created from the MODFLOW binary head files;~
    &type using the modbin2arc program.;~
    &mess &on' ~
CHARACTER
%input5 INPUT LAYER 11 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %inp7 ~
QUERY '&mess &popup;&type Integer - Modflow model layer number of interest;~
    &type See ROOT file name;~
    &mess &on' ~
RANGE 1 100 ~
INTEGER
%inp7 INPUT FIRST 4 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input8 ~
QUERY '&mess &popup;&type Integer - First stress period to process;~
    &type See ROOT file name;~
    &mess &on' ~
RANGE 1 1000 ~
INTEGER
%input8 INPUT LAST 4 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input9 ~
QUERY '&mess &popup;~
    &type Integer - Last stress period to process;~
    &type See ROOT file name;~
    &mess &on' ~
RANGE 1 1000 ~
INTEGER
%input9 INPUT PREVTIME 12 TYPEIN YES SCROLL NO ~
REQUIRED ~
NEXT %input1 ~
QUERY '&mess &popup;&type Real - Starting model time of;~
    &type first stress period (%first%) being processed;~
    &type See ROOT file name;~
    &mess &on' ~
RANGE 0 99999 ~
REAL
%button1 BUTTON KEEP 'Execute hydroperiod amls' ~
    &s run = 1; &return
%button2 BUTTON KEEP RETURN 'Exit without executing amls'~
    &s run = 0;&return
%FORMOPT STARTFIELD %input1 SETVARIABLES RETURN MESSAGEVARIABLE msg

```


REFERENCES CITED

Environmental Systems Research Institute, Inc., 1990, Understanding GIS, the ARC/INFO method: Redlands, CA, Environmental Systems Research Institute, Inc.

Maltby, E. and Dugan, P.J., 1994, Wetland ecosystem protection, management, and restoration: An international perspective, *in* Davis, S.M., and Ogden, J.C., eds.,

Everglades: The ecosystem and its restoration: Delray Beach, Florida, St. Lucie Press, p. 29-46.

McDonald, M.G., and Harbaugh, A.W., 1988, A modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 6, Chapter A1, 586 p.

Mitsch, W.J., and Gosselink, J.G., 1993, Wetlands: New York, Van Nostrand Reinhold, 722 p.