A.R.M. Loxahatchee National Wildlife Refuge

Refuge 39-Box Model

User's Manual

by
Hamid Bazgirhoob
Ehab A. Meselhe
Michael G. Waldon

Prepared for the U.S. Fish and Wildlife Service under a
cooperative-agreement with the University of Louisiana-Lafayette

July 2011
Revised May 2015

TABLE OF CONTENTS

1. Introduction

As restoration of the Arthur R. Marshall Loxahatchee National Wildlife Refuge (Refuge) continues, there is a need for a quantitative methodology for predicting impacts of proposed management changes on Refuge water quantity and quality. A series of compartment-based models have been developed for the hydrologic and water quality modeling of the Refuge (http://LoxModel.MWaldon.com). The first of these models, termed the Simple Refuge Screening Model (SRSM), simulates the water budget, regulation schedule implementation, and constituent dynamics for chloride (Cl), sulfate (SO4), and total phosphorus (TP). The SRSM uses two compartments to describe water volume and stage, and four compartments to simulate constituent mass and concentration. SRSM version 1.0 (Arceneaux *et al*, 2007; Meselhe *et al*., 2007a, 2007b) was implemented as a daily water budget using a Microsoft Excel workbook for water volume modeling, and the EPA WASP model (http://www.epa.gov/athens/wwqtsc/html/wasp.html) simulating TP. Versions 2 and 3 of the SRSM were transitional and are no longer available. SRSM version 4.0 (Meselhe *et al*, 2009) was developed using the commercial numerical differential equation solver package of Berkeley-Madonna (Macey, *et al*, 2000). It allows greater clarity in coding, supports shorter time steps obviating the need for the ad-hoc procedures of Version 1.0, and provides a well-documented user interface.

Wang *et al* (2008) applied cluster analysis of concentrations of chloride, total phosphorus, sulfate, and calcium measured at sites distributed throughout the Refuge to objectively determine the number of compartments and to spatially delineate these compartments with similar features within the refuge. That delineation led to a 9-Box (compartment) model structure (Wang, et al, 2012).

Consideration of management objectives and compatibility with possible future model scenarios then led to further disaggregation of the 9-Box model into the 39-Box model. The 39-Box Model defines twenty (28) marsh compartments and eleven (11) canal segments (Figure 1).

As with the SRSM and 9-Box model, the 39 Box model is based on a link-node conceptualization. Each node in a link-node conceptualization has the state variables of water volume, and mass for each constituent. Stage, depth, and constituent concentration are calculated from these state variables. Flow of water, and transport of constituent mass (mass flux) occurs between nodes through links. Each link has an upstream and downstream end which defines the direction of positive flow and mass transport. Links have properties of length and cross-sectional area, nodes have the property of surface area. In addition to advective or flow-related mass transport through links, mass transport is also modeled as dispersion at links which moves mass from the node with higher to the node with lower concentration.

Each of these models (the SRSM Version 1, SRSM version 4, 9-Box Model, and 39-Box Model) has specific advantages and limitations. The models were designed to be a complimentary suite. As such, the 39-Box Model is not intended to replace the other models, but instead to provide additional options for Refuge modeling applications.

This manual describes the 39-Box (compartment) model which was developed from the 9-Box model and the SRSM. It aims to provide users with an understanding of both the theory and implementation of the model. The objective is to give users the ability to accurately simulate various water management scenarios for the Refuge in order to gain a better understanding of this wetland system and its dynamics. This manual assumes that the reader is generally familiar with the Refuge location, hydrologic features, and water quality. For more information, users are directed to the Refuge Comprehensive Conservation Plan (USFWS 2000).

1.1. General Description of Model Structure

The refuge is compartmentalized into 39 compartments (also termed boxes, nodes, or cells), and each is classified as canal or marsh. Surface areas are assumed constant (Table 1), with marsh area totaling 4.03 million m^2 and canal area totaling 560.02 million m^2. The link-node diagram of the refuge is shown in Figure 2. As shown in Table 1 and Figure 2, compartments 1-11 represent the canal cells and compartments 12-39 represent the marsh cells.

The model simulates the mass and concentration of chloride (Cl), total phosphorus (TP), and sulfate (SO4). Note that Cl mass is measured as chlorine, and TP mass is measured as phosphorus, and not as phosphate. However, SO4 mass is measured as sulfate, and not as sulfur. This convention for SO4 maintains consistency with South Florida Water Management District laboratory reporting.

Compartments 1-11 represents the canal cells and compartments 12-39 represents the marsh cells. Exchange flow is calculated in the marsh based on compartment surface area ratios. For this purpose there is a Node factor defined for each exchange node between a canal and marsh cell.

There are water quality, meteorological, stage, and discharge monitoring stations in the refuge which measure the inflow and outflow, water level, precipitation, evapotranspiration and the simulated three constituent concentrations. Sites are located in both canal cells and marsh cells. However, some cells do not have any associated water quality monitoring sites, and most cells do not have an associated stage monitoring site. Figures 3, 4, 5 and 6 map the location of monitoring sites in the Loxahatchee Refuge.

Figure 1: This map delineates the model 39 Compartments. Also shown are stage gauges and inflow/outflow structures.

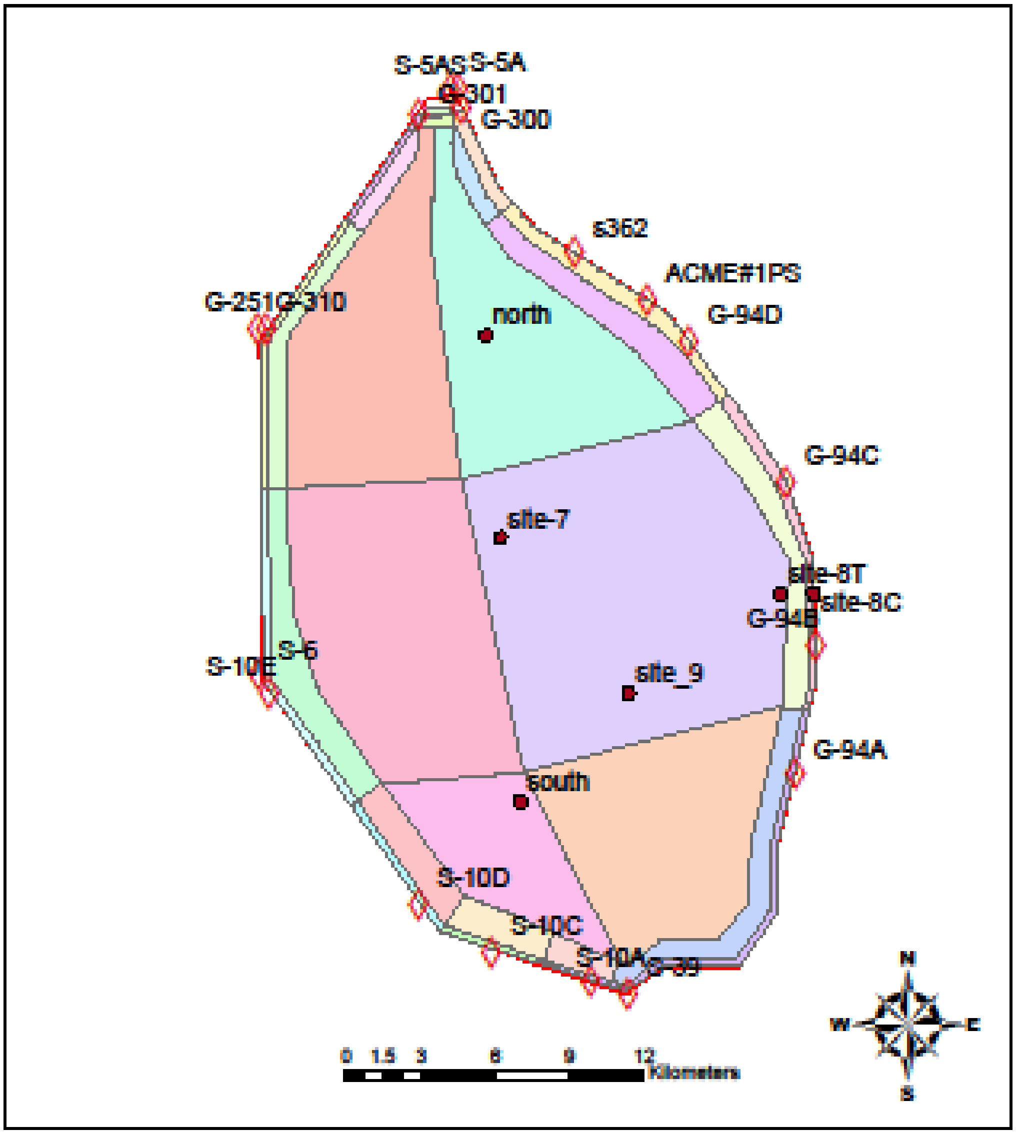


Figure 2: Link-node diagram of the 39-Compartment design

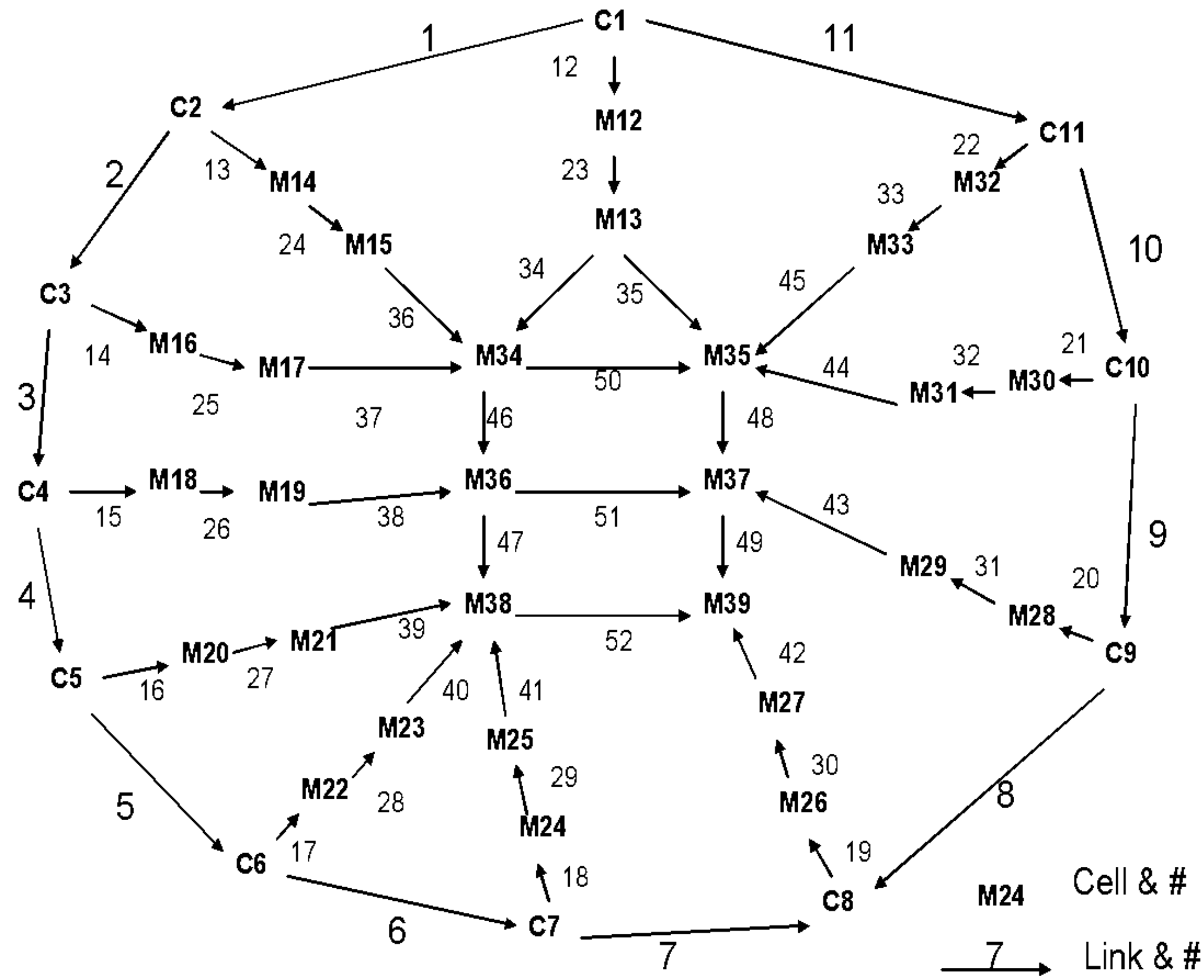


Figure 3: Location of rain gages used in model daily rainfall estimation.

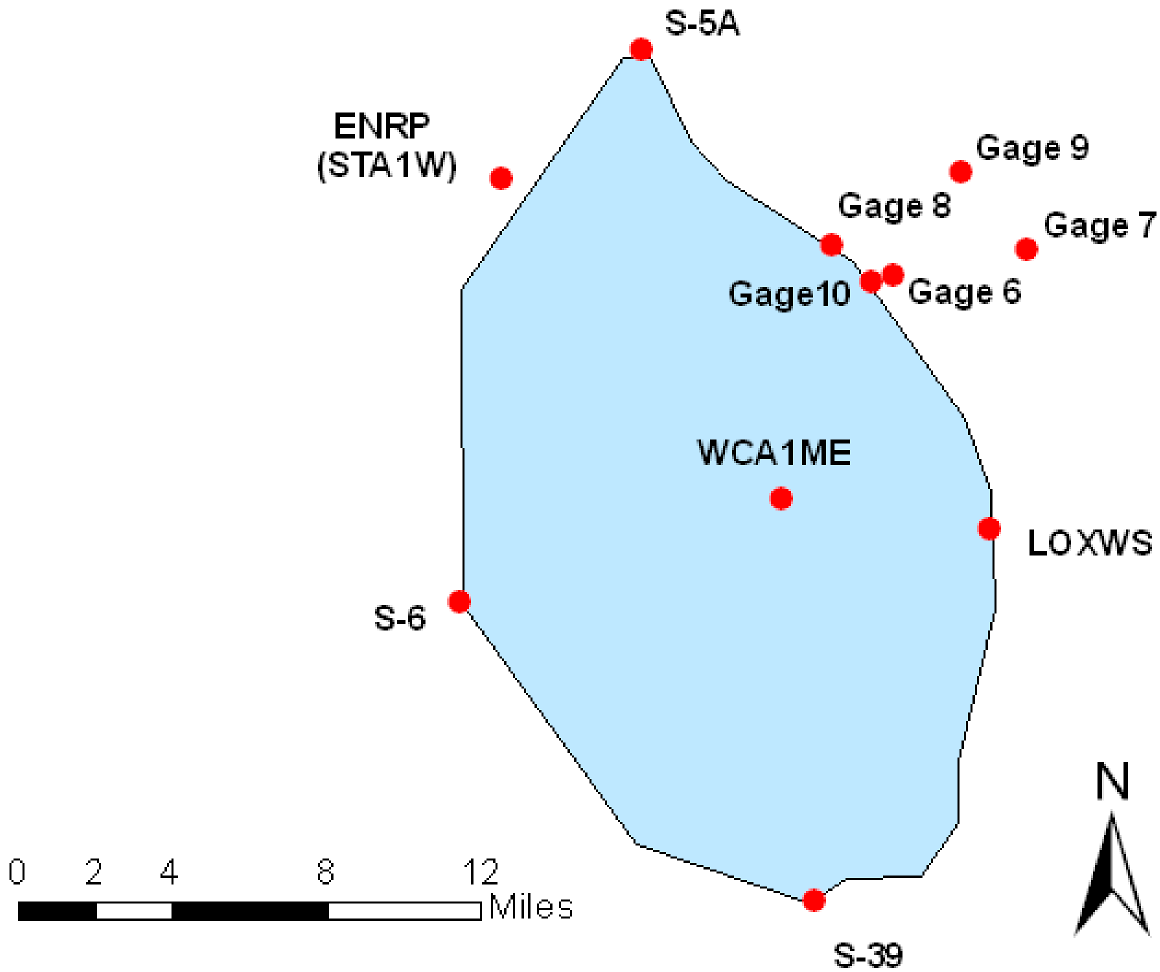


Figure 4: Hydraulic structures located in the Loxahatchee Refuge. Discharge at each structure is monitored. This figure is adapted from Meselhe *et al* (2005).

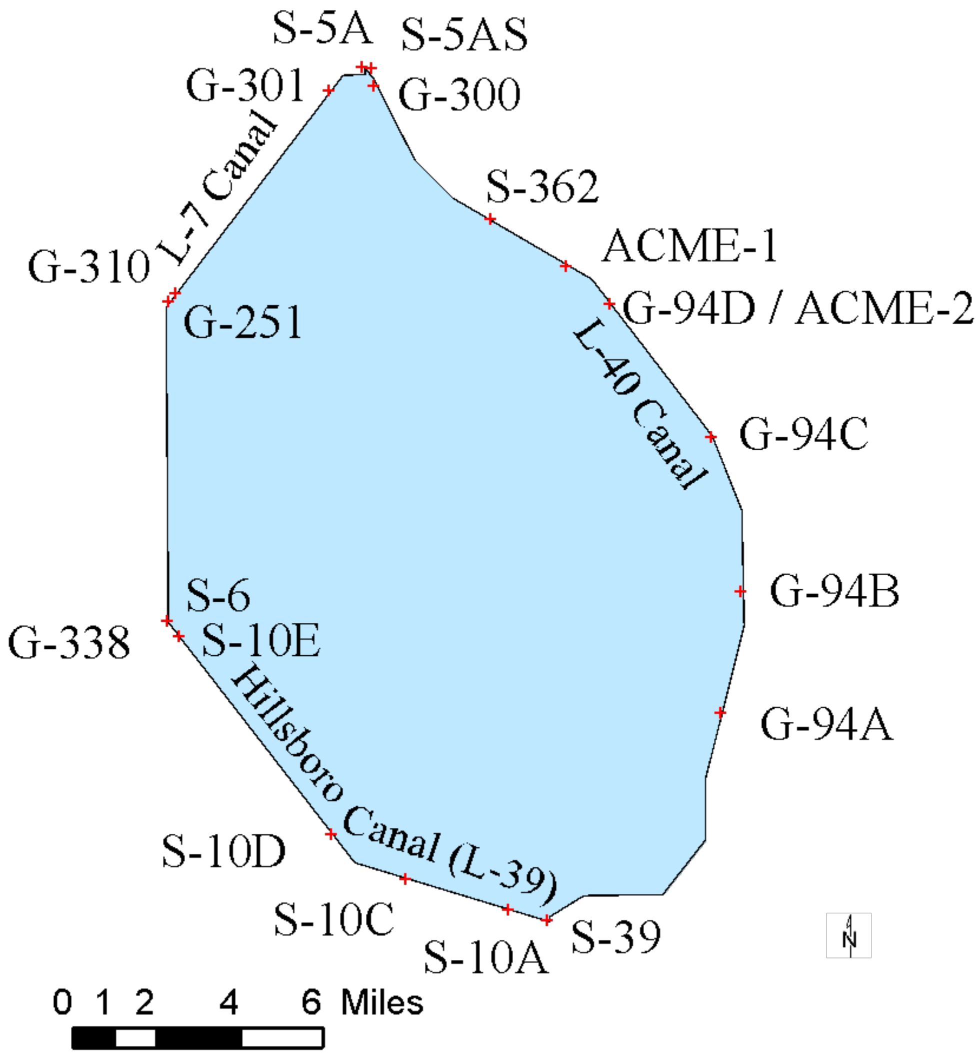


Figure 5: Water level gages located in the Loxahatchee Refuge

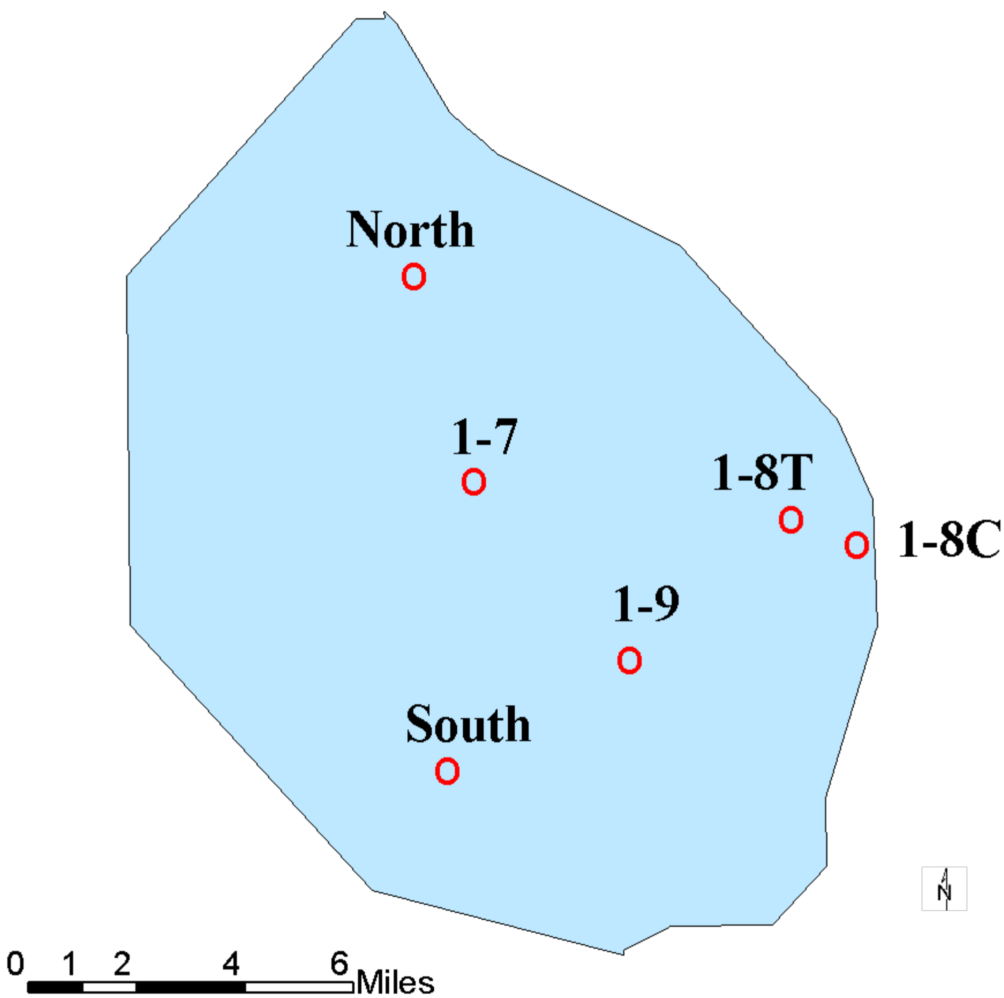


Figure 6: XYZ and EVPA water quality monitoring sites located inside the Loxahatchee Refuge.

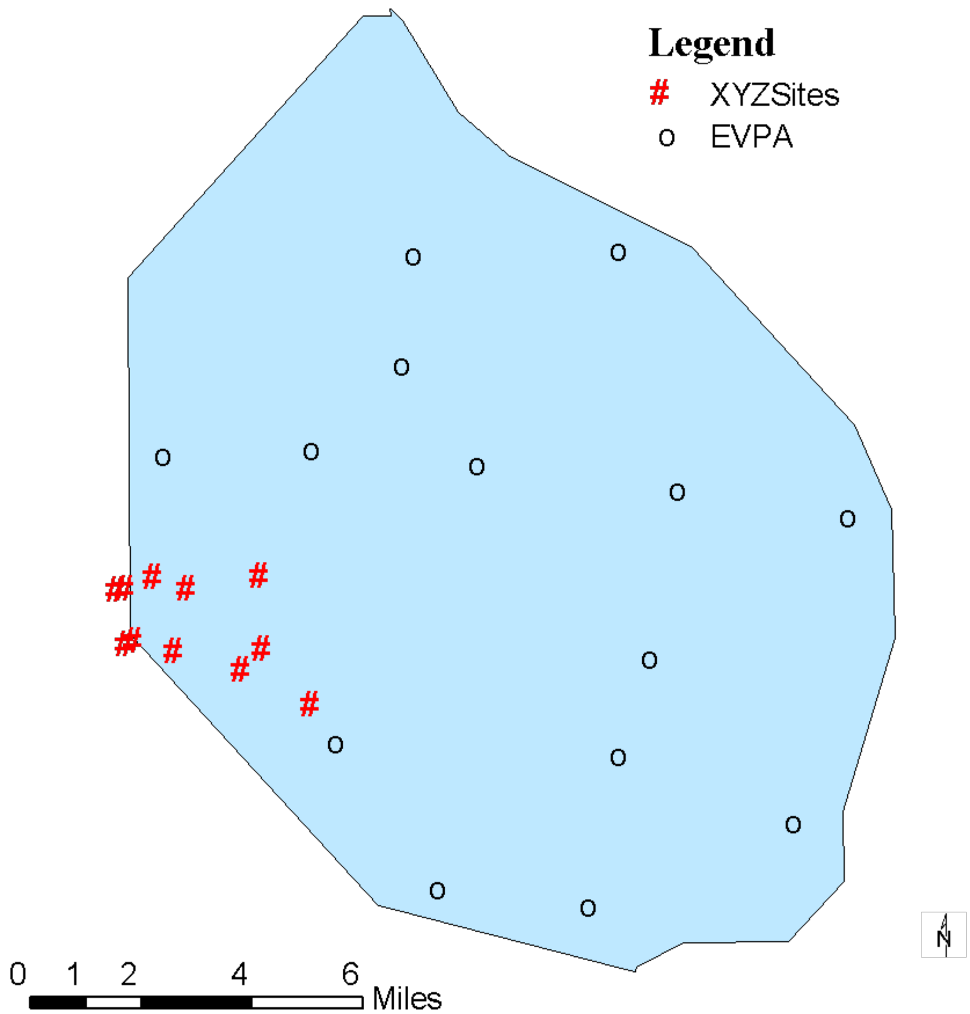


Table 1: Classification of each compartment and compartmental area.

Compartment Number	Description	Area (m^2)
1	Canal	67451.135821
2	Canal	432580.505184
3	Canal	1321081.928510
4	Canal	1070766.748980
5	Canal	343866.846945
6	Canal	242568.499782
7	Canal	290820.545859
8	Canal	1172416.460760
9	Canal	745969.419186
10	Canal	770845.383037
11	Canal	325422.412436
12	Marsh	1181402.531870
13	Marsh	2501726.141840
14	Marsh	293003.038514
15	Marsh	543785.352232
16	Marsh	3485961.830800
17	Marsh	4045182.150470
18	Marsh	3195752.157130
19	Marsh	7636805.725330
20	Marsh	7200836.398630
21	Marsh	13400480.995600
22	Marsh	4195293.495740
23	Marsh	14539203.085100
24	Marsh	1890978.078230
25	Marsh	7512971.274630
26	Marsh	988502.221609
27	Marsh	5248955.803790
28	Marsh	522323.800231
29	Marsh	2903428.104900
30	Marsh	4434932.932820
31	Marsh	12224954.711100
32	Marsh	4740640.266060
33	Marsh	12213974.991300
34	Marsh	61064630.228800
35	Marsh	62998466.731800
36	Marsh	87163883.382300
37	Marsh	135435126.208000
38	Marsh	31888461.517200
39	Marsh	63423036.384400

Table 2: Parameters

Parameter	Description	Unit	Value
lseep	canal seepage constant	1/day	0.042
rseep	marsh seepage constant	1/day	0.000131527
B1	transport coefficient between canal and marsh cells	1/m/day	8.7
B2	transport coefficient between marsh and marsh cells	1/m/day	2.3
B5	transport coefficient between canal and canal cells	1/m/day	6.5
ETmin	minimum ET reduction factor	-	0.2
Het	depth below which ET is reduced	m	0.25
evap	fraction of ET that is evaporation	-	0.65
kd	dispersion coefficient	m2/day	43200
KhalfSO4	sulfate half saturation constant	g/m3	1
KmaxSO4	sulfate maximum removal rate	g/m2/year	14.4
K1tp	total phosphorus maximum uptake rate	m3/g/year	0.1064 (EMG), 0.221 (PEW)
K2tp	total phosphorus recycle rate	m2/g/year	0.002 (EMG), 0.0042 (PEW)
K3tp	total phosphorus burial rate	1/year	0.3192 (EMG), 0.6631 (PEW)
interP	total phosphorus internal loading rate	mg/m2/year	0
wd_cl	wet deposition, chloride	mg/L	2
dd_cl	dry deposition, chloride	mg/m2/year	1136
wd_tp	wet deposition, total phosphorus	mg/L	0.01
dd_tp	dry deposition, total phosphorus	mg/m2/year	40
wd_so4	wet deposition, sulfate	mg/L	1
dd_so4	dry deposition, sulfate	mg/m2/year	138.2

1.1.2 **Equations & Simulations**

The 39-Box model simulates the mass and concentration of chloride (Cl), total phosphorus (TP), and sulfate (SO4). Note that TP mass is measured as phosphorus, not phosphate, and SO4 mass is measured as sulfate, not sulfur. Compartments 1-11 represents the canal cells and compartments 12-39 represents the marsh cells. Exchange flow is calculated in the marsh based on compartment surface area ratios. For this purpose there is a Node factor defined for each exchanging node between a canal and marsh cell.

Simulation of each constituent is based on a mass balance equation. The loading terms of the mass budget (*qnet*, *gload*, sload and *aload*) are similar for all three constituents. However, the reactive load term (*rload*) is uniquely structured for each constituent. Chloride is modeled as a conservative constituent with zero reactive load. Its mass is lost or gained solely through the transport of water into or out of the system. Total Phosphorus (TP) dynamics are approximated with equations adapted from those presented in the Dynamic Model for Stormwater Treatment Areas (DMSTA) developed by Walker and Kadlec (2011; see also http://wwwalker.net/dmsta/index.htm). Finally, sulfate (SO4) is simulated using a Monod relationship (SRSM v. 4.0 User's Manual by Meselhe, *et al*, 2009).

The following samples of model code illustrate the volume and mass balance differential equations defined in the model. The rate of canal volume increase (m3/day) in compartment 1 is calculated as

```
d/dt(vol[1])=(P - ET - Gc[1])*area[1] + (Qin[1] -Q[1]-Q[11] -Q[12] -QoutHistoric[1])
```

This equation illustrates the use of Berkeley Madonna subscript syntax. In Berkeley-Madonna, subscripts are enclosed in square brackets. Constituent differential equations are more complex. Constituent mass rate of change depends on loads from flow and dispersion between cells (qload and dload), loads associated with structure flow (sload), load from aerial deposition (aload), load from groundwater recharge (gload), and reactive load (rload) which can include both uptake and return to the water column. The following sample of code defines this mass balance. This code also illustrates the use of comments for program self-documentation. In Berkeley-Madonna, comments may be inserted using two alternatives. Any text on a line following a semicolon (;) is ignored by the compiler, and all text falling between braces (aka curly or squiggly brackets, {}) is ignored.

```
{Canal cells}
;  Canals --- structure load + flow (advective) load + dispersive load + aerial load -
groundwater load + reactive loads
d/dt(mass[1..nconstit, 1..nc]) = sload[i,j] + qdload[i, j] + aload[i, j] - gload[i, j] +
rload[i, j]

{marsh cells --- flow (advective) load + dispersive load + aerial load - groundwater load
for each constituent + reactive loads}
d/dt(mass[1..nconstit, (nc+1)..ncell]) =  qdload[i, j] + aload[i, j] - gload[i, j] +
rload[i, j]
```

The following code sample defines loads and also illustrates

Berkeley-Madonna implied loops which use the variables i, j, and k for the first second, and third implied subscripts.

```
;Dispersion loads:
dload[1..nconstit, 1..nc] = kd1*(exarea[j]/Radius[j])*(conc[i,up[j]]-conc[i,dn[j]])
dload[1..nconstit, (nc+1)..22] = kd2*(exarea[j]/Radius[j])*(conc[i,up[j]]-conc[i,dn[j]])
dload[1..nconstit, 23..nlinks] = kd3*(exarea[j]/Radius[j])*(conc[i,up[j]]-conc[i,dn[j]])

;Aerial deposition
aload[1..nconstit, 1..ncell] = area[ j]*((DD[i]/1000) + (precip[i]*P))

;Groundwater seepage + transpiration
gload[1..nconstit, (nc+1)..ncell] = conc[i, j]*(Gm[j]+(transp*ET))*area[ j]    ; marsh seepage + transp load
gload[1..nconstit, 1..nc]          =  conc[i, j]*Gc[j]*area[j]                  ; canal seepage load, no
transpiration in canal

;Net canal structure inflow - outflow loads
sload[1..nconstit,1..4]      = LOAD[i,j] - QoutHistoric[j]*conc[i, j]
sload[1..nconstit,5]      = LOAD[i,5] - Qout[5]*conc[i, 5]
sload[1..nconstit,6]      = LOAD[i,6] - Qout[6]*conc[i, 6]
sload[1..nconstit,7]      = LOAD[i,7] - Qout[7]*conc[i, 7]
sload[1..nconstit,8]      = LOAD[i,8] - Qout[8]*conc[i, 8]
sload[1..nconstit,9..nc]      = LOAD[i,j] - QoutHistoric[j]*conc[i, j]

;Reactive loads (losses)
rload[cl, 1..ncell] = 0       ; conservative

rload[so4, (nc+1)..ncell] = -(kso4[j]/depth[j])*max(mass[i, j],0)
rload[so4, 1..nc]          = 0

rload[so4eco, (nc+1)..ncell] = -(MaxSO4Removal*area[j]) * conc[i, j]/(khalfSO4+conc[i, j])    ;g/day
rload[so4eco, 1..nc]          = 0

rload[dmsta_tp, (nc+1)..ncell] = Release[i, j] - Uptake[i, j]

interP = 0.0

rload[dmsta_tp, 1..nc]  =  interP*area[j]
```

Where:

- *i* = constituent,
- *j* = compartment number (1-4),
- *k* = DMSTA calibration set (Phosphorus only),
- *M* = mass (g),
- *t* = time (days),
- *qnet* = net mass flow in surface water (g/day),
- *gload* = loss to groundwater seepage and evapotranspiration (g/day),
- *aload* = gain from wet and dry deposition (g/day), and
- *rload* = loss to storage uptake/release (TP) or reaction (SO4) (g/day).
- sload = Net canal structure inflow - outflow loads

1.2. Model Platform – Berkeley Madonna

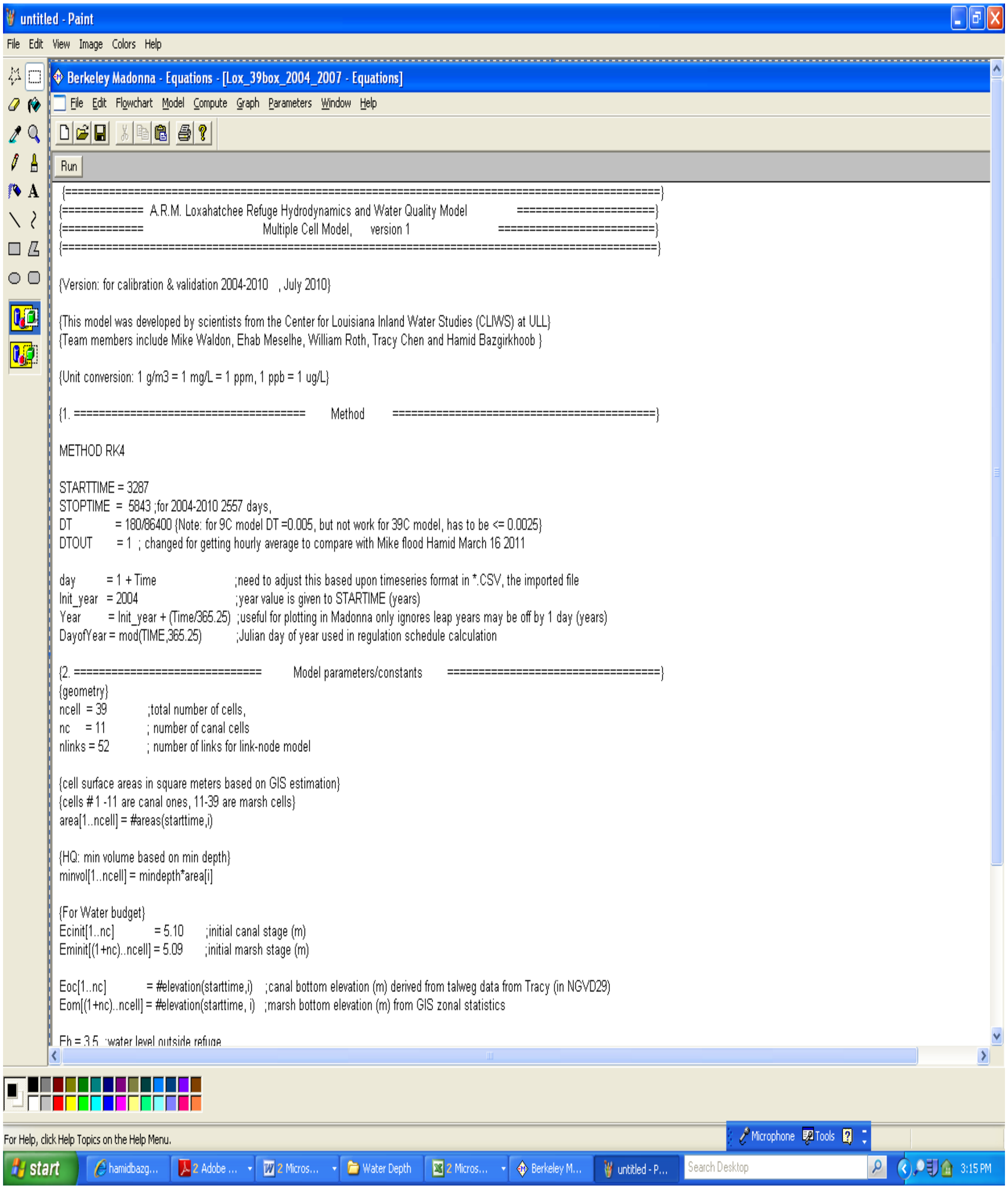
The model is implemented using the differential equations solver Berkeley Madonna version 8.3.9, which is a proprietary software developed by Robert I. Macey and George F. Oster. Berkeley-Madonna has several advantages. Compared with the STELLA program which was first applied in transition from Version 1 of the SRSM, Berkeley-Madonna is less expensive, faster, and accommodates much larger models. Both platforms are user-friendly and employ graphical user interfaces.

The Berkeley-Madonna program which is the backdrop for the code of the SRSM, 9-Box, and 39-Box models, has a number of useful built-in functions and operations. The Berkeley-Madonna user interface consists of a number of windows which may be opened, moved, resized, minimized, or closed by the user at any time. Model components and processes are user-defined in the **Equations** window; the optional Berkeley-Madonna Flowchart window was not used in the 39-Box model development. That is in Berkeley-Madonna terminology, the model is a "plain-text" rather than a "visual" model (Macey, *et al*, 2000;Meselhe, *et al*, 2009). Figure 7 shows the general format of the Berkeley-Madonna desktop.

Berkeley-Madonna has many features which improve ease of use and allow rapid coding and testing of alternatives. However, Berkeley-Madonna also has several disadvantages. At all times, users have full access to the model equations and parameter values. This is often convenient, but there is no executable module which shields the inner workings of the model, and this makes the model susceptible to inadvertent revision of the model and to subsequent errors. Users must be vigilant and take care not to corrupt the model code or parameterization. The requirement that the model always run within the Berkeley-Madonna system also makes application of alternative optimization codes or automated simulation outside of the Berkeley-Madonna interface impossible. Berkeley-Madonna also has no run-time capability to read or write data in external files. Berkeley-Madonna also lacks the capability of dealing with date or time formats, and as a consequence, all time in the 39-Box model is simply measured in days after January 1, 1995. Finally, the Berkeley-Madonna editor and compiler do not support macros or subroutines. Because of this deficiency, code in the equation window may be highly repetitive, prone to inadvertent error, and makes the code less readable.

No practical workarounds for these system deficiencies were identified by the model team. One alternative which was rejected as impractical is to copy/paste text between the equation window and a more advanced editor that supports macros. However, this alternative was rejected because it would be awkward and be a potential source of error. Implementing the model code in the Modelica simulation language (https://www.modelica.org/) was also considered, but this too was rejected because constraints of time and effort.

Figure 7: The general format of the Berkeley-Madonna desktop



Berkeley-Madonna also has the capability to perform optimizations, curve-fitting, and sensitivity analyses. For a comprehensive description of all pre-programmed functions, users of the model are encouraged to download the Berkeley-Madonna user's guide from www.berkeleymadonna.com. Additionally, users may download a demo version of this software from the Berkeley-Madonna web site and run models. However, while the demo version of the Berkeley-Madonna program allows users to modify and run models, the demo version does not allow the user to save model files or output of any kind.

1.3. User's Manual Objectives

This manual presents the pertinent information required for users to understand the components of the model (i.e., imported data, model equation format, and post-processing methods). Ultimately, users should use this document as a companion for the model to assure accurate execution and interpretation.

1.4. Caveats

1.4.1. If unfamiliar with Berkeley-Madonna, model users are strongly urged to consult the user's guide before attempting to run or manipulate any of the model components.

1.4.2. All parameter values represent those used to accurately validate and calibrate this model; discretion should be used when altering these values.

1.4.3. This model is set up to simulate a 16-year (1995-2010) period. All time series data needed to successfully run the model are stored within the model file. This manual provides the user with the background and understanding needed to revise this model simulation for other user-selected time periods or scenarios. Should the user want to simulate another time period or alternative conditions, the proper data must be obtained, properly formatted, and imported into the model.

1.4.4. Because of the level of spatial aggregation in the model, the model is not appropriate for applications that involve site-specific events. All results should be considered as spatial average values for the area of study.

1.4.5. This document offers a brief summary of model theory and equations. Users should consult the referenced documentation, the model code, and the companion manuscript for more in-depth descriptions of equations and calibration parameter values.

2. Data Preparation

This section describes how to import the necessary time series text files. The 7 separate data input files are outlined in Table 1.

The user need to prepare the Chloride, Inflow, Outflow, PET (precipitation and evapotranspiration), Regulation (hurricanes and hydraulic- hydrologic events), SO4 and TP observed data in files, in the text format for the time period of run; with the day as the first column.

Table 3: Model input files

Filename (alphabetical)	**Data Vectors**	**Summary**
CL.txt	14	Chloride concentration values (mg/L)
INFLOW.txt	20	Historic inflow values (m3/d)
OUTFLOW.txt	20	Historic outflow values (m3/d)
PET.txt	3	Precipitation and Evapotranspiration (m/d)
Regulation.txt	5	Water supply release from S-39 and hurricane releases from S-10 structures (m3/d)
SO4.txt	14	Sulfate concentration values (mg/L)
TP.txt	14	TP concentration values (mg/L)

Users may create data files for importing in any spreadsheet editing program. Berkeley-Madonna imports data files in either tab-delimited text format or comma-separated values (CSV) format. The preloaded input files in the model are primarily derived from data downloaded from the South Florida Water Management District's database (DBHYDRO[1]). Although these data are readily available, it is strongly suggested that the user first run model simulations with the preloaded datasets.

[1] http://www.sfwmd.gov/portal/page?_pageid=2894,19708232&_dad=portal&_schema=PORTAL

2.1 Spreadsheet Formatting

Berkeley-Madonna supports definition of one and two dimensional piecewise linear arbitrary functions through importing text files of defining the function vertices. These functions are termed datasets, and are displayed and imported through the dataset window. Text files used in development of the 39-Box model were formatted as comma delimited values (*.csv). One dimensional datasets are defined by data in two columns which may be considered x-values and corresponding y-values on each row. The x-values must be monotonically increasing. A two-dimensional dataset also has x-values in the first column which are monotonically increasing, but begin in row 2. The first row defines a y-value for each column beginning in column 2. The value in (row 1, column 1) is not used and may be set at zero. One dimensional datasets are often used to define time series functions. In the 39-Box model, two dimensional datasets are used to define multiple related time series such as precipitation and evapotranspiration in a single dataset with the y-value designating the column of the desired variable, and the x-value designating time in days. See Appendix 1 for an example of a dataset spreadsheet, and Appendix 2 for a list of datasets and variables.

The 39-Box Model imports time series from 2-dimensional array dataset. The first column of the array is typically the time value in days increasing monotonically from the simulation initial time set at zero, to the final simulation day (in this case, 3287 to 4747). Berkeley-Madonna applies linear interpolation between data values. In order to avoid interpolation in the time series data, the user is encouraged to format the time series such that there are two values (the same value) for each time period (e.g., $t_{1.000}$ = 5.656 and $t_{1.999}$ = 5.656); thus, the imported data become similar to a stepwise constant function. To minimize the effort, a data organization subroutine may be written into a Visual Basic for Applications (VBA) module in Microsoft Excel. Additionally, the user must note that text and other non-numerical symbols will not be imported into Berkeley-Madonna; in fact, the importing process will cease if Berkeley-Madonna encounters such a symbol. As a reference, an abbreviated example spreadsheet, along with its VBA data organization subroutine, is provided in the Appendix of this document.

2.2 Importing Data

Once a time series spreadsheet has been created, it may be imported into Berkeley-Madonna by choosing **Import Dataset** from the **File** menu, or by choosing the **Dataset** from the **Model** menu. The user is then prompted to specify the dataset type and filename; in this case all datasets are entered as **1D** and given a specified filename per Table 1 (the file extension should not be included). Berkeley-Madonna syntax requires that the name of the input file be preceded by a pound sign (#). Typically, for 2-dimensional datasets a timing variable must be given; this variable tells Berkeley-Madonna at which times to read imported data. A simple solution is to use the build in function named **TIME**. In Berkeley-Madonna the syntax **TIME** represents a linear function that counts from the specified start time (**STARTTIME**) value to the specified stop time (**STOPTIME**) value based upon the time step (**DT**). The following equations set variables for precipitation (P) and evapotranspiration (ET) to the appropriate imported time series data values:

```
P = #PET (day, 1); (m/day)
ET = #PET (day, 2); (m/day)
```

where **#PET** indicates the file of imported data being used, **day** is the integer day of the simulation, and the numerical value indicates a column in the imported dataset. Once data are successfully imported, the file name will appear in the Datasets window on the Berkeley-Madonna desktop. Additionally, it is important to note that imported data are saved directly in the Berkeley-Madonna model file (*.mmd). There is no dynamic link between these data and the parent spreadsheet; therefore, any changes to the time series data must be made in the parent *.csv or *.txt file and then re-imported into the Berkeley-Madonna model.

2.3 Comments, Constant Values, Arrays, & Equations

Comments within the equation window provide model self-documentation, and are an important part of the model documentation. There are two alternative syntaxes for comments in Berkeley-Madonna a. Any text between left and right curly brackets, { }, is treated as a comment and not processed. This form of comment can span multiple lines of text. On a single line, all text following a semicolon is also treated as a comment.

Additionally, in imported text data files, all characters on a line beginning with the first non-numeric character are ignored. This allows comments identifying source or column names to be included within these text files.

Equation syntax in Berkeley-Madonna is similar to that in other programming languages such as Basic or FORTRAN. The value calculated on the right-hand-side of an equals sign is assigned to the variable on the left-hand-side. Unlike common programming languages, but similar to spreadsheets, Berkeley-Madonna is non-procedural; meaning the ordering of the equations is not significant. Berkeley-Madonna effectively sorts the equations in order to calculate the value of variables before they are used in subsequent calculations; the program recognizes circular references if such a sorting cannot be accomplished (Macey *et al*, 2000).

Many of the model equations have been consolidated by using arrays. Such equations are set up by using the square brackets ([]) for the values to be arrayed. There are 3 sets of arrayed variables: constituents, compartment, and DMSTA calibration sets. For all equations displayed in sections labeled **3.4.** The user can see examples of arrayed initial conditions and differential equations. Labels for each of the arrayed variables are given below.

Arrays are used extensively in the model to express equations that are repeated for a range of cells or constituents. Many of the array index values have been programmed as constants to enhance clarity of the code. For example, the equation “tp=3” defines a constant named tp that can be used as an array index (subscript) in place of simply the more obscure number 3. Berkeley-Madonna does identify constants during compilation, and there is apparently no runtime cost associated with this programming style.

```
; DEFINE ARRAYS

  ; REFUGE GEOMETRY
    Ncell=11; total number of cells, canal is cell ncell
    Nm=ncell-11; number of marsh cells
    Canal=ncell; cell number for canal (there are 11 canal cells)

  ; CONSTITUENTS
    nconstit= 3
    cl= 1; chloride; conservative
    so4=2; sulfate; monod relationship;
    tp= 3; tp modeled with DMSTA equations

  ; DMSTA CALIBRATION SETS
    emerg = 1; Emergent marsh
    pew = 2; Pre-existing wetland
```

Berkeley-Madonna has a unique notation for array operations (Macey *et al*, 2000). Equations imply looping through a range of subscripts through ranges specified on the left-hand-side of the equation (see for examples sections 3.4.5 and 3.4.6 below). The variables i, j, and k are reserved in Berkeley-Madonna to refer on the right-hand-side of the equation to the first, second, and third array index, respectively, of the variable on the left. This notation replaces loops that are more commonly used in other programming languages.

2.4 Runtime Options

Berkeley-Madonna offers several numerical methods to solve ODEs, Euler's Method, Runge Kutta-2, Runge Kutta-4, Auto stepsize and Rosenbrock (Stiff). The model may be executed accurately and expeditiously using the **RK4** (fourth order Runge-Kutta) method.

The current model is set up to simulate the 7 year period from 2004 to 2010 The user may specify the simulation period with the **STARTTIME** and **STOPTIME** functions. Model coding for the runtime parameters is given below.

3. Code

```
METHOD RK4
STARTTIME = 3287 {JAN04}; 4748 {JAN08};
STOPTIME=5843 {DEC10}; 4747 {DEC07};
```

```
DT = 0.00208333(3 minutes in the form of 180/86400, for the 39-Box DT has to be
<= 0.0025
DTOUT = 1
```

By default, Berkeley-Madonna saves model output every calculation time step, which can become costly as model size and complexity increases. The built-in variable **DTOUT** defines the time period that elapses between data storage for a simulation run. Setting **DTOUT** can reduce memory requirements. Here, model output is stored every one time unit (i.e., one day). If the user desires to store all output data, then **DTOUT** should be set equal to zero or, alternatively, the **DTOUT** statement can be removed.

3.1 Parameters

These values fall into two categories for the model code – simulation option parameters, and model parameters. The simulation option parameters are given at the beginning of the code (found in the **Equations** window). These allow some flexibility with model calculations, input data, and initial conditions. The user can choose outflow type, scale flow and constituent load, choose time series or constant values for boundary concentration, and choose different initial condition sets. The remaining parameters are calibrated and calculated values needed for an accurate base simulation of the model.

All model parameters (constant values) that are not arrayed can also be viewed in the **Parameters** window, which allows the user to change values and reset them without directly changing the code. Parameters with values modified from those set in the **Equations** window are flagged by an asterisk in the **Parameters** window. Users are cautioned that if parameter values are changed using the **Parameter** window, the altered values may persist in future model runs until they are reset. Additionally, the **Overlay Plots** (Figure 3) button can be used to display multiple model runs on the same graph; this feature is very helpful when visually assessing parameter alterations. Lists of all parameters are given in the appendix of this document.

3.2 Processes

Model processes are those equations that contribute to state variable calculation (e.g. groundwater seepage, corrected evapotranspiration, and reaction losses). Such equations represent values that can change with each time step. The user should consult the referenced material for more in depth discussions and explanations of model processes.

3.3 State Variables

Berkeley-Madonna has several ways to code state variables. The model uses the **d/dt()** option to define differential equations. All model differential equations are given below. It is necessary to specify an initial value for all state variables using the **INIT** initializer syntax.

The model directly calculates the change in volume of the Canal and Marsh compartments as per the 2-compartment structure described by Arceneaux *et al* (2007). The stage is then calculated from the volume. It must be noted that the area of the compartments is constant (i.e., it does not change with stage). The calculated value for the exchange flow,B2 (canal to marsh flow) is used to drive the volume differential equations for the 39-compartment constituent model.

3.4.1 Initial Volume Values: 39-Compartment Model

```
Init vol[1..nc]          = D_CO[i]*area[i];
Init vol[(nc+1)..ncell]  = D_MO[i]*area[i];
```

In which:

```
D_CO [1...nc]           = Ecinit[i] - Eoc[i]     initial depth in Canal
D_MO [(nc+1)...ncell]   = Eminit[i] – Eom[i]     initial depth in marsh
```

3.4.2 Volume Differential Equations

```
ETm[(nc+1)..ncell]     = cor_ET[i]                             ;ETm = corrected ET in marsh (m/day)
cor_ET[(nc+1)..ncell] = Fet[i] *ET                             ;cor_ET = corrected ET (m/day)
Fet[(nc+1)..ncell]     = MAX(ETmin, MIN(1, (Hm[i]/Het)))       ;Fet = reduction factor for marsh ET
(dimensionless)
Hm[(nc+1)..ncell]      = MAX(0, (Em[i] - Eom[i]))              ;Hm = marsh water depth (m)
transp = 1-evap                                                ;fraction of ET that is transpired

dn[1..nlinks] = #linkdn(starttime, i)
up[1..nlinks] = #linkup(starttime, i)

{For Qmc estimation}
Q[1..nlinks] =  (10^7)*B[i]*W[i]*(depth[dn[i]]^3)*(E[up[i]]-E[dn[i]])/Radius[i]




{For seepage}

Gc[1..nc]             = lseep*(Ec[i] -  Eb)                    ;Gc = canal seepage loss (m/day)
Gm[(nc+1)..ncell] = rseep*(Em[i] - Eb)                         ;Gm = marsh seepage loss (m/day)

canal seepage loss, m3/day
SpC[1..nc] = Gc[i]*area[i]
SpCanal = ARRAYSUM (SpC[*])

marsh seepage loss, m3/day
SpM[(nc+1)..ncell] = Gm[i]*area[i]
SpMarsh = ARRAYSUM (SpM[*])

totalSeepage = SpCanal + SpMarsh

{Stage}
Ec[1..nc]             = depth[i] + Eoc[i]
Em[(nc+1)..ncell] = depth[i] + Eom[i]
```

d/dt(vol[canal])= (P-ET-Gc[i])*area[i]+(Qin[i]-Qout[i]-QoutHistoric[i])
d/dt(vol[marsh])= (P-ETm[i]-Gm[i])*area[i] + Qin[i]-Q[out]

3.4.3 Mass Calculations

Calculate advective loads for each flow connection, + load is with positive flow

qload[1..nconstit, 1..nlinks]=(max(q[j],0)*conc[i,up[j]])-(max(-q[j],0)*conc[I,dn[j]])
exarea[1..nlinks]=W[i]*min(depth[dn[i]],depth[i]])

Dispersion coefficients: kd1, kd2, kd3

Dispersion loads:
dload[1..nconstit, 1..nc] = kd1*(exarea[j]/Radius[j])*(conc[i,up[j]]-conc[i,dn[j]])
dload[1..nconstit, (nc+1)..22] = kd2*(exarea[j]/Radius[j])*(conc[i,up[j]]-conc[i,dn[j]])
dload[1..nconstit, 23..nlinks] = kd3*(exarea[j]/Radius[j])*(conc[i,up[j]]-conc[i,dn[j]])

Aerial deposition
aload[1..nconstit, 1..ncell] = area[j]*((DD[i]/1000) + (precip[i]*P))

Groundwater seepage + transpiration
gload[1..nconstit, (nc+1)..ncell] = conc[i, j]*(Gm[j]+(transp*ET))*area[j] ; marsh seepage + transp load
gload[1..nconstit, 1..nc] = conc[i, j]*Gc[j]*area[j] ; canal seepage load, no transpiration in canal

Net canal structure inflow - outflow loads
sload[1..nconstit,1..4] = LOAD[i,j] - QoutHistoric[j]*conc[i, j]
sload[1..nconstit,5] = LOAD[i,5] - Qout[5]*conc[i, 5]
sload[1..nconstit,6] = LOAD[i,6] - Qout[6]*conc[i, 6]
sload[1..nconstit,7] = LOAD[i,7] - Qout[7]*conc[i, 7]
sload[1..nconstit,8] = LOAD[i,8] - Qout[8]*conc[i, 8]
sload[1..nconstit,9..nc] = LOAD[i,j] - QoutHistoric[j]*conc[i, j]

Reactive loads (losses)
rload[cl, 1..ncell] = 0 ; conservative

rload[so4, (nc+1)..ncell] = -(kso4[j]/depth[j])*max(mass[i, j],0)
rload[so4, 1..nc] = 0

rload[so4eco, (nc+1)..ncell] = -(MaxSO4Removal*area[j]) * conc[i, j]/(khalfSO4+conc[i, j]) ;g/day
rload[so4eco, 1..nc] = 0

rload[dmsta_tp, (nc+1)..ncell] = Release[i, j] - Uptake[i, j]

interP = 0.0

rload[dmsta_tp, 1..nc]   = interP*area[j]

Where:

i = constituent,
j = compartment number (1-4),
k = DMSTA calibration set (Phosphorus only),
M = mass (g),
t = time (days),
qnet = net mass flow in surface water (g/day),
gload = loss to groundwater seepage and evapotranspiration (g/day),
aload = gain from wet and dry deposition (g/day), and
rload = loss to storage uptake/release (TP) or reaction (SO4) (g/day).
sload = Net canal structure inflow - outflow loads

4.1. SRSM Regulatory Release

A regulatory release is a discharge of water out of the Refuge that occurs as a result of the Refuge stage in relation to the Regulation Schedule (USFWS, 2000). The Regulation Schedule mandates a release of water from the Refuge when a date-dependent stage is exceeded. Magnitude of outflow during a regulatory release is not specified within the Regulation Schedule. It is therefore necessary to assume a water management rule in order to model regulatory releases. Version 4 of the SRSM (Meselhe *et al*, 2009) modeled regulatory releases based on historic discharges, and the same management rule for regulatory release is applied in the 39-Box model.

4.2. Regulatory Release Calculations

A regulatory release is a discharge of water out of the Refuge that occurs as a result of the Refuge stage in relation to the Regulation Schedule. Magnitude of outflow during a regulatory release is not specified within the Regulation Schedule. It is therefore necessary to make assumptions related to water management in order to model regulatory releases. The prior version of the SRSM modeled regulatory releases based on historic discharges.

4. Model Execution & Post-processing

Berkeley-Madonna's user interface for model execution and post-processing is simple and straightforward. All operations needed for a general simulation run can be performed in the **Graph** window. The provided image designates the pertinent buttons with which the user should be familiar; however, for explicit explanations on each button, the Berkeley-Madonna user's guide (Macey *et al*, 2000) should be consulted. The model may be executed from the **Graph** window by pressing the **Run** button; otherwise, the user may select **Run** from the **Compute** menu.

Figure 8: Graphical Output

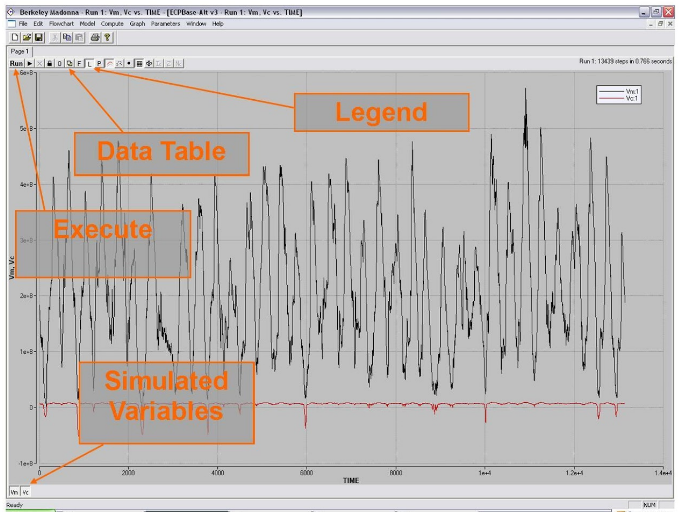
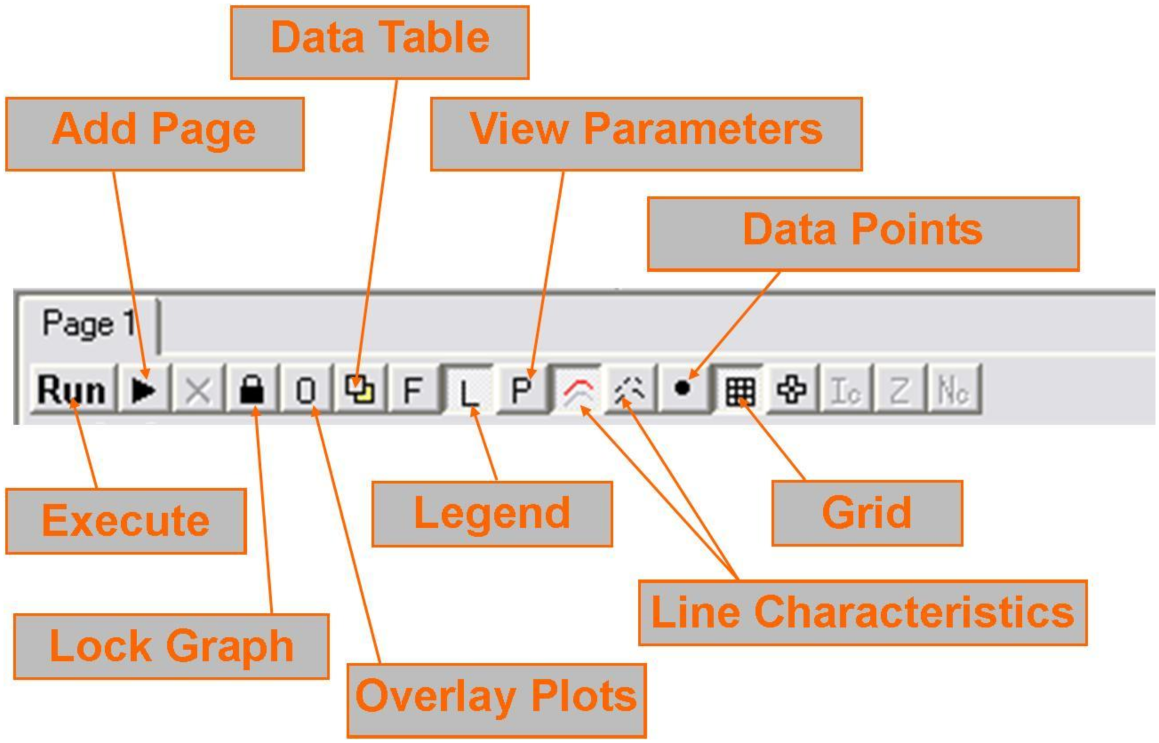


Figure9: Graph Toolbar



After pressing the **Run** button, Berkeley-Madonna will automatically output several variables from the model in the **Graph** window. To specify which variables to output, the user can double-click in the window and choose them from a list or select **Choose Variables** from the **Graph** menu. Once the desired variables are chosen and a model run is complete, the user may print the graph directly from Berkeley-Madonna or export the data as *.csv or *.txt. To export data the table must be displayed in the **Graph** window by clicking the **Data Table** button (see Figure 3). Then the user can select **Export Table** from the **File** menu, or use the **Copy Table** selection under the edit drop down menu.

4.1 Optimization

Berkeley-Madonna Equation solver has the ability to find an optimum value for the parameters in the equation window, to have the least error or divergence from the observed values for different period of times for different input files.

4.1.1 One Parameter Optimization

The one parameter optimization can be done by following these steps:

a) Importing the observed data set for the parameter for the length of time of optimization. This step can be done for one or multiple stations.

b) Write the optimization formula in the equation window, which the user finds more suitable to calculate the optimized factor. For instance, the user may want the optimized factor has the least error with the observed data for that parameter. Here the user can use the error formula like this:

```
AbsError_3cl = ABS (cl3-Obs_3cl)
d/dt (IAE3cl) = AbsError_3cl
INIT IAE3cl = 0
```

Here, the user has defined the absolute error for chloride for compartment 3, equal to the absolute value of the difference between the observed and simulation. And also, he has defined the initial value of this error to be zero. So, for the period of optimization there will be an accumulative error for the compartment 3 for chloride. Now, the summation of all the available observed data for the compartments will reach us to the minimum error. Here it may the compartment 4. So we have:

```
AbsError_4cl=ABS (cl4-Obs_4cl)
d/dt (IAE4cl) = AbsError_4cl
INIT IAE4cl = 0
```

And at the end, the user needs to define a summation for all the absolute errors for the observed stations, for which he wants the value to minimum. This part will be discussed in part 4.c.

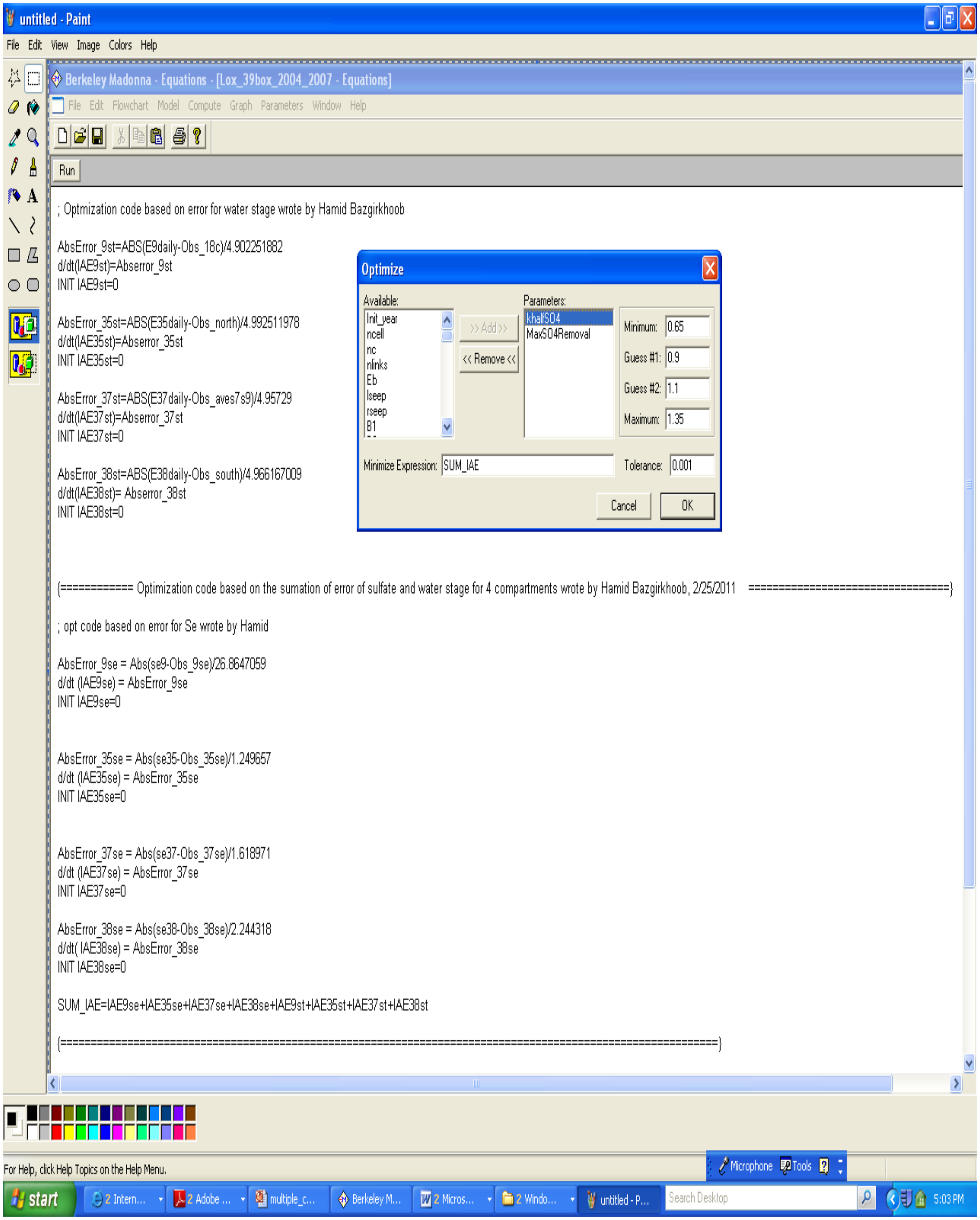
c) Select optimization part from the parameter window. The code will ask for the parameter you want to optimize. As Berkeley-Madonna will do calculations for 7 digits after the point, it will assume each gained answer to be different from with the other one, though the difference is so small as 10^{-7}. At the same time, we know that the Berkeley-Madonna are doing the calculations in numerical equations, which means there may be multiple answers for different ranges which all of them will have the least error with our observed data set. So, here in order to prevent wasting a lot of time to find the optimal value, we can define a minimum and a maximum value for the optimized factor, which actually will be the range for the software to find the optimal value in; and two initial guesses with a tolerance for the answer, to start the calculation around them and also can move to the next guess with a predefined accuracy. The user can predict the optimized value to be in a especial domain, and so can easily define the maximum and minimum and the two initial guesses.

d) At the end, the user need to define what parameter he needs to be minimum; which in the example above will be the summation of the errors for the chloride for different compartments.

4.1.1 4.1.2 Optimization for multiple parameters

The optimization for two or more parameters can be done the same way as for one parameter, except that the parameters may have different units. For instance, the user may want the optimal value of a parameter in such a way that the simulation has the least error with the observed data for a constituent and water level at the same time. The constituent simulation unit in the code has the unit of mg/l and the water level has the unit of meters. Consequently, the user needs to deactivate the role of units in the optimization formula. He needs to divide the result of the absolute error for each individual parameter with the observation by the average of the observed for that period. By this way, the units will be affectless in the formula and at the same time each set of observed data will have its own strength. At the end the user can add up all the absolute errors for different data sets and ask the model to find the optimal value in case the summation of the absolute errors be minimum. Figure 10, shows this calculation for optimization of sulfate and water level for especial compartments at the same time.

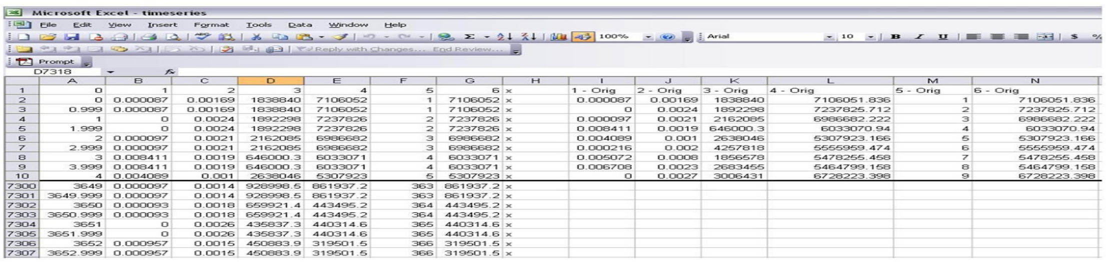
Figure 10: Optimization for two factors at the same time in the Berkeley-Madonna 39-Box Model



Appendix

1. Example Spreadsheet Format

Figure 11: Sample Excel Spreadsheet


Microsoft Excel - timeseries

D7318

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	0	1	2	3	4	5	6	x	1 - Orig	2 - Orig	3 - Orig	4 - Orig	5 - Orig	6 - Orig
2	0	0.000087	0.00169	1838840	7106052	1	7106052	x	0.000087	0.00169	1838840	7106051.836	1	7106051.836
3	0.999	0.000087	0.00169	1838840	7106052	1	7106052	x	0	0.0024	1892298	7237825.712	2	7237825.712
4	1	0	0.0024	1892298	7237826	2	7237826	x	0.000097	0.0021	2162085	6986682.222	3	6986682.222
5	1.999	0	0.0024	1892298	7237826	2	7237826	x	0.008411	0.0019	646000.3	6033070.94	4	6033070.94
6	2	0.000097	0.0021	2162085	6986682	3	6986682	x	0.004089	0.001	2638046	5307923.166	5	5307923.166
7	2.999	0.000097	0.0021	2162085	6986682	3	6986682	x	0.000216	0.002	4257818	5555959.474	6	5555959.474
8	3	0.008411	0.0019	646000.3	6033071	4	6033071	x	0.005072	0.0008	1856578	5478255.458	7	5478255.458
9	3.999	0.008411	0.0019	646000.3	6033071	4	6033071	x	0.006708	0.0023	2683455	5464799.158	8	5464799.158
10	4	0.004089	0.001	2638046	5307923	5	5307923	x	0	0.0027	3006431	6728223.398	9	6728223.398
7300	3649	0.000097	0.0014	928998.5	861937.2	363	861937.2	x						
7301	3649.999	0.000097	0.0014	928998.5	861937.2	363	861937.2	x						
7302	3650	0.000093	0.0018	659921.4	443495.2	364	443495.2	x						
7303	3650.999	0.000093	0.0018	659921.4	443495.2	364	443495.2	x						
7304	3651	0	0.0026	435837.3	440314.6	365	440314.6	x						
7305	3651.999	0	0.0026	435837.3	440314.6	365	440314.6	x						
7306	3652	0.000957	0.0015	450883.9	319501.5	366	319501.5	x						
7307	3652.999	0.000957	0.0015	450883.9	319501.5	366	319501.5	x						

2. Imported Datasets

Variable name	File	Column	Description
P	PET	1	Area Average Precipitation(m/day)
ET	PET	2	Observed Evapotranspiration(m/day)
S39_Out	Outflow	1	Obseved Structure Outflow (m3/day)
G94A_Out	Outflow	2	Obseved Structure Outflow (m3/day)
G94B_Out	Outflow	3	Obseved Structure Outflow (m3/day)
G94C_Out	Outflow	4	Obseved Structure Outflow (m3/day)
G300_Out	Outflow	5	Obseved Structure Outflow (m3/day)
S5AS_Out	Outflow	6	Obseved Structure Outflow (m3/day)
G301_Out	Outflow	7	Obseved Structure Outflow (m3/day)
G338_Out	Outflow	8	Obseved Structure Outflow (m3/day)
S10E_Out	Outflow	9	Obseved Structure Outflow (m3/day)
S10D_Out	Outflow	10	Obseved Structure Outflow (m3/day)
S10C_Out	Outflow	11	Obseved Structure Outflow (m3/day)
S10A_Out	Outflow	12	Obseved Structure Outflow (m3/day)
G94A_in	Inflow	1	Obseved Structure Inflow (m3/day)
G94C_in	Inflow	2	Obseved Structure Inflow (m3/day)

G94D_in	Inflow	3	Obseved Structure Inflow (m3/day)
ACME1_in	Inflow	4	Obseved Structure Inflow (m3/day)
S362_in	Inflow	5	Obseved Structure Inflow (m3/day)
G300_in	Inflow	6	Obseved Structure Inflow (m3/day)
S5AS_in	Inflow	7	Obseved Structure Inflow (m3/day)
S5A_in	Inflow	8	Obseved Structure Inflow (m3/day)
G301_in	Inflow	9	Obseved Structure Inflow (m3/day)
G310_in	Inflow	10	Obseved Structure Inflow (m3/day)
G251_in	Inflow	11	Obseved Structure Inflow (m3/day)
S6_in	Inflow	12	Obseved Structure Inflow (m3/day)
G338_in	Inflow	13	Obseved Structure Inflow (m3/day)
S10A_hurricane	Regulation	1	Supplementary emergency water release (m3/day)
S10C_hurricane	Regulation	2	Supplementary emergency water release (m3/day)
S10D_hurricane	Regulation	3	Supplementary emergency water release (m3/day)
S39_WS	Regulation	4	Supplementary emergency water release (m3/day)
G94A_TP	TP	1	Observed total phosphorus concentration (mg/L)
G94C_TP	TP	2	Observed total phosphorus concentration (mg/L)
G94D_TP	TP	3	Observed total phosphorus concentration (mg/L)
ACME1_TP	TP	4	Observed total phosphorus concentration (mg/L)
S362_TP	TP	5	Observed total phosphorus concentration (mg/L)
G300_TP	TP	6	Observed total phosphorus concentration (mg/L)
S5AS_TP	TP	7	Observed total phosphorus concentration (mg/L)
S5A_TP	TP	8	Observed total phosphorus concentration (mg/L)
G301_TP	TP	9	Observed total phosphorus concentration (mg/L)
G310_TP	TP	10	Observed total phosphorus concentration (mg/L)
G251_TP	TP	11	Observed total phosphorus concentration (mg/L)
S6_TP	TP	12	Observed total phosphorus concentration (mg/L)
G338_TP	TP	13	Observed total phosphorus concentration (mg/L)
G94A_Cl	CL	1	Observed chloride concentration (mg/L)
G94C_Cl	CL	2	Observed chloride concentration (mg/L)
G94D_Cl	CL	3	Observed chloride concentration (mg/L)
ACME1_Cl	CL	4	Observed chloride concentration (mg/L)
S362_Cl	CL	5	Observed chloride concentration (mg/L)
G300_Cl	CL	6	Observed chloride concentration (mg/L)
S5AS_Cl	CL	7	Observed chloride concentration (mg/L)
S5A_Cl	CL	8	Observed chloride concentration (mg/L)
G301_Cl	CL	9	Observed chloride concentration (mg/L)
G310_Cl	CL	10	Observed chloride concentration (mg/L)
G251_Cl	CL	11	Observed chloride concentration (mg/L)
S6_Cl	CL	12	Observed chloride concentration (mg/L)
G338_Cl	CL	13	Observed chloride concentration (mg/L)
G94A_SO4	SO4	1	Observed sulfate concentration (mg/L)
G94C_SO4	SO5	2	Observed sulfate concentration (mg/L)
G94D_SO4	SO6	3	Observed sulfate concentration (mg/L)
ACME1_SO4	SO7	4	Observed sulfate concentration (mg/L)
S362_SO4	SO8	5	Observed sulfate concentration (mg/L)

G300_SO4	SO9	6	Observed sulfate concentration (mg/L)
S5AS_SO4	SO10	7	Observed sulfate concentration (mg/L)
S5A_SO4	SO11	8	Observed sulfate concentration (mg/L)
G301_SO4	SO12	9	Observed sulfate concentration (mg/L)
G310_SO4	SO13	10	Observed sulfate concentration (mg/L)
G251_SO4	SO14	11	Observed sulfate concentration (mg/L)
S6_SO4	SO15	12	Observed sulfate concentration (mg/L)
G338_SO4	SO16	13	Observed sulfate concentration (mg/L)

3. Simulation Option Parameters

Variable Name	Default Value	Explanation
CalcQRo	1	Distinguishes between calculated (1) or historic outflow (0)
RSQfact	1	Scaling factor for regulatory release

4. Model Parameters

Variable Name	Value	Description
ncell	39	Total number of cells
nc	11	Number of canal cells
nlinks	52	Number of links for link-node model
C1_area	67451.13582	Canal Surface Area (m2)
C2_area	432580.5052	Canal Surface Area (m2)
C3_area	1321081.929	Canal Surface Area (m2)
C4_area	1070766.749	Canal Surface Area (m2)
C5_area	343866.8469	Canal Surface Area (m2)
C6_area	242568.4998	Canal Surface Area (m2)
C7_area	290820.5459	Canal Surface Area (m2)
C8_area	1172416.461	Canal Surface Area (m2)
C9_area	745969.4192	Canal Surface Area (m2)
C10_area	770845.383	Canal Surface Area (m2)
C11_area	325422.4124	Canal Surface Area (m2)
M12_area	1181402.532	Marsh Surface Area (m2)
M13_area	2501726.142	Marsh Surface Area (m2)
M14_area	293003.0385	Marsh Surface Area (m2)
M15_area	543785.3522	Marsh Surface Area (m2)
M16_area	3485961.831	Marsh Surface Area (m2)
M17_area	4045182.15	Marsh Surface Area (m2)

M18_area	3195752.157	Marsh Surface Area (m2)
M19_area	7636805.725	Marsh Surface Area (m2)
M20_area	7200836.399	Marsh Surface Area (m2)
M21_area	13400481	Marsh Surface Area (m2)
M22_area	4195293.496	Marsh Surface Area (m2)
M23_area	14539203.09	Marsh Surface Area (m2)
M24_area	1890978.078	Marsh Surface Area (m2)
M25_area	7512971.275	Marsh Surface Area (m2)
M26_area	988502.2216	Marsh Surface Area (m2)
M27_area	5248955.804	Marsh Surface Area (m2)
M28_area	522323.8002	Marsh Surface Area (m2)
M29_area	2903428.105	Marsh Surface Area (m2)
M30_area	4434932.933	Marsh Surface Area (m2)
M31_area	12224954.71	Marsh Surface Area (m2)
M32_area	4740640.266	Marsh Surface Area (m2)
M33_area	12213974.99	Marsh Surface Area (m2)
M34_area	61064630.23	Marsh Surface Area (m2)
M35_area	62998466.73	Marsh Surface Area (m2)
M36_area	87163883.38	Marsh Surface Area (m2)
M37_area	135435126.2	Marsh Surface Area (m2)
M38_area	31888461.52	Marsh Surface Area (m2)
M39_area	63423036.38	Marsh Surface Area (m2)
Ecinit	5.10	Initial Canal Stage (m)
Eminit	5.09	Initial Marsh Stage (m)
Eoc_C1	2.44289	Canal bottom elevation (m)
Eoc_C2	0.98225	Canal bottom elevation (m)
Eoc_C3	0.941286	Canal bottom elevation (m)
Eoc_C4	0.498	Canal bottom elevation (m)
Eoc_C5	0.58	Canal bottom elevation (m)
Eoc_C6	0.716	Canal bottom elevation (m)
Eoc_C7	0.6905	Canal bottom elevation (m)
Eoc_C8	-0.106778	Canal bottom elevation (m)
Eoc_C9	1.42233	Canal bottom elevation (m)
Eoc_C10	1.64929	Canal bottom elevation (m)
Eoc_C11	1.24433	Canal bottom elevation (m)
Eom_M12	4.645	marsh bottom elevation (m)
Eom_M13	4.645	marsh bottom elevation (m)
Eom_M14	4.55	marsh bottom elevation (m)
Eom_M15	4.55	marsh bottom elevation (m)
Eom_M16	4.55	marsh bottom elevation (m)
Eom_M17	4.55	marsh bottom elevation (m)
Eom_M18	4.13	marsh bottom elevation (m)
Eom_M19	4.42	marsh bottom elevation (m)
Eom_M20	4.42	marsh bottom elevation (m)
Eom_M21	4.33	marsh bottom elevation (m)
Eom_M22	4.04	marsh bottom elevation (m)
Eom_M23	4.04	marsh bottom elevation (m)
Eom_M24	4.04	marsh bottom elevation (m)
Eom_M25	4.04	marsh bottom elevation (m)
Eom_M26	4.29	marsh bottom elevation (m)

Eom_M27	4.29	marsh bottom elevation (m)
Eom_M28	4.58	marsh bottom elevation (m)
Eom_M29	4.56	marsh bottom elevation (m)
Eom_M30	4.6	marsh bottom elevation (m)
Eom_M31	4.63	marsh bottom elevation (m)
Eom_M32	4.74	marsh bottom elevation (m)
Eom_M33	4.74	marsh bottom elevation (m)
Eom_M34	4.75	marsh bottom elevation (m)
Eom_M35	4.83	marsh bottom elevation (m)
Eom_M36	4.62	marsh bottom elevation (m)
Eom_M37	4.55	marsh bottom elevation (m)
Eom_M38	4.19	marsh bottom elevation (m)
Eom_M39	4.43	marsh bottom elevation (m)
Eb	3.5	Water Stage outside Refuge (m)
lseep	0.0484046	Canal Seepage Constant (1/day)
rseep	8.16E-10	Marsh Seepage Constant (1/day)
B1	6.97621	Transport Coefficient Between Canal and Canal(1/m-day)
B2	1.13863	Transport Coefficient Between Canal and Marsh(1/m-day)
B3	4.55002	Transport Coefficient Between Marsh and Marsh(1/m-day)
Radius_link 1	3345	Radius of link 1
Radius_link 2	8059	Radius of link 2
Radius_link 3	12408	Radius of link 3
Radius_link 4	9781	Radius of link 4
Radius_link 5	5366	Radius of link 5
Radius_link 6	3279	Radius of link 6
Radius_link 7	7120	Radius of link 7
Radius_link 8	14101	Radius of link 8
Radius_link 9	12465	Radius of link 9
Radius_link 10	7607	Radius of link 10
Radius_link 11	2730	Radius of link 11
Radius_link 12	108	Radius of link 12
Radius_link 13	182	Radius of link 13
Radius_link 14	671	Radius of link 14
Radius_link 15	429	Radius of link 15
Radius_link 16	248	Radius of link 16
Radius_link 17	477	Radius of link 17
Radius_link 18	183	Radius of link 18
Radius_link 19	472	Radius of link 19
Radius_link 20	1314	Radius of link 20
Radius_link 21	338	Radius of link 21
Radius_link 22	327	Radius of link 22
Radius_link 23	375	Radius of link 23
Radius_link 24	495	Radius of link 24
Radius_link 25	541	Radius of link 25
Radius_link 26	805	Radius of link 26
Radius_link 27	778	Radius of link 27
Radius_link 28	798	Radius of link 28
Radius_link 29	592	Radius of link 29
Radius_link 30	843	Radius of link 30
Radius_link 31	1103	Radius of link 31

Radius_link 32	1022	Radius of link 32
Radius_link 33	850	Radius of link 33
Radius_link 34	9990	Radius of link 34
Radius_link 35	9892	Radius of link 35
Radius_link 36	7476	Radius of link 36
Radius_link 37	3222	Radius of link 37
Radius_link 38	4213	Radius of link 38
Radius_link 39	3551	Radius of link 39
Radius_link 40	3028	Radius of link 40
Radius_link 41	5420	Radius of link 41
Radius_link 42	3427	Radius of link 42
Radius_link 43	5946	Radius of link 43
Radius_link 44	3631	Radius of link 44
Radius_link 45	7994	Radius of link 45
Radius_link 46	10674	Radius of link 46
Radius_link 47	10051	Radius of link 47
Radius_link 48	10497	Radius of link 48
Radius_link 49	9684	Radius of link 49
Radius_link 50	6275	Radius of link 50
Radius_link 51	9330	Radius of link 51
Radius_link 52	6390	Radius of link 52
Width_link 1	37.5	Width of link 1
Width_link 2	55	Width of link 2
Width_link 3	55	Width of link 3
Width_link 4	55	Width of link 4
Width_link 5	55	Width of link 5
Width_link 6	55	Width of link 6
Width_link 7	47.5	Width of link 7
Width_link 8	40	Width of link 8
Width_link 9	40	Width of link 9
Width_link 10	40	Width of link 10
Width_link 11	35	Width of link 11
Width_link 12	896	Width of link 12
Width_link 13	5148	Width of link 13
Width_link 14	11683	Width of link 14
Width_link 15	13712	Width of link 15
Width_link 16	6128	Width of link 16
Width_link 17	4163	Width of link 17
Width_link 18	2872	Width of link 18
Width_link 19	15871	Width of link 19
Width_link 20	13305	Width of link 20
Width_link 21	11729	Width of link 21
Width_link 22	4798	Width of link 22
Width_link 23	1130	Width of link 23
Width_link 24	4659	Width of link 24
Width_link 25	11671	Width of link 25
Width_link 26	13680	Width of link 26
Width_link 27	6152	Width of link 27
Width_link 28	4155	Width of link 28
Width_link 29	2869	Width of link 29

Width_link 30	15967	Width of link 30
Width_link 31	13315	Width of link 31
Width_link 32	11687	Width of link 32
Width_link 33	3932	Width of link 33
Width_link 34	867	Width of link 34
Width_link 35	788	Width of link 35
Width_link 36	4779	Width of link 36
Width_link 37	11212	Width of link 37
Width_link 38	12928	Width of link 38
Width_link 39	5496	Width of link 39
Width_link 40	3907	Width of link 40
Width_link 41	2960	Width of link 41
Width_link 42	14143	Width of link 42
Width_link 43	12725	Width of link 43
Width_link 44	11618	Width of link 44
Width_link 45	4219	Width of link 45
Width_link 46	7107	Width of link 46
Width_link 47	5582	Width of link 47
Width_link 48	9358	Width of link 48
Width_link 49	10656	Width of link 49
Width_link 50	14161	Width of link 50
Width_link 51	12202	Width of link 51
Width_link 52	8643	Width of link 52
ETmin	0.2	minimum ET reduction factor for marsh
HET	0.25	ET depth reduction boundary(m)
evap	0.65	Fraction of ET that is evaporation
transp	0.35	Fraction of ET that is transpiration
mindepth	0.05	Minimum water depth (m)
minvol_C1	1986.7	Minimum volume of Canal1
minvol_C2	14618	Minimum volume of Canal2
minvol_C3	32939.1	Minimum volume of Canal3
minvol_C4	38640.1	Minimum volume of Canal4
minvol_C5	17292.4	Minimum volume of Canal5
minvol_C6	11932	Minimum volume of Canal6
minvol_C7	8164.05	Minimum volume of Canal7
minvol_C8	33192.9	Minimum volume of Canal8
minvol_C9	27266.9	Minimum volume of Canal9
minvol_C10	23701.2	Minimum volume of Canal10
minvol_C11	11028	Minimum volume of Canal11
minvol_M12	14650.2	Minimum volume of Marsh12
minvol_M13	27189.3	Minimum volume of Marsh13
minvol_M14	59070.1	Minimum volume of Marsh14
minvol_M15	174298	Minimum volume of Marsh15
minvol_M16	159788	Minimum volume of Marsh16
minvol_M17	381840	Minimum volume of Marsh17
minvol_M18	209765	Minimum volume of Marsh18
minvol_M19	726960	Minimum volume of Marsh19
minvol_M20	94548.9	Minimum volume of Marsh20
minvol_M21	375649	Minimum volume of Marsh21
minvol_M22	49425.1	Minimum volume of Marsh22

minvol_M23	262448	Minimum volume of Marsh23
minvol_M24	26116.2	Minimum volume of Marsh24
minvol_M25	145171	Minimum volume of Marsh25
minvol_M26	221747	Minimum volume of Marsh26
minvol_M27	611248	Minimum volume of Marsh27
minvol_M28	237032	Minimum volume of Marsh28
minvol_M29	610699	Minimum volume of Marsh29
minvol_M30	360042	Minimum volume of Marsh30
minvol_M31	670024	Minimum volume of Marsh31
minvol_M32	125086	Minimum volume of Marsh32
minvol_M33	202259	Minimum volume of Marsh33
minvol_M34	3.05323E+06	Minimum volume of Marsh34
minvol_M35	3.14992E+06	Minimum volume of Marsh35
minvol_M36	4.35819E+06	Minimum volume of Marsh36
minvol_M37	6.77179E+06	Minimum volume of Marsh37
minvol_M38	1.59442E+06	Minimum volume of Marsh38
minvol_M39	3.17115E+06	Minimum volume of Marsh39
nconstit	5	Number of calculated constituents in the model
Cl	1	Chloride, conservative
SO4	2	Sulfate, apparent settling
tp	3	Total Phosphorus, k-c*model
SO4eco	4	Sulfate, monod relationship, ecolab
dmsta_tp	5	TP modeled with DMSTA equations
Precip_Cl	2.00	Chloride Concentration from Rainfall (mg/L)
DD[cl]	3.110198	Dry deposition of chloride (mg/m2-day)
Precip_SO4	1.00	Sulfate Concentration from Rainfall (mg/L)
DD[SO4]	0.378371	Dry deposition of SO4 (mg/m2-day)
Precip[SO4eco]	1.00	SO4eco Concentration from rainfall (mg/L)
DD[SO4eco]	0.378371	Dry deposition of SO4eco (mg/m2-day)
Precip_TP	0.010	TP Concentration from Rainfall (mg/L)
DD[tp]	0.110	Dry deposition of Phosphorus (mg/m2-day)
Precip[dmsta_tp]	0.010	wet deposition of TP(mg/L)
DD[dmsta_tp]	0.110	dry deposition of TP (mg/m2.day)
KhalfSO4	0.650927	Sulfate half saturation constant (g/m3)
MaxSO4removal	0.074991	Maximum sulfate removal (g/m2-yr)
Ktp	0.045995	settling rate (m/day)
cstarm	0.008	C* in marsh cells
cstarc	0.080	C* in canal cells
K1[emerg]	0.291307	Phosphorus (emergent wetland) maximum uptake rate(m3/g/day)
K1[Pew]	0.605065	Phosphorus (Pre-existing wetland) maximum uptake rate(m3/g/day)
K2[emerg]	0.005476	Phosphorus (Emergent wetland) recycle rate(m2/g/day)
K2[Pew]	0.011499	Phosphorus (Pre-existing wetland) recycle rate(m2/g/day)
K3[emerg]	0.000874	Phosphorus (Emergent wetland) burial rate(1/day)
K3[Pew]	0.001815	Phosphorus (Pre-existing wetland) burial rate(1/day)
kd1	0	Dispersion Coefficient between canal and canal (m2/day)
kd2	0	Dispersion Coefficient between canal and marsh (m2/day)
kd3	43200	Dispersion Coefficient between marsh and marsh (m2/day)

LITERATURE CITED

Arceneaux, J., Meselhe, E. A., Griborio, A., and Waldon, M. G. (2007). "The Arthur R. Marshall Loxahatchee National Wildlife Refuge Water Budget and Water Quality Models." *Report No. LOXA-07-004, currently available at < http://mwaldon.com/LoxModel/reports/LOXA07-004--2007-06.pdf >*, University of Louisiana at Lafayette in cooperation with the US Fish and Wildlife Service, Lafayette, Louisiana.

Macey, R., Oster, G., and Zahnley, T. (2000). "Berkeley Madonna User's Guide Version 8.0." University of California, Department of Molecular and Cellular Biology, Berkeley, CA.

Meselhe, E. A., Griborio, A. G., and Gautam, S. (2005). "Hydrodynamic And Water Quality Modeling For The A.R.M. Loxahatchee National Wildlife Refuge, Phase 1: Preparation Of Data, Task 1: Data Acquisition and Processing." *Report #LOXA05-014 available at < http://sofia.usgs.gov/lox_monitor_model/advisorypanel/data_acq_report.html >*, University of Louisiana-Lafayette in cooperation with the US Fish and Wildlife Service, Lafayette, LA.

Meselhe, E., Arceneaux, J., and Waldon, M. G. (2007a). "Mass Balance Model Version 1.01." *LOXA-07-002, currently available at < http://loxmodel.mwaldon.com >*, University of Louisiana - Lafayette in cooperation with the US Fish and Wildlife Service, Lafayette, LA.

Meselhe, E., Arceneaux, J., Waldon, M. G., and Wang, H. (2007b). "Water Quality Model Version 1.00." *LOXA-07-003, currently available at < http://loxmodel.mwaldon.com >*, University of Louisiana - Lafayette in cooperation with the US Fish and Wildlife Service, Lafayette, LA.

Meselhe, E. A., Waldon, M. G., and Roth, W. (2009). "A.R.M. Loxahatchee National Wildlife Refuge Simple Refuge Screening Model Version 4.00 User's Manual." University of Louisiana - Lafayette, Lafayette, Louisiana.

Meselhe, E., Chen, C., Bazgirhoob, H., Wang, H., Roth, B., Arceneaux, J., and Waldon, M. G. (2011). "Near Zeroth Order Sulfate Disappearance Rate Estimated using Models of the Arthur R. Marshall Loxahatchee National Wildlife Refuge " Annual Greater Everglades Mercury/Sulfur Workshop, June 21-22, 2011, West Palm Beach, FL.

USFWS. (2000). "Arthur R. Marshall Loxahatchee National Wildlife Refuge Comprehensive Conservation Plan." *available at < http://loxahatchee.fws.gov >*, US Fish and Wildlife Service, Boynton Beach, Florida.

Walker, W. W., Jr., and Kadlec, R. H. (2011). "Modeling Phosphorus Dynamics in Everglades Wetlands and Stormwater Treatment Areas." Critical Reviews in Environmental Science and Technology, 41(6), 430-446.

Wang, H., Meselhe, E. A., Waldon, M. G., Surratt, D., Abdou, S., Chen, C., and Harwell, M. C. (2008). "Compartment delineation for a wetland water quality model in the Northern Everglades, Florida, USA." *Journal of Environmental Hydrology*, 16, Paper 36.

Wang, H., Meselhe, E. A., Waldon, M. G., Harwell, M. C., and Chen, C. (2012). "Compartment-based hydrodynamics and water quality modeling of a Northern Everglades Wetland, Florida, USA." Ecological Modelling, 247, 273-285.