Development of a User-Friendly Stormwater Pollution Model
Prediction of Nitrate, Phosphate, and TSS Pollutant Loading using ArcGIS 9

Brandon Luzar ‘05
Swarthmore College – Department of Engineering
Engineering 90 – Prof. Arthur E. McGarity

Abstract

GWLF was implemented into Arc9 so as to create a user friendly stormwater pollution model. The model was compared with real monitoring data from the month of July 2004 in a selected urbanized watershed in Springfield Township. It was found that improvements were made to the GWLF model with respect to urbanized watershed pollution prediction. Dissolved Phase nutrient loadings were added to the model to better represent urbanized nutrient loadings. Model calibration was limited by the amount of field verification data available across differing seasons. Future work still remains to be done on the model to calibrate it with urbanized watersheds and expand overall capabilities.

TABLE OF CONTENTS

INTRODUCTION

The human race is a continuously growing species that is growing to inhabit every corner of the Earth. To support the basic needs of its constituents, society has changed the Earth's surface in many ways; including the construction of cities, roads, power lines and the destruction of forests, wetlands, and other natural ecosystems. As a result of an overall increase of impervious surface area and decrease of natural water processing systems, nonpoint pollution from stormwater runoff now threatens the well being and beauty of the Earth and its inhabitants. Nonpoint stormwater pollution has been a growing concern among the environmentally active community over the past half-century, starting with the Clean Water Act of 1972 and the subsequent creation of the Environmental Protection Agency. Since then federal money has been allocated to help regulate nonpoint stormwater pollution and increase research of the problem.

The Swarthmore College Environmental Engineering Program, under the guidance of Prof. Arthur E. McGarity, has been involved with nonpoint stormwater pollution research for the past decade. Currently, the program is working on a multiphase stormwater pollution prediction and BMP (Best Management Practice) allocation model. This proposed model uses GIS (Geographic Information Systems) software to predict and model both stormwater pollutant loading and possible corrective solutions to problems caused by pollutants. The following program is the portion of the model that predicts nitrate, phosphate, and TSS (Total Suspended Solids) pollutant loadings using GIS software and data.

POLLUTANT LOADING BACKGROUND

The phosphate ion is one pollutant found in nonpoint stormwater runoff. Soaps, fertilizers, and organic wastes (fallen leaves, grass clippings, animal waste, suspended soil particles, etc.) are all major sources of the phosphate ion in stormwater runoff and can lead to eutrophication in bodies of water. Eutrophication is the presence of excess nutrients in a waterway such that the over-stimulation of organic matter (algal growth) can be considered a hazard to the aquatic ecosystem. More specifically, phosphate pollution leads to the depletion of dissolved oxygen concentrations, decreased light transmittance, the killing of aquatic and marine life, and the aesthetic degradation of affected waterways.

Another common pollutant found in nonpoint stormwater runoff is the nitrate ion. Primarily, nitrate found in runoff is a product of agricultural fertilizers. As is the case with the phosphate ion, excess nitrate leads to eutrophication in waterways but various human health risks, blue baby syndrome and non-Hodgkins lymphoma, connected with excess nitrate in drinking water are growing concerns as well.

A third pollutant resulting from nonpoint stormwater pollution is Total Suspended Solids. Stormwater runoff picks up TSS from agricultural land and stream banks in situations of abnormally high runoff and stream flows. This pollutant is visibly apparent in stormwater because it greatly decreases the depth of light transmittance through the body of water. TSS adversely affects the geomorphology of the environment by redistributing soil particles from stream banks and agricultural areas to fluvial sediment deposits and, thus, altering the habitats of downstream aquatic life and stream side vegetation. Lastly, various heavy metals and pathogenic compounds are present in TSS and can infiltrate drinking water and recreational sites causing health risks to humans and other animals.

BRIEF MODELING HISTORY OF STORMWATER POLLUTION

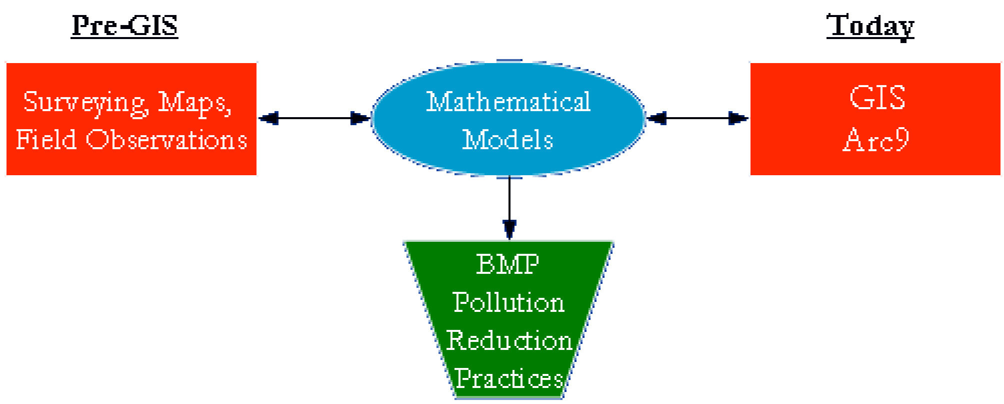


Figure 1 – Mathematical Models, GIS, and Other Parameter Sources

GIS software allows a user to integrate the information storage and alteration capabilities of databases with the visual and practical aspects of maps. Over the past two decades, public research projects and developments in GIS Software have expanded the capabilities of geographic modeling and, more specifically, nonpoint stormwater pollution modeling. Models have been created using the data storage and function building capabilities GIS software to calculate pollutant loading based on verified scientific methods and regionally specific parameters. Two decades ago, decision makers would be subject to the long and tedious task of locating the vast array of parameters required to run a mathematical model that simulates stormwater pollution. Today, these parameters are easily accessible in large GIS databases that include thousands of data layers containing millions of parameters. Most of this software and data can be easily accessed in the public domain, but a recent, up to date GIS based model that simulates smaller urban watersheds is not readily available.

Over the past three years, Dr. Barry M. Evans et al of the Pennsylvania State University have developed a version of GWLF (Globalized Watershed Loading Functions) called AVGWLF (ArcViewGWLF). GWLF, a program written in quickBasic, was created at Cornell University by Haith et al in 1992 for predicting monthly nitrate, phosphate, and TSS loads in agricultural watershed. Evans et al adopted GWLF to a GIS based, user-friendly interface in ArcView 3.2 and tested the program for accuracy across the state of Pennsylvania. They concluded that AVGWLF was an adequate tool for predicting nutrient pollutant loading in large, agricultural watersheds across Pennsylvania and AVGWLF is now used throughout the state's environmental modeling industry. Recent research completed by Dr. Arthur E. McGarity at Swarthmore College to assess the accuracy of AVGWLF when used in small, urbanized watersheds suggests some inaccuracy in the model at very small assessment levels.

GWLF is one of many stormwater pollution models currently available which uses moderately complex scientific theory. Simpler models, such as the PLOAD model created by the EPA, utilize flow average concentrations or export coefficients to calculate pollutant concentration in streams and stormwater runoff. These models are theoretically less complex than a GWLF, which uses a combination of a mass balances, the universal soil loss equation, and buildup coefficients to calculate pollutant loadings. Conversely, HSPF (also developed by the EPA) and other more complicated models than GWLF are available in the public domain.

ESRI recently released their latest version of GIS software, ArcView 9, in 2004. The new software has increased function building and script writing capabilities along with increase overall user-friendliness. ArcView 9 will serve as the interface to connect GWLF, GIS technology, and calibration for a small urbanized watershed model.

MODEL THEORY

Model theory was developed by Haith et al.

Rural and Urban Runoff

All runoff in GWLF is determined by the U.S Soil Conservation Service's Curve Number method shown below:

$$Q_{kt} = \frac{(R_t + M_t - 0.2DS_{kt})^2}{R_t + M_t + 0.8DS_{kt}} \quad (1)$$

where: R_t = Rainfall on day t in cm
M_t = Snow melt on day t in cm
DS_{kt} = Detention parameter which relates land usage to runoff

Snow melt is calculated on days where accumulated snow is present and the temperature is less than 0 ^{o}C.

$$M_t = 0.45T_t \tag{2}$$

where: T_t = Average daily temperature on day t in ^{o}C.

The detention parameter is calculated according to the curve number, CN_{kt}, and equation 3.

$$DS_{kt} = \frac{2540}{CN_{kt}} - 25.4 \tag{3}$$

where: CN_{kt} = Curve number for source area *k* on day *t* as determined from the antecedent precipitation equations.

Curve numbers are determined in accordance with the antecedent moisture of the previous five days. There are three curve numbers for the driest (CN_{1k}), average (CN_{2k}), and wettest (CN_{3k}) antecedent moisture periods. CN_{2k} is selected according to the land use, soil type, and hydrologic condition of the land. Tables containing curve number values are located in Appendix 3.

$$CN_{1k} = \frac{CN_{2k}}{2.334 - 0.01334CN_{2k}} \tag{4}$$

and

$$CN_{3k} = \frac{CN_{2k}}{0.4036 - 0.0059CN_{2k}} \tag{5}$$

The antecedent precipitation of the previous five days is the summation of the combined snow melt and rain.

$$A_t = \sum_{n=t-5}^{t-1} \left(R_n + M_n\right) \tag{6}$$

where R_n and M_n are as defined above.

Antecedent curve numbers are selected according to the value of A_t and whether or not the current month is in a growing season. For $A_t < 1.6$ cm in dormant seasons and $A_t < 3.6$ cm in wet seasons CN_{1k} is used, for $A_t > 2.8$ cm in dormant seasons and $A_t > 5.3$ cm in wet seasons CN_{3k} is used, and for At values that fall between these two limits CN_{2k} is used.

Sediment Loads

Total sediment loading is a monthly mass summation of the sediment loading on each individual source area k on day t. Total sediment loading on the stream is a function of the sediment delivery ratio and the amount of displaced erosion in the watershed during a storm event.

$$Xj = DR\sum_{k}\sum_{t=1}^{d_j} X_{kt} \qquad (7)$$

where: X_{kt} = Erosion from area *k* on day *t* (Mg)
DR = Sediment delivery ratio that various by region and watershed (mg/kg)
d_j = Number of days in month *j*

Rural erosion values for all source area *k* on day *t* are calculated using the Universal Soils Loss Equation shown in equation 9

$$X_{kt} = 0.132 RE_t K_k \left(LS\right)_k C_k P_k AR_k \qquad (8)$$

where: RE_t = Rainfall erosivity on day *t*
K_k, $(LS)_k$, C_k, and P_k are coefficients representing soil erodibility, topography, cover, and management. These parameters are lumped together into one $KLSCP_k$ factor for source area *k* for usage in the program.

The rainfall erosivity is defined as:

$$RE_t = 64.6 a_t R_t^{1.81} \qquad (9)$$

where: a_t = Geographic and seasonal coefficient on day *t* that varies by location
R_t = Rainfall on day *t* in cm

Sediment is then distributed over the remaining months of the year according to a transport factor and a transport capacity for each of the remaining months. The transport factor (TR_j) for each month is defined as:

$$TR_j = \sum_{t=1}^{d_j} Q_t^{5/3} \qquad (10)$$

and the transport capacity (B_j) for each month is:

$$B_j = \sum_{h=j}^{12} TR_h \qquad (11)$$

Finally, the available sediment in month m is defined as:

$$Y_m = TR_m \sum_{j=1}^{m} \left(X_j \big/ B_j \right) \tag{12}$$

Rural and Urban Dissolved Phase Nutrient Loads

Dissolved phase rural nutrient loads are calculated using flow weighted event mean concentrations and runoff from the source area.

$$LD_m = 0.1 \sum_{k} \sum_{t=1}^{d_m} Cd_k Q_{kt} AR_k \tag{13}$$

where: LD_w = Dissolved nutrient load in month m (kg)
Cd_k = Event mean concentration (mg/L)
Q_{kt} = Runoff from source area k on day t (cm)
AR_k = Area of source area k (ha)

Urban Solid Phase Nutrient Loads

Solid Phase urban nutrient loads are accumulated according to the constant accumulation rate equation.

$$W_{kt} = w_{kt} \left[N_{kt} e^{-0.12} + \left(n_k \big/ 0.12 \right) \left(1 - e^{-0.12} \right) \right] \tag{14}$$

where: w_{kt} = Unit erosion variable from first order washoff function
N_{kt} = Accumulated nutrient load on source area
n_k = Constant nutrient accumulation rate

Nutrients are related to solid phase sediment loading by a sediment unit variable. The unit variable is computed from the first order washoff equation such that 1.27 cm of runoff will remove 90% of the sediment accumulated.

$$w_{kt} = 1 - e^{-1.81 Q_{kt}} \tag{15}$$

where: Q_{kt} = unit runoff from source area k on day t

Groundwater Hydrology

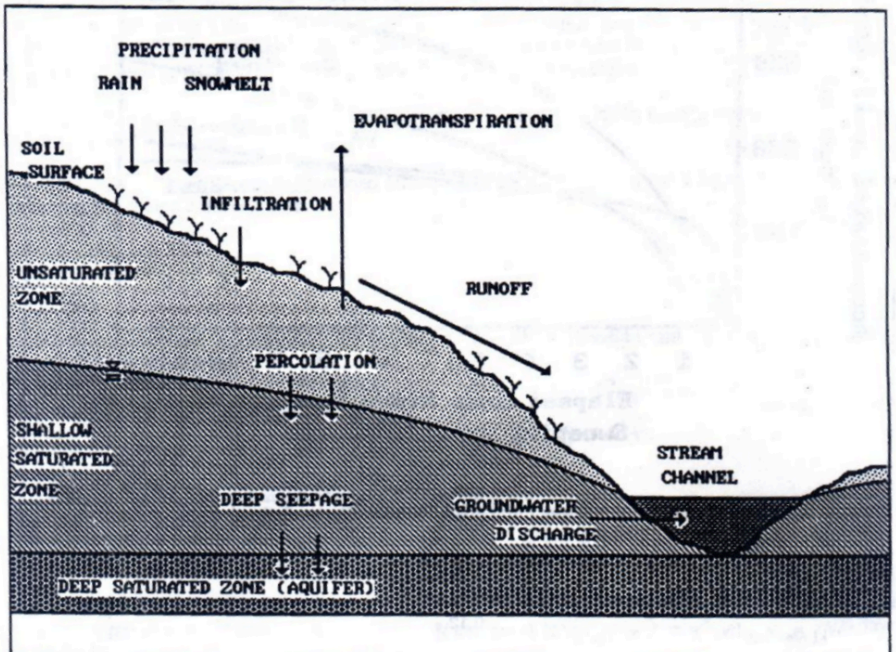


Figure 2 – Lumped parameter hydrology (Haith et al)

The model contains a daily mass balance that occurs over all groundwater contained in three storage zones: the unsaturated storage zone, the low saturated storage zone, and the deep saturated storage zone. All water flowing into the deep saturate storage zone and into the stream system will exit the mass balance system. The two governing equations for the unsaturated and low saturated storages zones are:

$$U_{t+1} = U_t + R_t + M_t - Q_t - E_t - PC_t \qquad (16)$$

and

$$S_{t+1} = S_t + PC_t - G_t - D_t \qquad (17)$$

where: U_t = Water in unsaturated storage zone on day t in cm
S_t = Water in saturated storage zone on day t in cm
R_t = Rain in cm on day t
M_t = Snow melt in cm on day t
Q_t = Runoff in cm on day t
E_t = Evapotranspiration in cm on day t
PC_t = Percolation in cm into the low saturated zone on day t
G_t = Groundwater discharge into stream on day t in cm
D_t = Seepage into the deep saturated zone in cm on day t

When the maximum unsaturated soil capacity is exceeded, evapotranspiration occurs according to the following optimization:

Maximize PC_t such that :

$$PC_t = 0$$
$$or \quad (18)$$
$$PC_t = U_t + R_t + M_t - Q_t - E_t - U_{\max}$$

where: U_{max} = Maximum water content in the unsaturated storage zone in cm

Conversely, evapotranspiration cannot exceed the available water in the unsaturated zone and is subject to the following optimization:

$$\text{Minimize } \mathrm{E}_t \text{ such that :}$$
$$E_t = CV_t PE_t$$
$$or$$
$$E_t = U_t + R_t + M_t - Q_t$$

where: CV_t = Empirically derived cover coeffiecient
PE_t = Potential evapotranspiration on day *t* in cm

Potential evapotranspiration is computed by:

$$PE_t = \frac{0.021 H_t^2 e_t}{T_t + 273} \quad (19)$$

where: H_t = Average number of daylight hours in a day for the particular month
e_t = Saturated water vapor pressure in millibars
T_t = Average daily temperature on day t

Saturated vapor pressure is governed by the equation for all $T_t > 0$:

$$e_t = 33.8639\left[\left(0.00738 T_t + 0.8072\right)^8 - 0.000019\left(1.8 T_t + 48\right) + 0.001316\right] \quad (20)$$

Finally, the discharge from the low saturated zone to the stream system and deep saturated zones are as follows:

$$G_t = r S_t \quad (21)$$

and

$$D_t = s S_t \quad (22)$$

where: r = Groundwater recession coefficient
s = Groundwater seepage coefficient

Groundwater Nutrient Loading

Groundwater nutrient loading is all in the dissolved phase and is computed using the equation:

$$DG_m = 0.1 C_g AT \sum_{t=1}^{d_m} G_t \tag{21}$$

where: DG_m = Groundwater nutrient load in month m on day t in kg
C_g = Concentration of nutrient in ground water (mg/L)
AT = Total watershed area in ha
G_t = Groundwater discharge into stream on day t in cm

Septic System Loads

Septic system loads on the watershed are calculated by summing the septic system loads from three sources (A common fourth septic system type, ponded systems, were omitted because of ponded systems do not exist in and around Springfield Township). The total septic load is:

$$DS_m = DS_n + DS_s + DS_d \tag{22}$$

where: DS_m = Total septic system nutrient load in kg
DS_n = Normal discharge septic system load in kg
DS_s = Short circuited discharge septic system load in kg
DS_d = Direct discharge septic system load in kg

Normal discharge septic systems only yield nitrogen pollutant loads to the watershed. These loads are a function of the nutrient effluent load and the uptake by vegitation above the system.

$$SL_n = 0.001 a_n d_m \left(e - u_m\right) \tag{23}$$

where: SL_n = nitrogen nutrient load to the groundwater system in kg
a_n = Population served by the normal discharge system in month m
d_m = number of days in month m
e = per capita nutrient load in septic tank effluent in g/day
u_m = per capita daily nutrient uptake by vegitation above septic system in month m

Since normal discharge septic systems are disctanced far from stream systems, the amount of nutrient discharge in month m (in kg) is proportionate to the amount of ground water discharge in month m compared to the annual groundwater discharge. The monthly discharge from normal discharge septic systems is:

$$DS_n = \frac{GR_m \sum_{m=1}^{12} SL_n}{\sum_{m=1}^{12} GR_m} \qquad (24)$$

where: GR_m = Groundwater discharge into the stream system in month m

Short circuit septic systems are governed by plant uptake and contain both phosphorus and nitrogen loads. The governing equation for the nutrient loads of these systems is:

$$DS_s = 0.001 a_s d_m (e - u_m) \qquad (25)$$

Direct discharge systems have no plant uptake and discharge directly into the stream system. They are rarely found but use the following loading function:

$$DS_d = 0.001 a_d d_m e \qquad (26)$$

MODELLING PROCEDURE

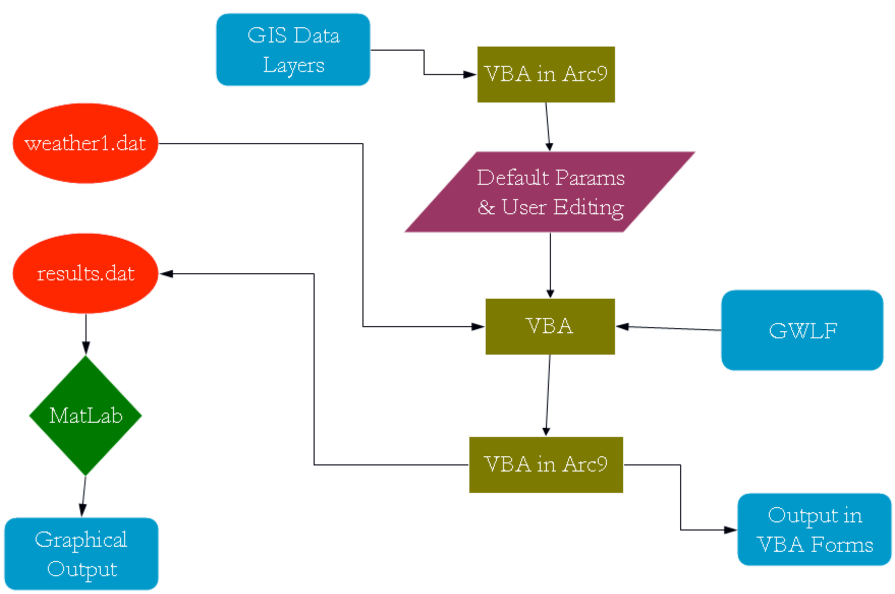


Figure 3 – General flow chart of model structure

Model Overview

Parameters for the model are gathered from three different locations as shown in Table 1. A general flow chart for the whole model is shown in Figure 3. The model accesses weather1.dat for all needed daily weather information. All parameters that vary significantly over the Springfield Township are stored in the global data layer and are accessed by the user when he/she specifies the watershed to be analyzed. Lastly, there are various parameters that have stored default values within the VBA code. When the program transitions into the editing parameters phase, the parameters from the global data layer and the stored default values are combined into editable forms where the user can adjust parameters as needed. These editable forms pass all of the parameters back to VBA to run the GWLF model. Finally, the model is told to run and output is displayed in a tabbed form and also passed to the user's workspace in a results.dat file that is easily run through three short Matlab scripts to display graphical results of the averaged annual hydrologic, sediment, and nutrient pollutant loadings.

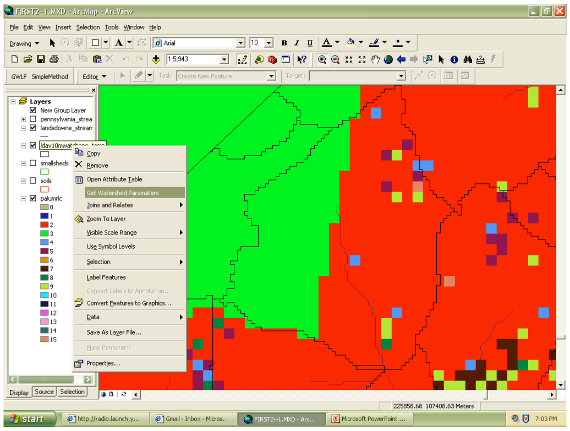
weather1.dat	# Days in Month – d_m Daily Average Temp (C) – T_t Daily Precipitation (cm) – R_t
Global Data Layer	Land Use Area (ha) – AR_k Initial Unsat Storage (cm) – U_t Initial Sat Storage (cm) - S_t Erosion Coefficients (L/day) – a_t Recess Coefficients (L/day) Sed Delivery Ratio – DR Sed Accum Factor Septic System Count Nutrient Point Source Loads (kg) Ground Water Nutrient Concentrations (mg/L) – C_g
Default (Stored in VBA)	Nutrient Plant Uptake (g/day) – u_m Per Capita Septic Tank Effluent (g/day) – e Rural Runoff Nutrient Concentrations (mg/L) - Cd Urban Nutrient Accumulation Rates (kg/ha/day) – n_{kt} Seepage Coefficient (L/day) – s CN and KLSCP for Each Land Use Average Hours of Daylight

Table 1 – Parameter gathering locations for the program

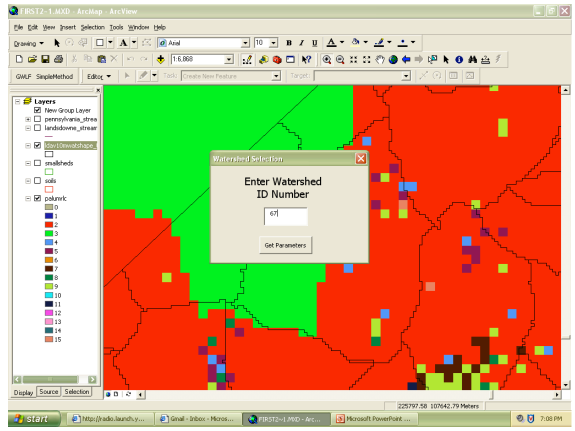
The following is a step by step how-to-use guide for the program.

1. Get Parameters

Prior to beginning the simulation, the user must first have loaded the 'ldav10mwatshape_temp' basins layer. This is the global data layer required to run the program. The user uses the identify button from the 'Tools' toolbar to selected the watershed and obtain the 'Watershed ID Number' that is listed under the 'fid' category in the 'Identify Results' window. Next, close the 'Identify Results' window and right click on the 'ldav10mwatshape_temp' layer. From the drop down menu select the 'Get Watershed Parameters' button (Figure 4). A form entitled 'Watershed Selection' will appear for the user to enter the 'Watershed ID Number' previously obtained. The 'Get Parameters' button is selected and now the program moves into the user editing phase.



(A)

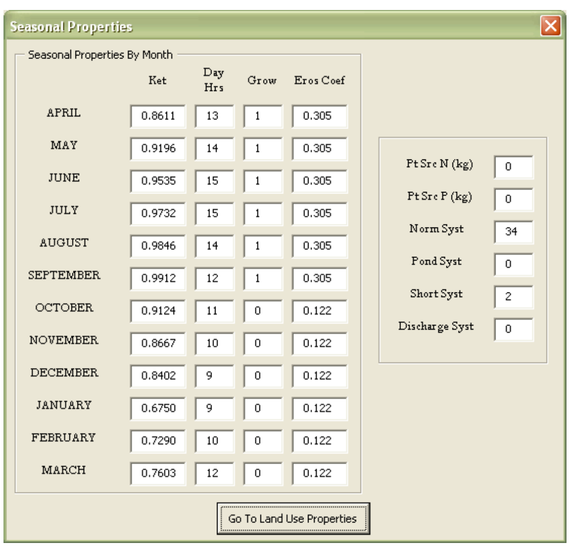


(B)

Figure 4 A & B – Get Watershed Parameters feature button and Watershed ID Number entry form

User Editing Phase

The user editing phase consists of a series of five forms that allow the user to change any parameters that he/she desires. Figures 5 A through E are the five editing forms in the order that they appear: 'Seasonal Properties,' 'Land Use Properties,' 'Hydrologic Properties,' 'Land Use Nutrient Coefficients,' and 'Hydrologic Nutrient Properties.' When the user finishes with the 'Hydrologic Nutrient Properties' form and clicks the 'OK' button the parameters are passed to GWLF with the weater1.dat file and the model is run.



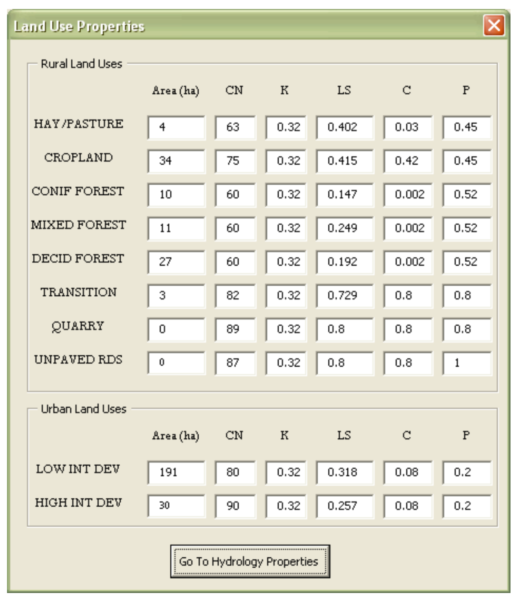
Seasonal Properties

Seasonal Properties By Month

	Ket	Day Hrs	Grow	Eros Coef
APRIL	0.8611	13	1	0.305
MAY	0.9196	14	1	0.305
JUNE	0.9535	15	1	0.305
JULY	0.9732	15	1	0.305
AUGUST	0.9846	14	1	0.305
SEPTEMBER	0.9912	12	1	0.305
OCTOBER	0.9124	11	0	0.122
NOVEMBER	0.8667	10	0	0.122
DECEMBER	0.8402	9	0	0.122
JANUARY	0.6750	9	0	0.122
FEBRUARY	0.7290	10	0	0.122
MARCH	0.7603	12	0	0.122

Pt Src N (kg)	0
Pt Src P (kg)	0
Norm Syst	34
Pond Syst	0
Short Syst	2
Discharge Syst	0

Go To Land Use Properties



Land Use Properties

Rural Land Uses

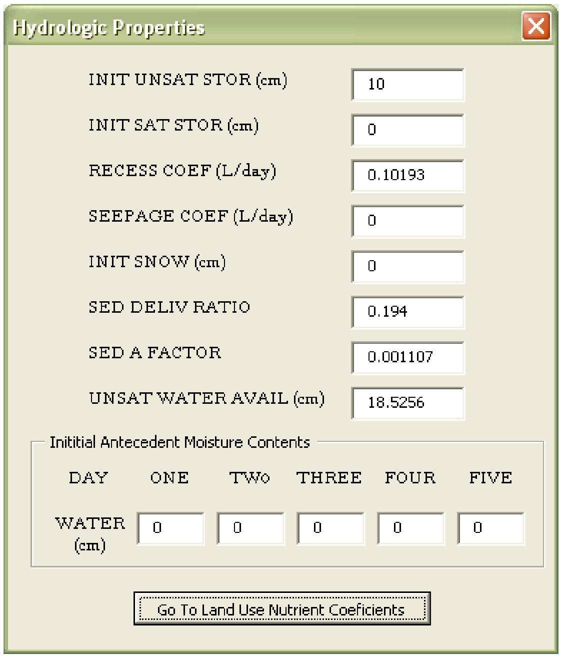
	Area (ha)	CN	K	LS	C	P
HAY/PASTURE	4	63	0.32	0.402	0.03	0.45
CROPLAND	34	75	0.32	0.415	0.42	0.45
CONIF FOREST	10	60	0.32	0.147	0.002	0.52
MIXED FOREST	11	60	0.32	0.249	0.002	0.52
DECID FOREST	27	60	0.32	0.192	0.002	0.52
TRANSITION	3	82	0.32	0.729	0.8	0.8
QUARRY	0	89	0.32	0.8	0.8	0.8
UNPAVED RDS	0	87	0.32	0.8	0.8	1

Urban Land Uses

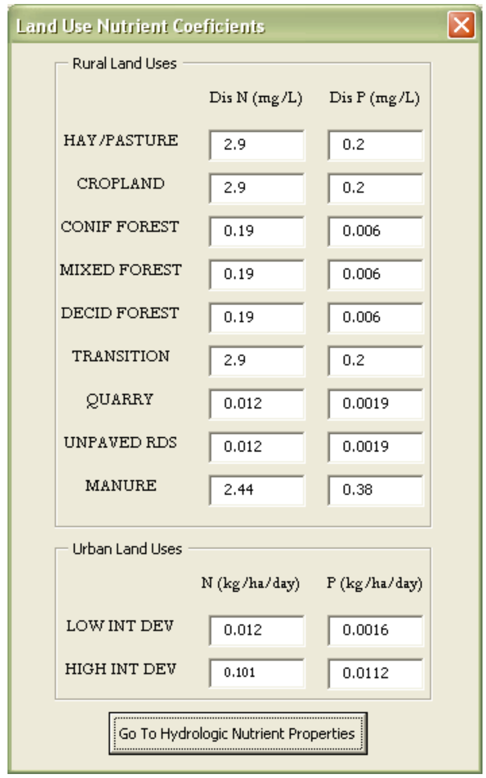
	Area (ha)	CN	K	LS	C	P
LOW INT DEV	191	80	0.32	0.318	0.08	0.2
HIGH INT DEV	30	90	0.32	0.257	0.08	0.2

Go To Hydrology Properties

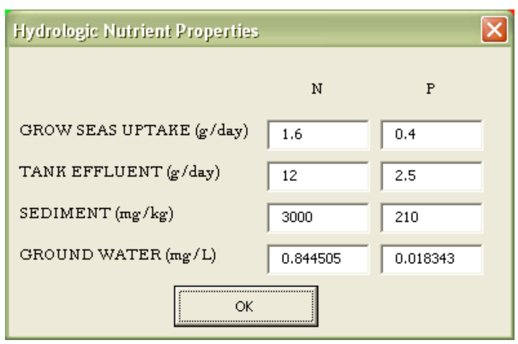
(A) (B)



(C)



(D)



(E)

Figure 5 A through E – User Editing Phase Forms

Run GWLF

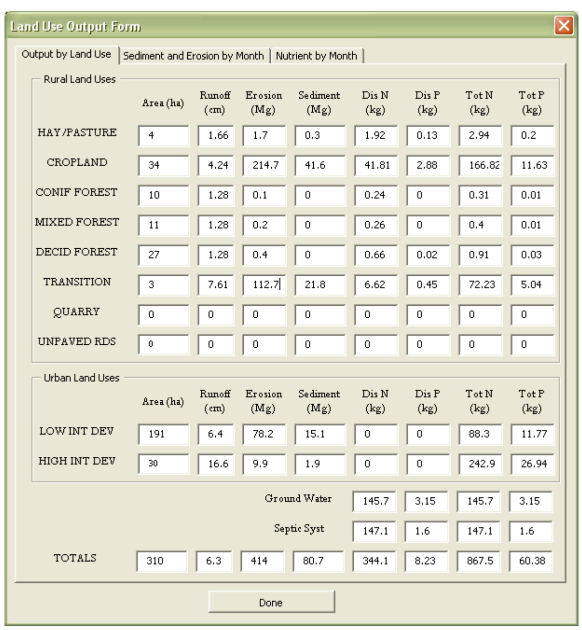
For the Run GWLF phase to run, a formatted weather1.dat file and a results.dat file must first be made and placed in the intended user's workspace. The weather1.dat file must be in the form of the text shown in Figure 6. The results.dat file does not need to be formatted because the program will write over the existing contents of the file.

```
30
12,0.0
10,0.0
10,2.7
7,0.1
4,0.6
1,2.8
-3,0.0
1,0.0
2,0.2
5,0.0
7,0.0
6,0.0
13,0.0
11,0.0
10,0.0
14,0.0
21,0.9
15,0.3
12,0.0
13,0.0
13,0.0
10,0.0
10,0.0
15,0.0
16,0.0
16,3.0
16,2.7
13,0.1
12,0.0
15,0.0
```

Figure 6 – weather1.dat file

Output Phase

The model output phase has two parts; tabbed output forms and Matlab graphical output. The tabbed output is shown immediately after the user clicks 'OK' on the 'Hydrologic Nutrient Properties' form in the user editing phase. This form is arranged by averaged annual loads according to land use, yearly averaged monthly sediment loads, and yearly averaged monthly nutrient loads (Figure 7). When the user is finished viewing this form the program can be terminated by clicking 'Done.'


Land Use Output Form

Output by Land Use | Sediment and Erosion by Month | Nutrient by Month

Rural Land Uses

	Area (ha)	Runoff (cm)	Erosion (Mg)	Sediment (Mg)	Dis N (kg)	Dis P (kg)	Tot N (kg)	Tot P (kg)
HAY/PASTURE	4	1.66	1.7	0.3	1.92	0.13	2.94	0.2
CROPLAND	34	4.24	214.7	41.6	41.81	2.88	166.82	11.63
CONIF FOREST	10	1.28	0.1	0	0.24	0	0.31	0.01
MIXED FOREST	11	1.28	0.2	0	0.26	0	0.4	0.01
DECID FOREST	27	1.28	0.4	0	0.66	0.02	0.91	0.03
TRANSITION	3	7.61	112.7	21.8	6.62	0.45	72.23	5.04
QUARRY	0	0	0	0	0	0	0	0
UNPAVED RDS	0	0	0	0	0	0	0	0

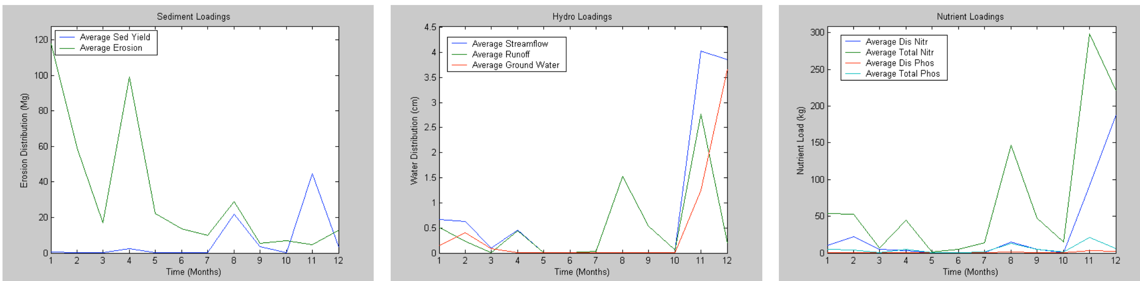
Urban Land Uses

	Area (ha)	Runoff (cm)	Erosion (Mg)	Sediment (Mg)	Dis N (kg)	Dis P (kg)	Tot N (kg)	Tot P (kg)
LOW INT DEV	191	6.4	78.2	15.1	0	0	88.3	11.77
HIGH INT DEV	30	16.6	9.9	1.9	0	0	242.9	26.94
Ground Water					145.7	3.15	145.7	3.15
Septic Syst					147.1	1.6	147.1	1.6
TOTALS	310	6.3	414	80.7	344.1	8.23	867.5	60.38

Done

Figure 7 – Tabbed output form

Also, the user has the option to run any of the three Matlab scripts (GWLFHydro.m, GWLFNutr.m, and GWLFRunoff.m) to see the hydrologic, nutrient, and sediment loads on the watershed in the form of averaged annual monthly loads in graphical format. Figure 8 is an example of the three graphs prepared by the Matlab scripts.



(A) **(B)** **(C)**

Figure 8 A through C – Matlab output graphs for GWLF model

MODEL VERIFICATION

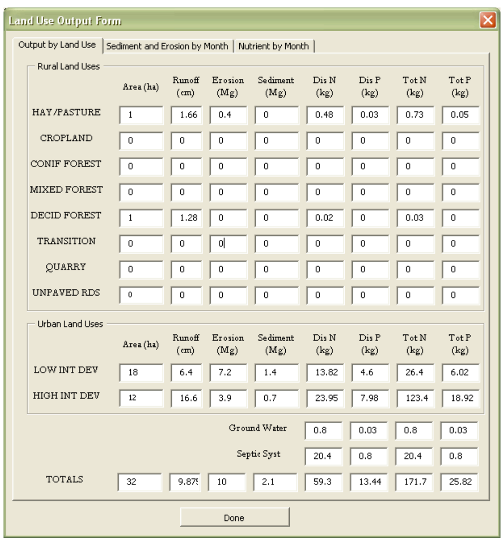
The model was compared to real monitoring data from the month of July 2004 at site P30I (Figure 11 shows the location of the site within Springfield Township). Event mean concentrations were obtained for the month of July by calculating flow weighted nutrient concentrations according to equation (27).

$$EMC = \frac{\sum_{t=0}^{T} C_t G_t}{G_T} \qquad (27)$$

where: C_t = Instantaneous pollutant concentration in mg/L
G_t = Instantaneous stream flow in m^3
G_T = Total stream flow for entire storm event in m^3
T = Time elapsed for entire storm event

Table 2 shows the calculated event mean concentrations for site P30I in July 2004. The data is compared to AVGWLF and Arc9 GWLF and show improvements in the utility of Arc9 GWLF in small urban watersheds. The watershed above site P30I is predominately urbanized development (38% low intensity development, 56% high intensity development, and 6% other rural land usages). Previous implementations of GWLF did not include dissolved phase urban nutrient loading, but these loading functions have been added to Arc9 GLWF. When AVGWLF is used to simulate site P30I in July 2004, the only dissolved phase urban loading is a result of groundwater flow and septic system loads. As a result, there is a large increase in dissolved nutrient loading in urban watersheds on the order two magnitudes. While this is an improvement to the model, future calibration of the model will be required with respect to the results shown. The model was only compared with pollutant loading during the month of July which is a low runoff month when compared to most winter months. For this reason the overall contribution of summer months to yearly pollutant loads is small compared to the winter months.

To more accurately calibrate the model, monitoring data for an entire year would be required to observe seasonal fluctuations in the model and real data. Figure 10 displays the seasonal fluctuation that occurs as a result of varying erosivities, daylight hours, and temperatures across seasons. As runoff values increase in the winter months, so does pollutant loading in the winter months. One of the largest problems encountered in this project was the lack of seasonal data to calibrate the model. Since the only data available was from the month on July 2004, the model was only roughly validated. Future validation would require more monitoring data from other months of the year.


Land Use Output Form

Output by Land Use | Sediment and Erosion by Month | Nutrient by Month

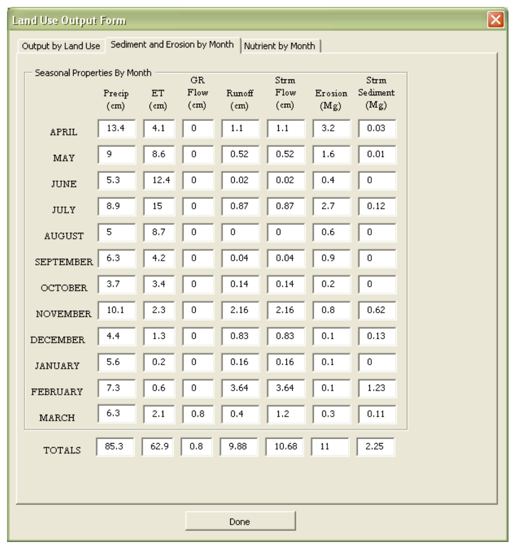
Rural Land Uses

	Area (ha)	Runoff (cm)	Erosion (Mg)	Sediment (Mg)	Dis N (kg)	Dis P (kg)	Tot N (kg)	Tot P (kg)
HAY/PASTURE	1	1.66	0.4	0	0.48	0.03	0.73	0.05
CROPLAND	0	0	0	0	0	0	0	0
CONIF FOREST	0	0	0	0	0	0	0	0
MIXED FOREST	0	0	0	0	0	0	0	0
DECID FOREST	1	1.28	0	0	0.02	0	0.03	0
TRANSITION	0	0	0	0	0	0	0	0
QUARRY	0	0	0	0	0	0	0	0
UNPAVED RDS	0	0	0	0	0	0	0	0

Urban Land Uses

	Area (ha)	Runoff (cm)	Erosion (Mg)	Sediment (Mg)	Dis N (kg)	Dis P (kg)	Tot N (kg)	Tot P (kg)
LOW INT DEV	18	6.4	7.2	1.4	13.82	4.6	26.4	6.02
HIGH INT DEV	12	16.6	3.9	0.7	23.95	7.98	123.4	18.92
Ground Water					0.8	0.03	0.8	0.03
Septic Syst					20.4	0.8	20.4	0.8
TOTALS	32	9.87	10	2.1	59.3	13.44	171.7	25.82

Done


Land Use Output Form

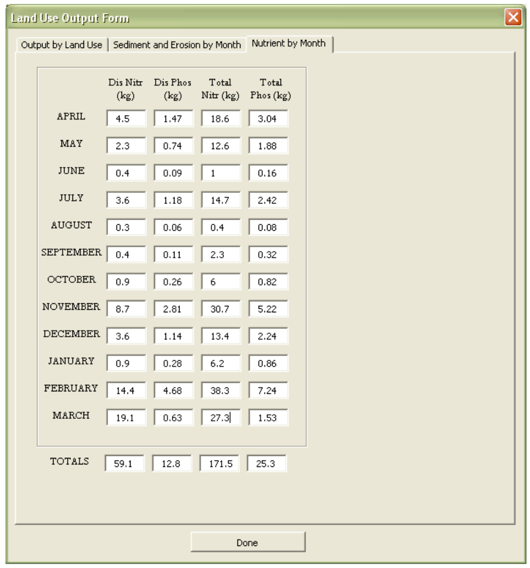
Output by Land Use | Sediment and Erosion by Month | Nutrient by Month

Seasonal Properties By Month

	Precip (cm)	ET (cm)	GR Flow (cm)	Runoff (cm)	Strm Flow (cm)	Erosion (Mg)	Strm Sediment (Mg)
APRIL	13.4	4.1	0	1.1	1.1	3.2	0.03
MAY	9	8.6	0	0.52	0.52	1.6	0.01
JUNE	5.3	12.4	0	0.02	0.02	0.4	0
JULY	8.9	15	0	0.87	0.87	2.7	0.12
AUGUST	5	8.7	0	0	0	0.6	0
SEPTEMBER	6.3	4.2	0	0.04	0.04	0.9	0
OCTOBER	3.7	3.4	0	0.14	0.14	0.2	0
NOVEMBER	10.1	2.3	0	2.16	2.16	0.8	0.62
DECEMBER	4.4	1.3	0	0.83	0.83	0.1	0.13
JANUARY	5.6	0.2	0	0.16	0.16	0.1	0
FEBRUARY	7.3	0.6	0	3.64	3.64	0.1	1.23
MARCH	6.3	2.1	0.8	0.4	1.2	0.3	0.11
TOTALS	85.3	62.9	0.8	9.88	10.68	11	2.25

Done

(A) (B)



Land Use Output Form

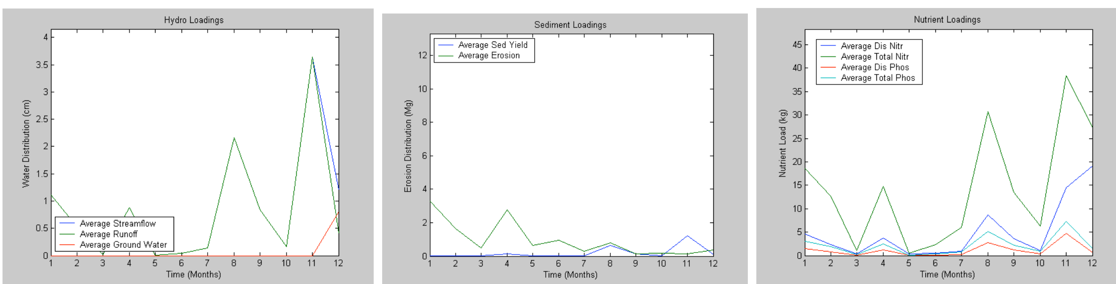
Output by Land Use | Sediment and Erosion by Month | Nutrient by Month

	Dis Nitr (kg)	Dis Phos (kg)	Total Nitr (kg)	Total Phos (kg)
APRIL	4.5	1.47	18.6	3.04
MAY	2.3	0.74	12.6	1.88
JUNE	0.4	0.09	1	0.16
JULY	3.6	1.18	14.7	2.42
AUGUST	0.3	0.06	0.4	0.08
SEPTEMBER	0.4	0.11	2.3	0.32
OCTOBER	0.9	0.26	6	0.82
NOVEMBER	8.7	2.81	30.7	5.22
DECEMBER	3.6	1.14	13.4	2.24
JANUARY	0.9	0.28	6.2	0.86
FEBRUARY	14.4	4.68	38.3	7.24
MARCH	19.1	0.63	27.3	1.53
TOTALS	59.1	12.8	171.5	25.3

Done

(C)

Figure 9 A through C – Arc9 GWLF Output for P30I in July 2004



(A) **(B)** **(C)**

Figure 10 A through C – Arc9 GWLF Output graphs for P30I in July 2004

	GWLF Mean Conc	Arc9 GWLF Mean Conc	Measured Event Mean Conc
Sediment (mg/L)	83	83	40 – 60
Dissolved N (mg/L)	0.2	3.6	1.0 – 1.2
Dissolved P (mg/L)	0.04	1.18	0.2 - 0 .3

Table 2 – Summarized Results of model comparison with monitored loadings at site P20I in July 2004

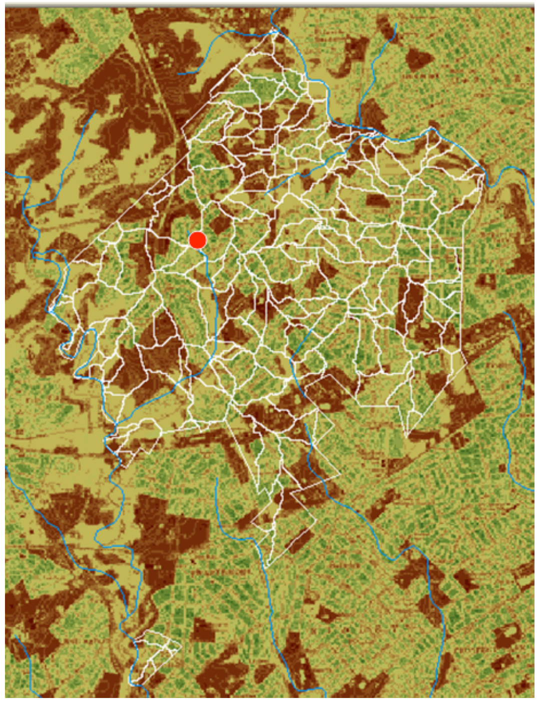


Figure 11 – Monitoring site P30I2 in Whiskey Run watershed

FUTURE WORK

Future work on the model should include calibration of the model over seasonally fluctuating monitoring data. This could include field verification of the land uses in Springfield, event mean concentration calculation, or the implementation of a more advanced data layer structure. A multi-layer model gathering information more readily from other sources could increase the power and potential accuracy of the model.

With respect to the future work of Dr. Arthur E. McGarity, this model could be used as a pollutant loading model within a BMP optimization model. This BMP model would take the pollutant loading and suggest BMP's to improve the quality of stormwater runoff within a watershed. A possible future flow chart of this model is shown in Figure 12.

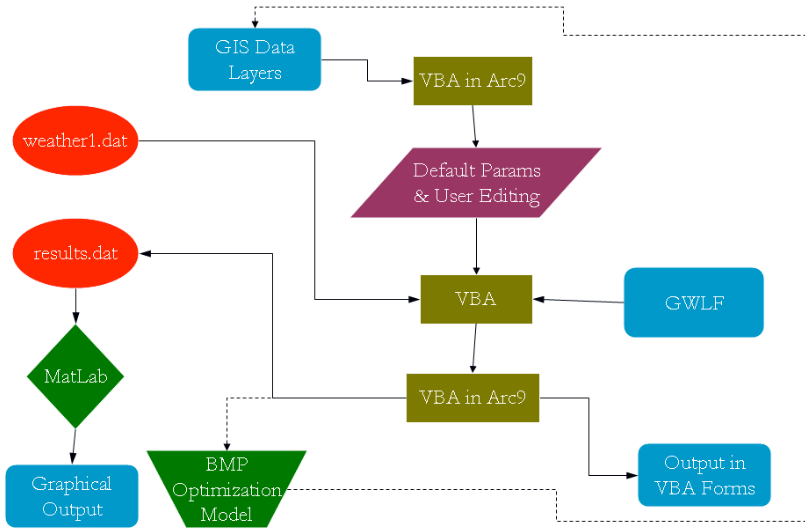


Figure 12 – Future model structure

REFERENCES

Evans, Barry M., David W. Lehning, Kenneth J. Corradini, Gary W. Petersen, Egide Nizeyimana, James M. Hamlet, Paul D. Robilaro, and Rick L. Day. *A Comprehensive GIS-Based Modeling Approach for Predicting Nutrient Loads in Watersheds.* Journal of Spacial Hydrology Volume 2 No 2.

Haith, Douglas A., Ross Mandel & Ray Shyan Wu. *GWLF: Generalized Watershed Loading Functions Version 2.0*. December, 15 1992: Department of Agricultural & Biological Engineering. Cornell University.

PLOAD version 3.0. United States Environmental Protection Agency.

ACKNOWLEDGEMENTS

PADEP and NOAA Coastal Zone Program
Howard Hughes Medical Institute
Professor Arthur E. McGarity – Swarthmore College Engineering Department
Paul Horna
Frank Dowman
Kevin Kane – Springfield Township Engineer
Steven Bhardwaj, Jorge Aguilera, Chelsea Rogers

APPENDIX A – PROGRAM SCRIPT

```
Public Sub GetParam_Click()

    Dim gisTotalArea, gisHiIntDev, gisLowIntDev, gisHayPast, gisCropland, gisConifFor,
gisMixedFor, gisDecidFor, gisTransition, gisQuarry, gisUnpaved As Integer
    Dim gisRecessCoef, gisSedAccumFact, gisUnsatWatAvail, gisErosGrow, gisErosNoGrow,
gisSedDelivRat As Double
    Dim gisGrWatN, gisGrWatP As Double
    Dim gisNormSys, gisPondSys, gisShortSys, gisDischargeSys, gisPtSrcN, gisPtSrcP As Integer
    Dim WatershedID As Integer

    frmShedSelect.Show

    WatershedID = frmShedSelect.TextBox1.Value

    Dim pMxDoc As IMxDocument
    Set pMxDoc = ThisDocument

    Dim pFLayer As IFeatureLayer
    Set pFLayer = pMxDoc.ContextItem

    Dim pFClass As IFeatureClass
    Set pFClass = pFLayer.FeatureClass

    Dim pFields As IFields
    Set pFields = pFClass.Fields

    Dim intArea As Integer
    intArea = pFields.FindField("Area")

    Dim intHighInt As Integer
    intHighInt = pFields.FindField("HI_INT_DEV")

    Dim intLowInt As Integer
    intLowInt = pFields.FindField("LO_INT_DEV")

    Dim intHayPast As Integer
    intHayPast = pFields.FindField("HAY_PAST")

    Dim intCropland As Integer
    intCropland = pFields.FindField("CROPLAND")

    Dim intConifFor As Integer
    intConifFor = pFields.FindField("CONIF_FOR")

    Dim intMixedFor As Integer
    intMixedFor = pFields.FindField("MIXED_FOR")

    Dim intDecidFor As Integer
    intDecidFor = pFields.FindField("DECID_FOR")

    Dim intTransition As Integer
    intTransition = pFields.FindField("TRANSITION")
```

```
Dim intQuarry As Integer
intQuarry = pFields.FindField("QUARY")

Dim intUnpaved As Integer
intUnpaved = pFields.FindField("UNPAVED")

Dim dblRecessCoef As Double
dblRecessCoef = pFields.FindField("RECESSCOEF")

Dim dblSedAccumFact As Double
dblSedAccumFact = pFields.FindField("SEDAFACT")

Dim dblUnsatWatAvail As Double
dblUnsatWatAvail = pFields.FindField("UNSATWATAV")

Dim dblErosGrow As Double
dblErosGrow = pFields.FindField("EROSGROW")

Dim dblErosNoGrow As Double
dblErosNoGrow = pFields.FindField("EROSNOGROW")

Dim intNormSys As Integer
intNormSys = pFields.FindField("NORMSYS")

Dim intPondSys As Integer
intPondSys = pFields.FindField("PONDSYS")

Dim intShortSys As Integer
intShortSys = pFields.FindField("SHORTSYS")

Dim intDischargeSys As Integer
intDischargeSys = pFields.FindField("DISCHASYS")

Dim intPtSrcN As Integer
intPtSrcN = pFields.FindField("PTSRCN")

Dim intPtSrcP As Integer
intPtSrcP = pFields.FindField("PTSRCP")

Dim dblGrWatN As Double
dblGrWatN = pFields.FindField("GRWATN")

Dim dblGrWatP As Double
dblGrWatP = pFields.FindField("GRWATP")

Dim dblSedDelRat As Double
dblSedDelRat = pFields.FindField("SEDELRATIO")

Dim intID As Integer
intID = pFields.FindField("FID")

Dim pFCursor As IFeatureCursor
Set pFCursor = pFClass.Update(Nothing, True)

Dim pFeature As IFeature
```

```
    Set pFeature = pFCursor.NextFeature

    If WatershedID < 0 Or WatershedID > 127 Then
        MsgBox ("NOT A VALID WATERSHED ID NUMBER!")
    End If

    Do Until pFeature Is Nothing
        If pFeature.Value(intID) = WatershedID Then
            gisTotalArea = pFeature.Value(intArea)
            gisHiIntDev = pFeature.Value(intHighInt)
            gisLowIntDev = pFeature.Value(intLowInt)
            gisHayPast = pFeature.Value(intHayPast)
            gisCropland = pFeature.Value(intCropland)
            gisConifFor = pFeature.Value(intConifFor)
            gisMixedFor = pFeature.Value(intMixedFor)
            gisDecidFor = pFeature.Value(intDecidFor)
            gisTransition = pFeature.Value(intTransition)
            gisQuarry = pFeature.Value(intQuarry)
            gisUnpaved = pFeature.Value(intUnpaved)
            gisRecessCoef = pFeature.Value(dblRecessCoef)
            gisSedAccumFact = pFeature.Value(dblSedAccumFact)
            gisUnsatWatAvail = pFeature.Value(dblUnsatWatAvail)
            gisErosGrow = pFeature.Value(dblErosGrow)
            gisErosNoGrow = pFeature.Value(dblErosNoGrow)
            gisNormSys = pFeature.Value(intNormSys)
            gisPondSys = pFeature.Value(intPondSys)
            gisShortSys = pFeature.Value(intShortSys)
            gisDischargeSys = pFeature.Value(intDischargeSys)
            gisPtSrcN = pFeature.Value(intPtSrcN)
            gisPtSrcP = pFeature.Value(intPtSrcP)
            gisGrWatN = pFeature.Value(dblGrWatN)
            gisGrWatP = pFeature.Value(dblGrWatP)
            gisSedDelivRat = pFeature.Value(dblSedDelRat)
        End If

        pFCursor.UpdateFeature pFeature
        Set pFeature = pFCursor.NextFeature
    Loop

frmLandUseProps.TextBox1.Value = gisHayPast
frmLandUseProps.TextBox2.Value = gisCropland
frmLandUseProps.TextBox3.Value = gisConifFor
frmLandUseProps.TextBox4.Value = gisMixedFor
frmLandUseProps.TextBox5.Value = gisDecidFor
frmLandUseProps.TextBox6.Value = gisTransition
frmLandUseProps.TextBox7.Value = gisQuarry
frmLandUseProps.TextBox8.Value = gisUnpaved
frmLandUseProps.TextBox51.Value = gisLowIntDev
frmLandUseProps.TextBox52.Value = gisHiIntDev

frmSeasonProps.TextBox25.Value = gisErosGrow
frmSeasonProps.TextBox26.Value = gisErosGrow
frmSeasonProps.TextBox27.Value = gisErosGrow
frmSeasonProps.TextBox28.Value = gisErosGrow
frmSeasonProps.TextBox29.Value = gisErosGrow
frmSeasonProps.TextBox30.Value = gisErosGrow
```

```
frmSeasonProps.TextBox31.Value = gisErosNoGrow
frmSeasonProps.TextBox32.Value = gisErosNoGrow
frmSeasonProps.TextBox45.Value = gisErosNoGrow
frmSeasonProps.TextBox46.Value = gisErosNoGrow
frmSeasonProps.TextBox47.Value = gisErosNoGrow
frmSeasonProps.TextBox48.Value = gisErosNoGrow

frmHydroNutr.TextBox3.Value = gisGrWatN
frmHydroNutr.TextBox11.Value = gisGrWatP

frmSeasonProps.TextBox49.Value = gisNormSys
frmSeasonProps.TextBox52.Value = gisPondSys
frmSeasonProps.TextBox53.Value = gisShortSys
frmSeasonProps.TextBox61.Value = gisDischargeSys
frmSeasonProps.TextBox50.Value = gisPtSrcN
frmSeasonProps.TextBox51.Value = gisPtSrcP

frmHydroProps.TextBox5.Value = gisSedDelivRat
frmHydroProps.TextBox7.Value = gisUnsatWatAvail
frmHydroProps.TextBox4.Value = gisRecessCoef
frmHydroProps.TextBox8.Value = gisSedAccumFact

End Sub

‘////////////////////////////////////////////////////////////////

Private Sub CommandButton1_Click()

frmHydroNutr.Hide

rungwlf

‘//////////////////////////////////////////////////////////////////////////////

End Sub

Public Sub rungwlf()

Dim J As Integer ' month incrementor
Dim I As Integer ' number of days in month incrementor
Dim DayMonth As Integer ' keeps track of number of days in month
Dim L As Integer ' land usage incrementor
Dim NLU As Integer ' total number of land uses
Dim Y As Integer ' total number of years in simulation

Dim DailyTemp As Double ' average daily temperature
Dim DailyPrec As Double ' total daily precipitation

Dim Grow(12) As Integer ' turns growing season on and off
Dim melt As Double ' indicates snow melt or no snow melt
Dim year As Double ' year

Dim Water As Double ' total water
Dim Area(10) As Double  ' Area area for each land usage
```

```
Dim Temp(15, 12, 31) As Double ' temperature array for entire simulation
Dim Prec(15, 12, 31) As Double ' precipitation array for entire simulation

Dim DaysHrs(12) As Double ' number of daylight hours in each day

Dim NRur As Double ' number of rural land uses (default = 8)

Dim RurEros As Double ' rural erosion variable
Dim Erosiv As Double ' erosivity
Dim KLSCP(10) As Double ' KLSCP values for each land use

Dim CV(12) As Double ' cover coeficient for evapotrans calc
Dim Acoef(12) As Double ' erosion coefficient for erosivity calc
Dim AntMoist(5) As Double ' five day antecedent moisture contents

Dim QTotal As Double ' total runoff
Dim AreaTotal As Double ' total watershed area

' . . . local variables
Dim CNum As Double ' local curve number
Dim Retention As Double ' local retention value
Dim Washoff As Double ' local erosion

Dim AMC5 As Double ' total antecedent moisture content

Dim Qrun As Double ' local runoff

Dim CN(3, 15) As Double 'curven number array

Dim DimYrs As Double ' number of years in simulation

' local septic system arrays
Dim MonthPondNitr(12) As Double ' monthly nitr discharge from pond septic systems
Dim MonthPondPhos(12) As Double ' monthly phos discharge from pond septic systems
Dim MonthNormNitr(12) As Double ' monthly nitr discharge from normal discharge septic systems
Dim MonthShortNitr(12) As Double ' monthly nitr discharge from short discharge septic systems
Dim MonthShortPhos(12) As Double ' monthly phos discharge from short discharge septic
systems
Dim MonthDischargeNitr(12) As Double ' monthly nitr discharge from discharge septic systems
Dim MonthDischargePhos(12) As Double ' monthly phos discharge from discharge septic
systems

Dim AvStreamflow(12) As Double ' average monthly stream flow
Dim AvPrecipitation(12) As Double ' average monthly precip
Dim AvEvapoTrans(12) As Double ' average monthly et
Dim AvGroundWater(12) As Double ' average monthly ground water flow
Dim AvRunoff(12) As Double ' average monthly flow
Dim AvErosion(12) As Double ' avarage monthly erosion
Dim AvSedyield(12) As Double ' average monthly sediment yield
Dim AvGroundNitr(12) As Double ' average monthly dissolved nitr from ground water
Dim AvGroundPhos(12) As Double ' average monthly dissolved phos from ground water
Dim AvDisNitr(12) As Double ' average monthly dissolved nitr
Dim AvTotNitr(12) As Double ' average monthly total nitr
Dim AvDisPhos(12) As Double ' average monthly dissolved phos
Dim AvTotPhos(12) As Double ' average monthly total phos
Dim AvLuRunoff(25) As Double ' average annual runoff for land use
```

```
Dim AvLuErosion(25) As Double ' average annual erosion for land use
Dim AvLuDisNitr(25) As Double ' average annual dissolved nitr for land use
Dim AvLuTotNitr(25) As Double ' average annual total nitr for land use
Dim AvLuDisPhos(25) As Double ' average annual dissolved phos for land use
Dim AvLuTotPhos(25) As Double ' average annual total phos for land use
Dim AvLuSedYield(25) As Double ' average annual sediment yield for land use

Dim BSed(12) As Double ' bsed
Dim UrbanSed(25) As Double ' urbansediment
Dim ErosWashoff(25, 12) As Double ' erosion
Dim UrbErosWashoff(25, 12) As Double ' urban erosion unit variable
Dim QRunoff(25, 12) As Double ' runoff
Dim NumPondSys As Double ' number of pond septic systems
Dim NumNormalSys As Double ' number of normal discharge septic systems
Dim NumShortSys As Double ' number of short discharge septic systems
Dim NumDischargeSys As Double ' number of discharge septic systems

Dim NitrConc(25) As Double ' dissolved nitr in land use runoff
Dim PhosConc(25) As Double ' dissolved phos in land use runoff
Dim ManNitr As Double ' dissolved nitr in runoff from manured areas
Dim ManPhos As Double ' dissolved phos in runoff from manured areas
Dim GrNitrConc As Double ' dissolved nitr in ground water flow
Dim GrPhosConc As Double ' dissolved phos in ground water flow
Dim UrbanNitr(25) As Double ' urban solid nitr in land use
Dim UrbanPhos(25) As Double ' urban solid nitr in land use
Dim PointNitr(25) As Double ' point nitr in land use
Dim PointPhos(25) As Double ' point phos in land use

Dim Percolation As Double '
Dim MaxWaterCap As Double ' max water capacity in unsaturated storage zone
Dim UnsatStor As Double ' unsaturated storage zone capacity
Dim SatStor As Double ' saturated storage zone capacity
Dim SedelRatio As Double ' sediment delivary ratio
Dim SeepCoef As Double ' seepage coefficient
Dim RecessCoef As Double ' recess coefficient in ground water
Dim NitrSepticLoad As Double ' total nitr septic load
Dim PhosSepticLoad As Double ' total phos septic load
Dim PhosPlantUptake As Double ' phos uptake by vegitation over septic
Dim NitrPlantUptake As Double ' nitr uptake by vegitation over septic

Dim AnNormNitr As Double ' annual normal system septic nitr load

AreaTotal = 0

NYrs = 1 ' one year for simulation
NLU = 10 ' one land use is used

DimYrs = NYrs

ReDim StreamFlow(DimYrs, 12), Precipitation(DimYrs, 12), EvapoTrans(DimYrs, 12) As Double
ReDim Groundwater(DimYrs, 12), Runoff(DimYrs, 12) As Double
ReDim Erosion(DimYrs, 12), SedYield(DimYrs, 12) As Double
ReDim DaysMonth(DimYrs, 12) As Integer
ReDim GroundNitr(DimYrs, 12), GroundPhos(DimYrs, 12), DisNitr(DimYrs, 12) As Double
ReDim TotNitr(DimYrs, 12), DisPhos(DimYrs, 12), TotPhos(DimYrs, 12) As Double
ReDim LuRunoff(DimYrs, 25), LuErosion(DimYrs, 25), LuDisNitr(DimYrs, 25) As Double
```

```
ReDim LuTotNitr(DimYrs, 25), ludisphos(DimYrs, 25), LuTotPhos(DimYrs, 25) As Double
ReDim SedTrans(DimYrs, 12) As Double
ReDim SepticNitr(DimYrs), SepticPhos(DimYrs) As Double


'*******************************************
'coeficient assignments
'*******************************************

Acoef(1) = frmSeasonProps.TextBox25.Value
Acoef(2) = frmSeasonProps.TextBox26.Value
Acoef(3) = frmSeasonProps.TextBox27.Value
Acoef(4) = frmSeasonProps.TextBox28.Value
Acoef(5) = frmSeasonProps.TextBox29.Value
Acoef(6) = frmSeasonProps.TextBox30.Value
Acoef(7) = frmSeasonProps.TextBox31.Value
Acoef(8) = frmSeasonProps.TextBox32.Value
Acoef(9) = frmSeasonProps.TextBox45.Value
Acoef(10) = frmSeasonProps.TextBox46.Value
Acoef(11) = frmSeasonProps.TextBox47.Value
Acoef(12) = frmSeasonProps.TextBox48.Value

CV(1) = frmSeasonProps.TextBox1.Value
CV(2) = frmSeasonProps.TextBox2.Value
CV(3) = frmSeasonProps.TextBox3.Value
CV(4) = frmSeasonProps.TextBox4.Value
CV(5) = frmSeasonProps.TextBox5.Value
CV(6) = frmSeasonProps.TextBox6.Value
CV(7) = frmSeasonProps.TextBox7.Value
CV(8) = frmSeasonProps.TextBox8.Value
CV(9) = frmSeasonProps.TextBox34.Value
CV(10) = frmSeasonProps.TextBox35.Value
CV(11) = frmSeasonProps.TextBox36.Value
CV(12) = frmSeasonProps.TextBox37.Value

DaysHrs(1) = frmSeasonProps.TextBox9.Value
DaysHrs(2) = frmSeasonProps.TextBox10.Value
DaysHrs(3) = frmSeasonProps.TextBox11.Value
DaysHrs(4) = frmSeasonProps.TextBox12.Value
DaysHrs(5) = frmSeasonProps.TextBox13.Value
DaysHrs(6) = frmSeasonProps.TextBox14.Value
DaysHrs(7) = frmSeasonProps.TextBox15.Value
DaysHrs(8) = frmSeasonProps.TextBox16.Value
DaysHrs(9) = frmSeasonProps.TextBox38.Value
DaysHrs(10) = frmSeasonProps.TextBox39.Value
DaysHrs(11) = frmSeasonProps.TextBox40.Value
DaysHrs(12) = frmSeasonProps.TextBox41.Value

Grow(1) = frmSeasonProps.TextBox17.Value
Grow(2) = frmSeasonProps.TextBox18.Value
Grow(3) = frmSeasonProps.TextBox19.Value
Grow(4) = frmSeasonProps.TextBox20.Value
Grow(5) = frmSeasonProps.TextBox21.Value
Grow(6) = frmSeasonProps.TextBox22.Value
Grow(7) = frmSeasonProps.TextBox23.Value
Grow(8) = frmSeasonProps.TextBox24.Value
```

```
Grow(9) = frmSeasonProps.TextBox42.Value
Grow(10) = frmSeasonProps.TextBox43.Value
Grow(11) = frmSeasonProps.TextBox44.Value
Grow(12) = frmSeasonProps.TextBox45.Value

Area(1) = frmLandUseProps.TextBox1.Value
Area(2) = frmLandUseProps.TextBox2.Value
Area(3) = frmLandUseProps.TextBox3.Value
Area(4) = frmLandUseProps.TextBox4.Value
Area(5) = frmLandUseProps.TextBox5.Value
Area(6) = frmLandUseProps.TextBox6.Value
Area(7) = frmLandUseProps.TextBox7.Value
Area(8) = frmLandUseProps.TextBox8.Value
Area(9) = frmLandUseProps.TextBox51.Value
Area(10) = frmLandUseProps.TextBox52.Value

CN(2, 1) = frmLandUseProps.TextBox9.Value
CN(2, 2) = frmLandUseProps.TextBox10.Value
CN(2, 3) = frmLandUseProps.TextBox11.Value
CN(2, 4) = frmLandUseProps.TextBox12.Value
CN(2, 5) = frmLandUseProps.TextBox13.Value
CN(2, 6) = frmLandUseProps.TextBox14.Value
CN(2, 7) = frmLandUseProps.TextBox15.Value
CN(2, 8) = frmLandUseProps.TextBox16.Value
CN(2, 9) = frmLandUseProps.TextBox59.Value
CN(2, 10) = frmLandUseProps.TextBox60.Value

KLSCP(1) = frmLandUseProps.TextBox17.Value * frmLandUseProps.TextBox25.Value *
frmLandUseProps.TextBox33.Value * frmLandUseProps.TextBox41.Value
KLSCP(2) = frmLandUseProps.TextBox18.Value * frmLandUseProps.TextBox26.Value *
frmLandUseProps.TextBox34.Value * frmLandUseProps.TextBox42.Value
KLSCP(3) = frmLandUseProps.TextBox19.Value * frmLandUseProps.TextBox27.Value *
frmLandUseProps.TextBox35.Value * frmLandUseProps.TextBox43.Value
KLSCP(4) = frmLandUseProps.TextBox20.Value * frmLandUseProps.TextBox28.Value *
frmLandUseProps.TextBox36.Value * frmLandUseProps.TextBox44.Value
KLSCP(5) = frmLandUseProps.TextBox21.Value * frmLandUseProps.TextBox29.Value *
frmLandUseProps.TextBox37.Value * frmLandUseProps.TextBox45.Value
KLSCP(6) = frmLandUseProps.TextBox22.Value * frmLandUseProps.TextBox30.Value *
frmLandUseProps.TextBox38.Value * frmLandUseProps.TextBox46.Value
KLSCP(7) = frmLandUseProps.TextBox23.Value * frmLandUseProps.TextBox31.Value *
frmLandUseProps.TextBox39.Value * frmLandUseProps.TextBox47.Value
KLSCP(8) = frmLandUseProps.TextBox24.Value * frmLandUseProps.TextBox32.Value *
frmLandUseProps.TextBox40.Value * frmLandUseProps.TextBox48.Value
KLSCP(9) = frmLandUseProps.TextBox50.Value * frmLandUseProps.TextBox73.Value *
frmLandUseProps.TextBox81.Value * frmLandUseProps.TextBox89.Value
KLSCP(10) = frmLandUseProps.TextBox67.Value * frmLandUseProps.TextBox74.Value *
frmLandUseProps.TextBox82.Value * frmLandUseProps.TextBox90.Value

UnsatStor = frmHydroProps.TextBox1.Value
SatStor = frmHydroProps.TextBox2.Value
SeepCoef = frmHydroProps.TextBox3.Value
RecessCoef = frmHydroProps.TextBox4.Value
snow = frmHydroProps.TextBox6.Value
SedelRatio = frmHydroProps.TextBox5.Value
MaxWaterCap = frmHydroProps.TextBox7.Value
AntMoist(1) = frmHydroProps.TextBox9.Value
```

```
AntMoist(2) = frmHydroProps.TextBox10.Value
AntMoist(3) = frmHydroProps.TextBox11.Value
AntMoist(4) = frmHydroProps.TextBox12.Value
AntMoist(5) = frmHydroProps.TextBox13.Value

UrbanNitr(9) = frmLandUseNutr.TextBox51.Value
UrbanNitr(10) = frmLandUseNutr.TextBox52.Value
UrbanPhos(9) = frmLandUseNutr.TextBox59.Value
UrbanPhos(10) = frmLandUseNutr.TextBox60.Value
NitrConc(1) = frmLandUseNutr.TextBox1.Value
NitrConc(2) = frmLandUseNutr.TextBox2.Value
NitrConc(3) = frmLandUseNutr.TextBox3.Value
NitrConc(4) = frmLandUseNutr.TextBox4.Value
NitrConc(5) = frmLandUseNutr.TextBox5.Value
NitrConc(6) = frmLandUseNutr.TextBox6.Value
NitrConc(7) = frmLandUseNutr.TextBox7.Value
NitrConc(8) = frmLandUseNutr.TextBox8.Value
NitrConc(9) = 1.2
NitrConc(10) = 1.2
PhosConc(1) = frmLandUseNutr.TextBox9.Value
PhosConc(2) = frmLandUseNutr.TextBox10.Value
PhosConc(3) = frmLandUseNutr.TextBox11.Value
PhosConc(4) = frmLandUseNutr.TextBox12.Value
PhosConc(5) = frmLandUseNutr.TextBox13.Value
PhosConc(6) = frmLandUseNutr.TextBox14.Value
PhosConc(7) = frmLandUseNutr.TextBox15.Value
PhosConc(8) = frmLandUseNutr.TextBox16.Value
PhosConc(9) = 0.4
PhosConc(10) = 0.4
ManNitr = frmLandUseNutr.TextBox61.Value
ManPhos = frmLandUseNutr.TextBox62.Value

GrNitrConc = frmHydroNutr.TextBox3.Value
GrPhosConc = frmHydroNutr.TextBox11.Value
SedNitr = frmHydroNutr.TextBox4.Value
SedPhos = frmHydroNutr.TextBox12.Value
NitrPlantUptake = frmHydroNutr.TextBox1.Value
PhosPlantUptake = frmHydroNutr.TextBox9.Value
NumNormSys = frmSeasonProps.TextBox49.Value
NumPondSys = frmSeasonProps.TextBox52.Value
NumShortSys = frmSeasonProps.TextBox53.Value
NumDischargeSys = frmSeasonProps.TextBox61.Value
NitrSepticLoad = frmHydroNutr.TextBox2.Value
PhosSepticLoad = frmHydroNutr.TextBox10.Value

NRur = 8
melt = -1


For L = 1 To NLU
   AreaTotal = AreaTotal + Area(L)
   CN(1, L) = CN(2, L) / (2.334 - 0.01334 * CN(2, L))
   CN(3, L) = CN(2, L) / (0.4036 + 0.0059 * CN(2, L))
   If CN(3, L) > 100 Then CN(3, L) = 100
Next L
```

```
AMC5 = 0

For K = 1 To 5
   AMC5 = AMC5 + AntMoist(K)
Next K

' initialize variables used in summary file
For I = 0 To 12
   AvPrecipitation(I) = 0:  AvEvapoTrans(I) = 0:  AvGroundWater(I) = 0
   AvRunoff(I) = 0:  AvStreamflow(I) = 0:  AvErosion(I) = 0
   AvSedyield(I) = 0:  AvDisNitr(I) = 0:  AvTotNitr(I) = 0
   AvDisPhos(I) = 0:  AvTotPhos(I) = 0
   AvGroundNitr(I) = 0:  AvGroundPhos(I) = 0
Next I

For L = 1 To NLU
   AvLuRunoff(L) = 0: AvLuErosion(L) = 0:  AvLuDisNitr(L) = 0
   AvLuTotNitr(L) = 0:  AvLuDisPhos(L) = 0:  AvLuTotPhos(L) = 0
   UrbanSed(L) = 0
Next L

' initialize variables for septic system model
AvSeptNitr = 0
AvSeptPhos = 0

' the good stuff, the bulk of the program

' this is where the input file, weather1.dat is and output file results.dat are opened

Open "C:\Documents and Settings\elabusr\Desktop\weather1.dat" For Input As #1
Open "C:\Documents and Settings\elabusr\Desktop\results.dat" For Output As #2

For Y = 1 To NYrs
   For I = 1 To 12

      Input #1, DayMonth
      DaysMonth(Y, I) = DayMonth
      Temp(Y, I, 0) = DayMonth
      Prec(Y, I, 0) = DayMonth

      For J = 1 To DaysMonth(Y, I)

         Input #1, Temp(Y, I, J)
         Input #1, Prec(Y, I, J)

      Next J

   Next I
Next Y

Close #1

For Y = 1 To NYrs

      For I = 1 To 12
```

```
' initialize septic system variables
MonthPondNitr(I) = 0
MonthPondPhos(I) = 0
MonthNormNitr(I) = 0
MonthShortNitr(I) = 0
MonthShortPhos(I) = 0
MonthDischargeNitr(I) = 0
MonthDischargePhos(I) = 0

For L = 1 To NLU
  QRunoff(L, I) = 0:  ErosWashoff(L, I) = 0
Next L

For J = 1 To DaysMonth(Y, I)

  ' urban sediment accumulation updated

  For L = NRur + 1 To NLU
    UrbanSed(L) = UrbanSed(L) * Exp(-0.12) + (1 - Exp(-0.12)) / 0.12

  Next L

  DailyTemp = Temp(Y, I, J)
  DailyPrec = Prec(Y, I, J)

  melt = 0
  rain = 0
  Water = 0
  Erosiv = 0
  ET = 0
  QTotal = 0
  AreaTotal = 0

  If DailyTemp <= 0 Then
    ' . . . precipitation is snow
    snow = snow + DailyPrec

  Else
    rain = DailyPrec

    If snow > 0.001 Then

      ' . . . snow melt degree per day
      melt = 0.45 * DailyTemp

      If melt > snow Then melt = snow
      ' . . . eliminate remaining snow
      snow = snow - melt

    End If

    ' water available from melt

    Water = rain + melt

    If rain > 0 And snow < 0.001 Then
```

```
    Erosiv = 6.46 * Acoef(I) * rain ^ 1.81

End If

' if water is available then call RunoffErosionSediment to compute r, e, and s

If Water > 0.01 Then

    For L = 1 To NLU
        Qrun = 0
        If CN(2, L) > 0 Then
            If melt <= 0 Then
                If Grow(I) = 1 Then
                    '  . . . this is for the growing season
                    Select Case AMC5

                    Case Is >= 5.33
                        CNum = CN(3, L)
                    Case Is < 3.56
                        CNum = CN(1, L) + (CN(2, L) - CN(1, L)) * AMC5 / 3.56
                    Case Else
                        CNum = CN(2, L) + (CN(3, L) - CN(2, L)) * (AMC5 - 3.56) / 1.77
                    End Select
                Else
                    ' . . . this is when not growing season
                    Select Case AMC5

                    Case Is >= 2.79
                        CNum = CN(3, L)
                    Case Is < 1.27
                        CNum = CN(1, L) + (CN(2, L) - CN(1, L)) * AMC5 / 1.27
                    Case Else
                        CNum = CN(2, L) + (CN(2, L) - CN(2, L)) * (AMC5 - 1.27) / 1.52
                    End Select
                End If
            Else
                CNum = CN(3, L)
            End If

            Retention = (2540 / CNum) - 25.4

            If Area(L) > 0 Then
                If Retention < 0 Then Retention = 0

                If Water >= 0.2 * Retention Then
                    Qrun = (Water - 0.2 * Retention) ^ 2 / (Water + 0.8 * Retention)
                    QTotal = QTotal + Qrun * Area(L)
                    QRunoff(L, I) = QRunoff(L, I) + Qrun
                End If
            End If

            ' . . . now erosion and sediment washoff

            If L <= NLU Then
                RurEros = 1.32 * Erosiv * KLSCP(L) * Area(L)
```

```
                        Erosion(Y, I) = Erosion(Y, I) + RurEros: ErosWashoff(L, I) =
ErosWashoff(L, I) + RurEros
                    End If

                    If L > NRur Then
                        Washoff = (1 - Exp(-1.81 * Qrun)) * UrbanSed(L)
                        UrbanSed(L) = UrbanSed(L) - Washoff
                        UrbErosWashoff(L, I) = UrbErosWashoff(L, I) + Washoff

                    End If

                End If

                AreaTotal = AreaTotal + Area(L)
            Next L

            ' total watershed

            QTotal = QTotal / AreaTotal
            SedTrans(Y, I) = SedTrans(Y, I) + QTotal ^ 1.67
            Runoff(Y, I) = Runoff(Y, I) + QTotal

    ' end of RunoffErosionSediment calculations

            End If

        End If

        ' . . . update antecedent moisture contents

        AMC5 = AMC5 - AntMoist(5) + Water

        For K = 1 To 4
            AntMoist(6 - K) = AntMoist(5 - K)
        Next K

        AntMoist(1) = Water

        ' compute ET from saturated vapor pressure

        If DailyTemp > 0 Then
            SatVaPressure = 33.8639 * ((0.00738 * DailyTemp + 0.8072) ^ 8 - 0.000019 *
Abs(1.8 * DailyTemp + 48) + 0.001316)
            PotenET = 0.021 * DaysHrs(I) ^ 2 * SatVaPressure / (DailyTemp + 273)
            ET = CV(I) * PotenET
        End If

    ' end of DailyWeather calculations

        Infiltration = Water - QTotal
        grflow = RecessCoef * SatStor
        deepseep = SeepCoef * SatStor

        ' calculate evapotranspiration, percolation, and the next day's unsaturated storage as
limited by the unsaturated zone maximum water capacity
```

```
            UnsatStor = UnsatStor + Infiltration

            If ET >= UnsatStor Then
               ET = UnsatStor
               UnsatStor = 0
            Else
               UnsatStor = UnsatStor - ET
            End If

            If UnsatStor > MaxWaterCap Then
               Percolation = UnsatStor - MaxWaterCap
               UnsatStor = UnsatStor - Percolation
            Else
               Percolation = 0
            End If

            ' calculate storage in saturated zones and groundwater discharge

            SatStor = SatStor + Percolation - grflow - deepseep

            If SatStor < 0 Then SatStor = 0

            Flow = QTotal + grflow

            ' calculate total hydrology values

            Precipitation(Y, I) = Precipitation(Y, I) + Prec(Y, I, J) 'Precipitation(Y, I) =
Precipitation(Y, I) + Prec(Y, I, J)
            EvapoTrans(Y, I) = EvapoTrans(Y, I) + ET
            StreamFlow(Y, I) = StreamFlow(Y, I) + Flow
            Groundwater(Y, I) = Groundwater(Y, I) + grflow

            'calculate daily loads from septic systems

            'MonthPondNitr(I) = MonthPondNitr(I) + PondNitr note: omitted for springfield
            'MonthPondPhos(I) = MonthPondPhos(I) + PondPhos mote: omitted for springfield
            MonthNormNitr(I) = MonthNormNitr(I) + NitrSepticLoad - NitrPlantUptake * Grow(I)
            MonthShortNitr(I) = MonthShortNitr(I) + NitrSepticLoad - NitrPlantUptake * Grow(I)
            MonthShortPhos(I) = MonthShortPhos(I) + PhosSepticLoad - PhosPlantUptake *
Grow(I)
            MonthDischargeNitr(I) = MonthDischargeNitr(I) + NitrSepticLoad
            MonthDischargePhos(I) = MonthDischargePhos(I) + PhosSepticLoad

         Next J
      Next I

      For I = 1 To 12
         For L = 1 To NLU
            LuRunoff(Y, L) = LuRunoff(Y, L) + QRunoff(L, I)
         Next L

         Precipitation(Y, 0) = Precipitation(Y, 0) + Precipitation(Y, I)
         StreamFlow(Y, 0) = StreamFlow(Y, 0) + StreamFlow(Y, I)
         Runoff(Y, 0) = Runoff(Y, 0) + Runoff(Y, I)
         Groundwater(Y, 0) = Groundwater(Y, 0) + Groundwater(Y, I)
         EvapoTrans(Y, 0) = EvapoTrans(Y, 0) + EvapoTrans(Y, I)
```

```
Next I

' calculate annual nitrogen load from normal septic systems
AnNormNitr = 0

For I = 1 To 12
   AnNormNitr = AnNormNitr + MonthNormNitr(I) * NumNormSys
Next I

' calc sediment yields
For I = 1 To 12
   BSed(I) = 0

   For M = I To 12
      BSed(I) = BSed(I) + SedTrans(Y, M)
   Next M

   For M = 1 To I
      If BSed(M) > 0 Then SedYield(Y, I) = SedYield(Y, I) + Erosion(Y, M) / BSed(M)
   Next M

   SedYield(Y, I) = SedelRatio * SedTrans(Y, I) * SedYield(Y, I)
   SedYield(Y, 0) = SedYield(Y, 0) + SedYield(Y, I)
   Erosion(Y, 0) = Erosion(Y, 0) + Erosion(Y, I)

   For L = 1 To NLU
      LuErosion(Y, L) = LuErosion(Y, L) + ErosWashoff(L, I)
   Next L
Next I

' calculate nutrient fluxes
For I = 1 To 12
   For L = 1 To NLU
      ' calculate rural dissolved nutrients
      Nconc = NitrConc(L): Pconc = PhosConc(L)

      ' manure spreading days

      If L <= ManuredAreas And I >= FirstManureMonth And I <= LastManureMonth Then
         Nconc = ManNitr: Pconc = ManPhos
      End If

      nRunoff = 0.1 * Nconc * QRunoff(L, I) * Area(L): pRunoff = 0.1 * Pconc * QRunoff(L, I) *
Area(L)
      DisNitr(Y, I) = DisNitr(Y, I) + nRunoff: DisPhos(Y, I) = DisPhos(Y, I) + pRunoff
      LuTotNitr(Y, L) = LuTotNitr(Y, L) + nRunoff: LuTotPhos(Y, L) = LuTotPhos(Y, L) +
pRunoff
      LuDisNitr(Y, L) = LuDisNitr(Y, L) + nRunoff: ludisphos(Y, L) = ludisphos(Y, L) + pRunoff

      ' now add solid nutrients

      LuTotNitr(Y, L) = LuTotNitr(Y, L) + 0.001 * SedelRatio * ErosWashoff(L, I) * SedNitr
      LuTotPhos(Y, L) = LuTotPhos(Y, L) + 0.001 * SedelRatio * ErosWashoff(L, I) *
SedPhos
   Next L
```

```
        TotNitr(Y, I) = DisNitr(Y, I) + 0.001 * SedNitr * SedYield(Y, I)
        TotPhos(Y, I) = DisPhos(Y, I) + 0.001 * SedPhos * SedYield(Y, I)

        ' now add urban nutrients

        For L = NRur + 1 To NLU
          nRunoff = UrbanNitr(L) * UrbErosWashoff(L, I) * Area(L): pRunoff = UrbanPhos(L) *
UrbErosWashoff(L, I) * Area(L)
          LuTotNitr(Y, L) = LuTotNitr(Y, L) + nRunoff: LuTotPhos(Y, L) = LuTotPhos(Y, L) +
pRunoff
          TotNitr(Y, I) = TotNitr(Y, I) + nRunoff: TotPhos(Y, I) = TotPhos(Y, I) + pRunoff
        Next L

        ' now add groundwater nutrients

        GroundNitr(Y, I) = 0.1 * GrNitrConc * Groundwater(Y, I) * AreaTotal
        GroundPhos(Y, I) = 0.1 * GrPhosConc * Groundwater(Y, I) * AreaTotal

        DisNitr(Y, I) = DisNitr(Y, I) + GroundNitr(Y, I) + PointNitr(I)
        DisPhos(Y, I) = DisPhos(Y, I) + GroundPhos(Y, I) + PointPhos(I)
        TotNitr(Y, I) = TotNitr(Y, I) + GroundNitr(Y, I) + PointNitr(I)
        TotPhos(Y, I) = TotPhos(Y, I) + GroundPhos(Y, I) + PointPhos(I)

        ' now add septic system sourcesx to monthly dissolved nutrient totals

        MonthNormNitr(I) = AnNormNitr * Groundwater(Y, I) / Groundwater(Y, 0)
        DisSeptNitr = MonthNormNitr(I) + MonthPondNitr(I) + MonthShortNitr(I) * NumShortSys +
MonthDischargeNitr(I) * NumDischargeSys
        DisSeptPhos = MonthPondPhos(I) + MonthShortPhos(I) * NumShortSys +
MonthDischargePhos(I) * NumDischargeSys

        DisSeptNitr = DisSeptNitr / 1000
        DisSeptPhos = DisSeptPhos / 1000

        DisNitr(Y, I) = DisNitr(Y, I) + DisSeptNitr
        DisPhos(Y, I) = DisPhos(Y, I) + DisSeptPhos
        TotNitr(Y, I) = TotNitr(Y, I) + DisSeptNitr
        TotPhos(Y, I) = TotPhos(Y, I) + DisSeptPhos

        ' now annual totals

        DisNitr(Y, 0) = DisNitr(Y, 0) + DisNitr(Y, I): DisPhos(Y, 0) = DisPhos(Y, 0) + DisPhos(Y, I)
        TotNitr(Y, 0) = TotNitr(Y, 0) + TotNitr(Y, I): TotPhos(Y, 0) = TotPhos(Y, 0) + TotPhos(Y, I)
        GroundNitr(Y, 0) = GroundNitr(Y, 0) + GroundNitr(Y, I)
        GroundPhos(Y, 0) = GroundPhos(Y, 0) + GroundPhos(Y, I)

        ' now update annual septic loads

        SepticNitr(Y) = SepticNitr(Y) + DisSeptNitr
        SepticPhos(Y) = SepticPhos(Y) + DisSeptPhos

      Next I

        ' finally compute the annual mean values

      For I = 1 To 12
```

```
            AvPrecipitation(I) = AvPrecipitation(I) + Precipitation(Y, I) / NYrs

            AvEvapoTrans(I) = AvEvapoTrans(I) + EvapoTrans(Y, I) / NYrs
            AvGroundWater(I) = AvGroundWater(I) + Groundwater(Y, I) / NYrs

            AvRunoff(I) = AvRunoff(I) + Runoff(Y, I) / NYrs
            AvStreamflow(I) = AvStreamflow(I) + StreamFlow(Y, I) / NYrs
            AvErosion(I) = AvErosion(I) + Erosion(Y, I) / NYrs
            AvSedyield(I) = AvSedyield(I) + SedYield(Y, I) / NYrs
            AvDisNitr(I) = AvDisNitr(I) + DisNitr(Y, I) / NYrs
            AvTotNitr(I) = AvTotNitr(I) + TotNitr(Y, I) / NYrs
            AvDisPhos(I) = AvDisPhos(I) + DisPhos(Y, I) / NYrs
            AvTotPhos(I) = AvTotPhos(I) + TotPhos(Y, I) / NYrs
            AvGroundPhos(I) = AvGroundPhos(I) + GroundPhos(Y, I) / NYrs
            AvGroundNitr(I) = AvGroundNitr(I) + GroundNitr(Y, I) / NYrs

        Next I

        For I = 1 To 12
            AvTotSedYield = AvTotSedYield + AvSedyield(I)
            AvTotErosion = AvTotErosion + AvErosion(I)
        Next I

        AvSedFract = AvTotSedYield / AvTotErosion

        For L = 1 To NLU
            AvLuRunoff(L) = AvLuRunoff(L) + LuRunoff(Y, L) / NYrs
            AvLuErosion(L) = AvLuErosion(L) + LuErosion(Y, L) / NYrs
            AvLuDisNitr(L) = AvLuDisNitr(L) + LuDisNitr(Y, L) / NYrs
            AvLuDisPhos(L) = AvLuDisPhos(L) + ludisphos(Y, L) / NYrs
            AvLuTotNitr(L) = AvLuTotNitr(L) + LuTotNitr(Y, L) / NYrs
            AvLuTotPhos(L) = AvLuTotPhos(L) + LuTotPhos(Y, L) / NYrs
        Next L

        For L = 1 To NLU
            AvLuSedYield(L) = AvLuErosion(L) * AvSedFract
        Next L


        AvSeptNitr = AvSeptNitr + SepticNitr(Y) / NYrs
        AvSeptPhos = AvSeptPhos + SepticPhos(Y) / NYrs

Next Y

'OUTPUT TO FILE RESULTS.DAT FOR PLOTTING IN MATLAB

For I = 1 To 12
    Write #2, AvPrecipitation(I)
Next I

Write #2,

For I = 1 To 12
    Write #2, AvGroundWater(I)
Next I
```

```
Write #2,

For I = 1 To 12
   Write #2, AvRunoff(I)
Next I

Write #2,

For I = 1 To 12
   Write #2, AvStreamflow(I)
Next I

Write #2,

For I = 1 To 12
   Write #2, AvErosion(I)
Next I

Write #2,

For I = 1 To 12
   Write #2, AvSedyield(I)
Next I

Write #2,

For I = 1 To 12
   Write #2, AvDisNitr(I)
Next I

Write #2,

For I = 1 To 12
   Write #2, AvTotNitr(I)
Next I

Write #2,

For I = 1 To 12
   Write #2, AvDisPhos(I)
Next I

Write #2,

For I = 1 To 12
   Write #2, AvTotPhos(I)
Next I

'OUTPUT TO FORMS

frmOutput.TextBox1.Value = Area(1)
frmOutput.TextBox2.Value = Area(2)
frmOutput.TextBox3.Value = Area(3)
frmOutput.TextBox4.Value = Area(4)
frmOutput.TextBox5.Value = Area(5)
```

```
frmOutput.TextBox6.Value = Area(6)
frmOutput.TextBox7.Value = Area(7)
frmOutput.TextBox8.Value = Area(8)
frmOutput.TextBox51.Value = Area(9)
frmOutput.TextBox52.Value = Area(10)

frmOutput.TextBox9.Value = (Int(AvLuRunoff(1) * 100)) / 100
frmOutput.TextBox10.Value = (Int(AvLuRunoff(2) * 100)) / 100
frmOutput.TextBox11.Value = (Int(AvLuRunoff(3) * 100)) / 100
frmOutput.TextBox12.Value = (Int(AvLuRunoff(4) * 100)) / 100
frmOutput.TextBox13.Value = (Int(AvLuRunoff(5) * 100)) / 100
frmOutput.TextBox14.Value = (Int(AvLuRunoff(6) * 100)) / 100
frmOutput.TextBox15.Value = (Int(AvLuRunoff(7) * 100)) / 100
frmOutput.TextBox16.Value = (Int(AvLuRunoff(8) * 100)) / 100
frmOutput.TextBox59.Value = (Int(AvLuRunoff(9) * 100)) / 100
frmOutput.TextBox60.Value = (Int(AvLuRunoff(10) * 10)) / 10

frmOutput.TextBox17.Value = (Int(AvLuErosion(1) * 10)) / 10
frmOutput.TextBox18.Value = (Int(AvLuErosion(2) * 10)) / 10
frmOutput.TextBox19.Value = (Int(AvLuErosion(3) * 10)) / 10
frmOutput.TextBox20.Value = (Int(AvLuErosion(4) * 10)) / 10
frmOutput.TextBox21.Value = (Int(AvLuErosion(5) * 10)) / 10
frmOutput.TextBox22.Value = (Int(AvLuErosion(6) * 10)) / 10
frmOutput.TextBox23.Value = (Int(AvLuErosion(7) * 10)) / 10
frmOutput.TextBox24.Value = (Int(AvLuErosion(8) * 10)) / 10
frmOutput.TextBox50.Value = (Int(AvLuErosion(9) * 10)) / 10
frmOutput.TextBox67.Value = (Int(AvLuErosion(10) * 10)) / 10

frmOutput.TextBox25.Value = (Int(AvLuSedYield(1) * 10)) / 10
frmOutput.TextBox26.Value = (Int(AvLuSedYield(2) * 10)) / 10
frmOutput.TextBox27.Value = (Int(AvLuSedYield(3) * 10)) / 10
frmOutput.TextBox28.Value = (Int(AvLuSedYield(4) * 10)) / 10
frmOutput.TextBox29.Value = (Int(AvLuSedYield(5) * 10)) / 10
frmOutput.TextBox30.Value = (Int(AvLuSedYield(6) * 10)) / 10
frmOutput.TextBox31.Value = (Int(AvLuSedYield(7) * 10)) / 10
frmOutput.TextBox32.Value = (Int(AvLuSedYield(8) * 10)) / 10
frmOutput.TextBox73.Value = (Int(AvLuSedYield(9) * 10)) / 10
frmOutput.TextBox74.Value = (Int(AvLuSedYield(10) * 10)) / 10


frmOutput.TextBox33.Value = (Int(AvLuDisNitr(1) * 100)) / 100
frmOutput.TextBox34.Value = (Int(AvLuDisNitr(2) * 100)) / 100
frmOutput.TextBox35.Value = (Int(AvLuDisNitr(3) * 100)) / 100
frmOutput.TextBox36.Value = (Int(AvLuDisNitr(4) * 100)) / 100
frmOutput.TextBox37.Value = (Int(AvLuDisNitr(5) * 100)) / 100
frmOutput.TextBox38.Value = (Int(AvLuDisNitr(6) * 100)) / 100
frmOutput.TextBox39.Value = (Int(AvLuDisNitr(7) * 100)) / 100
frmOutput.TextBox40.Value = (Int(AvLuDisNitr(8) * 100)) / 100
frmOutput.TextBox81.Value = (Int(AvLuDisNitr(9) * 100)) / 100
frmOutput.TextBox82.Value = (Int(AvLuDisNitr(10) * 100)) / 100

frmOutput.TextBox41.Value = (Int(AvLuDisPhos(1) * 100)) / 100
frmOutput.TextBox42.Value = (Int(AvLuDisPhos(2) * 100)) / 100
frmOutput.TextBox43.Value = (Int(AvLuDisPhos(3) * 100)) / 100
frmOutput.TextBox44.Value = (Int(AvLuDisPhos(4) * 100)) / 100
frmOutput.TextBox45.Value = (Int(AvLuDisPhos(5) * 100)) / 100
```

```
frmOutput.TextBox46.Value = (Int(AvLuDisPhos(6) * 100)) / 100
frmOutput.TextBox47.Value = (Int(AvLuDisPhos(7) * 100)) / 100
frmOutput.TextBox48.Value = (Int(AvLuDisPhos(8) * 100)) / 100
frmOutput.TextBox89.Value = (Int(AvLuDisPhos(9) * 100)) / 100
frmOutput.TextBox90.Value = (Int(AvLuDisPhos(10) * 100)) / 100

frmOutput.TextBox91.Value = (Int(AvLuTotNitr(1) * 100)) / 100
frmOutput.TextBox92.Value = (Int(AvLuTotNitr(2) * 100)) / 100
frmOutput.TextBox93.Value = (Int(AvLuTotNitr(3) * 100)) / 100
frmOutput.TextBox94.Value = (Int(AvLuTotNitr(4) * 100)) / 100
frmOutput.TextBox95.Value = (Int(AvLuTotNitr(5) * 100)) / 100
frmOutput.TextBox96.Value = (Int(AvLuTotNitr(6) * 100)) / 100
frmOutput.TextBox97.Value = (Int(AvLuTotNitr(7) * 100)) / 100
frmOutput.TextBox98.Value = (Int(AvLuTotNitr(8) * 100)) / 100
frmOutput.TextBox107.Value = (Int(AvLuTotNitr(9) * 10)) / 10
frmOutput.TextBox108.Value = (Int(AvLuTotNitr(10) * 10)) / 10

frmOutput.TextBox99.Value = (Int(AvLuTotPhos(1) * 100)) / 100
frmOutput.TextBox100.Value = (Int(AvLuTotPhos(2) * 100)) / 100
frmOutput.TextBox101.Value = (Int(AvLuTotPhos(3) * 100)) / 100
frmOutput.TextBox102.Value = (Int(AvLuTotPhos(4) * 100)) / 100
frmOutput.TextBox103.Value = (Int(AvLuTotPhos(5) * 100)) / 100
frmOutput.TextBox104.Value = (Int(AvLuTotPhos(6) * 100)) / 100
frmOutput.TextBox105.Value = (Int(AvLuTotPhos(7) * 100)) / 100
frmOutput.TextBox106.Value = (Int(AvLuTotPhos(8) * 100)) / 100
frmOutput.TextBox109.Value = (Int(AvLuTotPhos(9) * 100)) / 100
frmOutput.TextBox110.Value = (Int(AvLuTotPhos(10) * 100)) / 100

frmOutput.TextBox113.Value = AvPrecipitation(1)
frmOutput.TextBox114.Value = AvPrecipitation(2)
frmOutput.TextBox115.Value = AvPrecipitation(3)
frmOutput.TextBox116.Value = AvPrecipitation(4)
frmOutput.TextBox117.Value = AvPrecipitation(5)
frmOutput.TextBox118.Value = AvPrecipitation(6)
frmOutput.TextBox119.Value = AvPrecipitation(7)
frmOutput.TextBox120.Value = AvPrecipitation(8)
frmOutput.TextBox144.Value = AvPrecipitation(9)
frmOutput.TextBox145.Value = AvPrecipitation(10)
frmOutput.TextBox146.Value = AvPrecipitation(11)
frmOutput.TextBox147.Value = AvPrecipitation(12)

frmOutput.TextBox121.Value = (Int(AvEvapoTrans(1) * 10)) / 10
frmOutput.TextBox122.Value = (Int(AvEvapoTrans(2) * 10)) / 10
frmOutput.TextBox123.Value = (Int(AvEvapoTrans(3) * 10)) / 10
frmOutput.TextBox124.Value = (Int(AvEvapoTrans(4) * 10)) / 10
frmOutput.TextBox125.Value = (Int(AvEvapoTrans(5) * 10)) / 10
frmOutput.TextBox126.Value = (Int(AvEvapoTrans(6) * 10)) / 10
frmOutput.TextBox127.Value = (Int(AvEvapoTrans(7) * 10)) / 10
frmOutput.TextBox128.Value = (Int(AvEvapoTrans(8) * 10)) / 10
frmOutput.TextBox148.Value = (Int(AvEvapoTrans(9) * 10)) / 10
frmOutput.TextBox149.Value = (Int(AvEvapoTrans(10) * 10)) / 10
frmOutput.TextBox150.Value = (Int(AvEvapoTrans(11) * 10)) / 10
frmOutput.TextBox151.Value = (Int(AvEvapoTrans(12) * 10)) / 10

frmOutput.TextBox112.Value = (Int(AvGroundWater(1) * 100)) / 100
frmOutput.TextBox129.Value = (Int(AvGroundWater(2) * 100)) / 100
```

```
frmOutput.TextBox130.Value = (Int(AvGroundWater(3) * 100)) / 100
frmOutput.TextBox131.Value = (Int(AvGroundWater(4) * 100)) / 100
frmOutput.TextBox132.Value = (Int(AvGroundWater(5) * 100)) / 100
frmOutput.TextBox111.Value = (Int(AvGroundWater(6) * 100)) / 100
frmOutput.TextBox133.Value = (Int(AvGroundWater(7) * 100)) / 100
frmOutput.TextBox134.Value = (Int(AvGroundWater(8) * 100)) / 100
frmOutput.TextBox152.Value = (Int(AvGroundWater(9) * 100)) / 100
frmOutput.TextBox143.Value = (Int(AvGroundWater(10) * 100)) / 100
frmOutput.TextBox153.Value = (Int(AvGroundWater(11) * 100)) / 100
frmOutput.TextBox154.Value = (Int(AvGroundWater(12) * 100)) / 100

frmOutput.TextBox135.Value = (Int(AvRunoff(1) * 100)) / 100
frmOutput.TextBox136.Value = (Int(AvRunoff(2) * 100)) / 100
frmOutput.TextBox137.Value = (Int(AvRunoff(3) * 100)) / 100
frmOutput.TextBox138.Value = (Int(AvRunoff(4) * 100)) / 100
frmOutput.TextBox139.Value = (Int(AvRunoff(5) * 100)) / 100
frmOutput.TextBox140.Value = (Int(AvRunoff(6) * 100)) / 100
frmOutput.TextBox141.Value = (Int(AvRunoff(7) * 100)) / 100
frmOutput.TextBox142.Value = (Int(AvRunoff(8) * 100)) / 100
frmOutput.TextBox155.Value = (Int(AvRunoff(9) * 100)) / 100
frmOutput.TextBox156.Value = (Int(AvRunoff(10) * 100)) / 100
frmOutput.TextBox157.Value = (Int(AvRunoff(11) * 100)) / 100
frmOutput.TextBox158.Value = (Int(AvRunoff(12) * 100)) / 100

frmOutput.TextBox159.Value = (Int(AvStreamflow(1) * 100)) / 100
frmOutput.TextBox160.Value = (Int(AvStreamflow(2) * 100)) / 100
frmOutput.TextBox161.Value = (Int(AvStreamflow(3) * 100)) / 100
frmOutput.TextBox162.Value = (Int(AvStreamflow(4) * 100)) / 100
frmOutput.TextBox163.Value = (Int(AvStreamflow(5) * 100)) / 100
frmOutput.TextBox164.Value = (Int(AvStreamflow(6) * 100)) / 100
frmOutput.TextBox165.Value = (Int(AvStreamflow(7) * 100)) / 100
frmOutput.TextBox166.Value = (Int(AvStreamflow(8) * 100)) / 100
frmOutput.TextBox167.Value = (Int(AvStreamflow(9) * 100)) / 100
frmOutput.TextBox168.Value = (Int(AvStreamflow(10) * 100)) / 100
frmOutput.TextBox169.Value = (Int(AvStreamflow(11) * 100)) / 100
frmOutput.TextBox170.Value = (Int(AvStreamflow(12) * 100)) / 100

frmOutput.TextBox171.Value = (Int(AvErosion(1) * 10)) / 10
frmOutput.TextBox172.Value = (Int(AvErosion(2) * 10)) / 10
frmOutput.TextBox173.Value = (Int(AvErosion(3) * 10)) / 10
frmOutput.TextBox174.Value = (Int(AvErosion(4) * 10)) / 10
frmOutput.TextBox175.Value = (Int(AvErosion(5) * 10)) / 10
frmOutput.TextBox176.Value = (Int(AvErosion(6) * 10)) / 10
frmOutput.TextBox177.Value = (Int(AvErosion(7) * 10)) / 10
frmOutput.TextBox178.Value = (Int(AvErosion(8) * 10)) / 10
frmOutput.TextBox179.Value = (Int(AvErosion(9) * 10)) / 10
frmOutput.TextBox180.Value = (Int(AvErosion(10) * 10)) / 10
frmOutput.TextBox181.Value = (Int(AvErosion(11) * 10)) / 10
frmOutput.TextBox182.Value = (Int(AvErosion(12) * 10)) / 10

frmOutput.TextBox183.Value = (Int(AvSedyield(1) * 100)) / 100
frmOutput.TextBox184.Value = (Int(AvSedyield(2) * 100)) / 100
frmOutput.TextBox185.Value = (Int(AvSedyield(3) * 100)) / 100
frmOutput.TextBox186.Value = (Int(AvSedyield(4) * 100)) / 100
frmOutput.TextBox187.Value = (Int(AvSedyield(5) * 100)) / 100
frmOutput.TextBox188.Value = (Int(AvSedyield(6) * 100)) / 100
```

```
frmOutput.TextBox189.Value = (Int(AvSedyield(7) * 100)) / 100
frmOutput.TextBox190.Value = (Int(AvSedyield(8) * 100)) / 100
frmOutput.TextBox191.Value = (Int(AvSedyield(9) * 100)) / 100
frmOutput.TextBox192.Value = (Int(AvSedyield(10) * 100)) / 100
frmOutput.TextBox193.Value = (Int(AvSedyield(11) * 100)) / 100
frmOutput.TextBox194.Value = (Int(AvSedyield(12) * 100)) / 100

frmOutput.TextBox280.Value = (Int(AvDisNitr(1) * 10)) / 10
frmOutput.TextBox281.Value = (Int(AvDisNitr(2) * 10)) / 10
frmOutput.TextBox282.Value = (Int(AvDisNitr(3) * 10)) / 10
frmOutput.TextBox283.Value = (Int(AvDisNitr(4) * 10)) / 10
frmOutput.TextBox284.Value = (Int(AvDisNitr(5) * 10)) / 10
frmOutput.TextBox285.Value = (Int(AvDisNitr(6) * 10)) / 10
frmOutput.TextBox286.Value = (Int(AvDisNitr(7) * 10)) / 10
frmOutput.TextBox287.Value = (Int(AvDisNitr(8) * 10)) / 10
frmOutput.TextBox288.Value = (Int(AvDisNitr(9) * 10)) / 10
frmOutput.TextBox289.Value = (Int(AvDisNitr(10) * 10)) / 10
frmOutput.TextBox290.Value = (Int(AvDisNitr(11) * 10)) / 10
frmOutput.TextBox291.Value = (Int(AvDisNitr(12) * 10)) / 10

frmOutput.TextBox292.Value = (Int(AvDisPhos(1) * 100)) / 100
frmOutput.TextBox293.Value = (Int(AvDisPhos(2) * 100)) / 100
frmOutput.TextBox294.Value = (Int(AvDisPhos(3) * 100)) / 100
frmOutput.TextBox295.Value = (Int(AvDisPhos(4) * 100)) / 100
frmOutput.TextBox296.Value = (Int(AvDisPhos(5) * 100)) / 100
frmOutput.TextBox297.Value = (Int(AvDisPhos(6) * 100)) / 100
frmOutput.TextBox298.Value = (Int(AvDisPhos(7) * 100)) / 100
frmOutput.TextBox299.Value = (Int(AvDisPhos(8) * 100)) / 100
frmOutput.TextBox300.Value = (Int(AvDisPhos(9) * 100)) / 100
frmOutput.TextBox301.Value = (Int(AvDisPhos(10) * 100)) / 100
frmOutput.TextBox302.Value = (Int(AvDisPhos(11) * 100)) / 100
frmOutput.TextBox303.Value = (Int(AvDisPhos(12) * 100)) / 100

frmOutput.TextBox304.Value = (Int(AvTotNitr(1) * 10)) / 10
frmOutput.TextBox305.Value = (Int(AvTotNitr(2) * 10)) / 10
frmOutput.TextBox306.Value = (Int(AvTotNitr(3) * 10)) / 10
frmOutput.TextBox307.Value = (Int(AvTotNitr(4) * 10)) / 10
frmOutput.TextBox308.Value = (Int(AvTotNitr(5) * 10)) / 10
frmOutput.TextBox309.Value = (Int(AvTotNitr(6) * 10)) / 10
frmOutput.TextBox310.Value = (Int(AvTotNitr(7) * 10)) / 10
frmOutput.TextBox311.Value = (Int(AvTotNitr(8) * 10)) / 10
frmOutput.TextBox312.Value = (Int(AvTotNitr(9) * 10)) / 10
frmOutput.TextBox313.Value = (Int(AvTotNitr(10) * 10)) / 10
frmOutput.TextBox314.Value = (Int(AvTotNitr(11) * 10)) / 10
frmOutput.TextBox255.Value = (Int(AvTotNitr(12) * 10)) / 10

frmOutput.TextBox316.Value = (Int(AvTotPhos(1) * 100)) / 100
frmOutput.TextBox317.Value = (Int(AvTotPhos(2) * 100)) / 100
frmOutput.TextBox318.Value = (Int(AvTotPhos(3) * 100)) / 100
frmOutput.TextBox319.Value = (Int(AvTotPhos(4) * 100)) / 100
frmOutput.TextBox320.Value = (Int(AvTotPhos(5) * 100)) / 100
frmOutput.TextBox321.Value = (Int(AvTotPhos(6) * 100)) / 100
frmOutput.TextBox322.Value = (Int(AvTotPhos(7) * 100)) / 100
frmOutput.TextBox323.Value = (Int(AvTotPhos(8) * 100)) / 100
frmOutput.TextBox324.Value = (Int(AvTotPhos(9) * 100)) / 100
frmOutput.TextBox325.Value = (Int(AvTotPhos(10) * 100)) / 100
```

```
frmOutput.TextBox326.Value = (Int(AvTotPhos(11) * 100)) / 100
frmOutput.TextBox315.Value = (Int(AvTotPhos(12) * 100)) / 100

'groundwater output

For I = 1 To 12
   frmOutput.TextBox331.Value = frmOutput.TextBox331.Value + (Int(AvGroundNitr(I) * 10)) / 10
Next I

For I = 1 To 12
   frmOutput.TextBox334.Value = frmOutput.TextBox334.Value + (Int(AvGroundNitr(I) * 10)) / 10
Next I

For I = 1 To 12
   frmOutput.TextBox332.Value = frmOutput.TextBox332.Value + (Int(AvGroundPhos(I) * 100)) /
100
Next I

For I = 1 To 12
   frmOutput.TextBox333.Value = frmOutput.TextBox333.Value + (Int(AvGroundPhos(I) * 100)) /
100
Next I

'septic output

frmOutput.TextBox357.Value = Int(AvSeptNitr * 10) / 10: frmOutput.TextBox359.Value =
Int(AvSeptNitr * 10) / 10
frmOutput.TextBox358.Value = Int(AvSeptPhos * 10) / 10: frmOutput.TextBox356.Value =
Int(AvSeptPhos * 10) / 10

'FILL IN TOTALS FOR OUTPUT FORM

For g = 1 To NLU
   frmOutput.TextBox335.Value = frmOutput.TextBox335.Value + Area(g)
Next g

For g = 1 To NLU
   frmOutput.TextBox337.Value = frmOutput.TextBox337.Value + (Int(AvLuRunoff(g) * Area(g)) *
10) / 10
Next g
frmOutput.TextBox337.Value = frmOutput.TextBox337 / AreaTotal

For g = 1 To NLU
   frmOutput.TextBox336.Value = frmOutput.TextBox336.Value + (Int(AvLuErosion(g)))
Next g

For g = 1 To NLU
   frmOutput.TextBox338.Value = frmOutput.TextBox338.Value + (Int(AvLuSedYield(g) * 10)) / 10
Next g

'total groundwaternitr, LuDisNitr, and septnitr

For g = 1 To NLU
   frmOutput.TextBox339.Value = frmOutput.TextBox339.Value + (Int(AvLuDisNitr(g) * 10)) / 10
Next g
```

```
For I = 1 To 12
   frmOutput.TextBox339.Value = frmOutput.TextBox339.Value + (Int(AvGroundNitr(I) * 10)) / 10
Next I

frmOutput.TextBox339.Value = frmOutput.TextBox339.Value + (Int(AvSeptNitr * 10) / 10)

'total groundwaterphos, ludisphos, and septphos

For g = 1 To NLU
   frmOutput.TextBox340.Value = frmOutput.TextBox340.Value + (Int(AvLuDisPhos(g) * 100)) /
100
Next g

For I = 1 To 12
   frmOutput.TextBox340.Value = frmOutput.TextBox340.Value + (Int(AvGroundPhos(I) * 100)) /
100
Next I

frmOutput.TextBox340.Value = frmOutput.TextBox340.Value + (Int(AvSeptPhos * 10) / 10)

'total groundwaternitr, lutotnitr, and septnitr

For g = 1 To NLU
   frmOutput.TextBox341.Value = frmOutput.TextBox341.Value + (Int(AvLuTotNitr(g) * 10)) / 10
Next g

For I = 1 To 12
   frmOutput.TextBox341.Value = frmOutput.TextBox341.Value + (Int(AvGroundNitr(I) * 10)) / 10
Next I

frmOutput.TextBox341.Value = frmOutput.TextBox341.Value + Int(AvSeptNitr * 10) / 10

'total groundwaterphos, lutotphos, and septphos

For g = 1 To NLU
   frmOutput.TextBox342.Value = frmOutput.TextBox342.Value + (Int(AvLuTotPhos(g) * 100)) /
100
Next g

For I = 1 To 12
   frmOutput.TextBox342.Value = frmOutput.TextBox342.Value + (Int(AvGroundPhos(I) * 100)) /
100
Next I

frmOutput.TextBox342.Value = frmOutput.TextBox342.Value + Int(AvSeptPhos * 10) / 10

For g = 1 To 12
   frmOutput.TextBox349.Value = frmOutput.TextBox349.Value + AvPrecipitation(g)
Next g

For g = 1 To 12
   frmOutput.TextBox348.Value = frmOutput.TextBox348.Value + (Int(AvEvapoTrans(g) * 10)) /
10
Next g

For g = 1 To 12
```

```
   frmOutput.TextBox347.Value = frmOutput.TextBox347.Value + (Int(AvGroundWater(g) * 100)) /
100
Next g

For g = 1 To 12
   frmOutput.TextBox346.Value = frmOutput.TextBox346.Value + (Int(AvRunoff(g) * 100)) / 100
Next g

For g = 1 To 12
   frmOutput.TextBox345.Value = frmOutput.TextBox345.Value + (Int(AvStreamflow(g) * 100)) /
100
Next g

For g = 1 To 12
   frmOutput.TextBox344.Value = frmOutput.TextBox344.Value + (Int(AvErosion(g) * 10)) / 10
Next g

For g = 1 To 12
   frmOutput.TextBox343.Value = frmOutput.TextBox343.Value + (Int(AvSedyield(g) * 100)) / 100
Next g

For g = 1 To 12
   frmOutput.TextBox352.Value = frmOutput.TextBox352.Value + (Int(AvDisNitr(g) * 10)) / 10
Next g

For g = 1 To 12
   frmOutput.TextBox353.Value = frmOutput.TextBox353.Value + (Int(AvDisPhos(g) * 10)) / 10
Next g

For g = 1 To 12
   frmOutput.TextBox354.Value = frmOutput.TextBox354.Value + (Int(AvTotNitr(g) * 10)) / 10
Next g

For g = 1 To 12
   frmOutput.TextBox355.Value = frmOutput.TextBox355.Value + (Int(AvTotPhos(g) * 10)) / 10
Next g

frmOutput.Show

Close #2
Close #10
Close #11

End Sub
```

Appendix 2 – Matlab Plotting Script

GWLFNutr.m

```
AvPrecipitation = [0];
AvDisNitr = [0];
AvDisPhos = [0];
AvTotNitr = [0];
AvTotPhos = [0];
Month = [0];

fid = fopen('results.dat');
temp = fscanf(fid, '%g', [1 inf]);
temp = temp';

for i = 1 : 12
   AvPrecipitation(i) = temp(i);
end

for i = 1:12
   AvDisNitr(i) = temp(i+72);
end

for i = 1:12
   AvTotNitr(i) = temp(i+84);
end

for i = 1:12
   AvDisPhos(i) = temp(i+96);
end

for i = 1:12
   AvTotPhos(i) = temp(i+108);
end

for i = 1:12
   Month(i) = i;
end

plot(Month, AvDisNitr, Month, AvTotNitr, Month, AvDisPhos, Month, AvTotPhos), axis([1 12 0
(max(AvTotNitr)+10)]);
   title('Nutrient Loadings'), xlabel('Time (Months)'), yLabel('Nutrient Load (kg)');

legend('Average Dis Nitr', 'Average Total Nitr', 'Average Dis Phos', 'Average Total Phos', 4);

fclose(fid);
```

GWLFHydro.m

```
AvPrecipitation = [0];
AvGroundWater = [0];
AvStreamflow = [0];
AvSedYield = [0];
Month = [0];
```

```
fid = fopen('results.dat');
temp = fscanf(fid, '%g', [1 inf]);
temp = temp';

for i = 1 : 12
   AvPrecipitation(i) = temp(i);
end

for i = 1: 12
   AvGroundWater(i) = temp(i+12);
end

for i = 1:12
   AvRunoff(i) = temp(i+24);
end

for i = 1:12
   AvStreamflow(i) = temp(i+36);
end

for i = 1:12
   Month(i) = i;
end

plot(Month, AvStreamflow, Month, AvRunoff, Month, AvGroundWater), axis([1 12 0
max(AvStreamflow)+.5]);
   title('Hydro Loadings'), xlabel('Time (Months)'), yLabel('Water Distribution (cm)');

legend('Average Streamflow', 'Average Runoff', 'Average Ground Water', 3);

fclose(fid);
```

GWLFRunoff.m

```
vPrecipitation = [0];
AvRunoff = [0];
AvErosion = [0];
Month = [0];

fid = fopen('results.dat');
temp = fscanf(fid, '%g', [1 inf]);
temp = temp';

for i = 1 : 12
   AvPrecipitation(i) = temp(i);
end

for i = 1:12
   AvErosion(i) = temp(i+48);
end

for i = 1:12
   AvSedYield(i) = temp(i+60);
end
```

```
for i = 1:12
   Month(i) = i;
end

plot(Month, AvSedYield, Month, AvErosion), axis([1 12 0 (max(AvErosion)+10)]);
   title('Sediment Loadings'), xlabel('Time (Months)'), yLabel('Erosion Distribution (Mg)');
h = legend('Average Sed Yield', 'Average Erosion', 2);
fclose(fid);
```

Appendix 3 – Relevant Tables of Parameters (Haith et al.)

Soil Hydrologic Group	Description
A	Low runoff potential and high infiltration rates even when thoroughly wetted. Chiefly deep, well to excessively drained sands or gravels. High rate of water transmission (> 0.75 cm/hr).
B	Moderate infiltration rates when thoroughly wetted. Chiefly moderately deep to deep, moderately well to well drained soils with moderately fine to moderately coarse textures. Moderate rate of water transmission (0.40-0.75 cm/hr).
C	Low infiltration rates when thoroughly wetted. Chiefly soils with a layer that impedes downward movement of water, or soils with moderately fine to fine texture. Low rate of water transmission (0.15-0.40 cm/hr).
D	High runoff potential. Very low infiltration rates when thoroughly wetted. Chiefly clay soils with a high swelling potential, soils with a permanent high water table, soils with a claypan or clay layer at or near the surface, or shallow soils over nearly impervious material. Very low rate of water transmission (0-0.15 cm/hr).
Disturbed Soils (Major altering of soil profile by construction, development):	
A	Sand, loamy sand, sandy loam.
B	Silt loam, loam
C	Sandy clay loam
D	Clay loam, silty clay loam, sandy clay, silty clay, clay.

Table B-1. Descriptions of Soil Hydrologic Groups (Soil Conservation Service, 1986)

A simplified procedure can be developed, however, based on a few general observations:

1. Cover coefficients should in principle vary between 0 and 1.
2. Cover coefficients will approach their maximum value when plants have developed full foliage.
3. Because evapotranspiration measures both transpiration and evaporation of soil water, the lower limit for cover coefficients will be greater than zero. This lower limit essentially represents a situation without any plant cover.
4. The protection of soil by impervious surfaces prevents evapotranspiration.

The cover coefficients given for annual crops in Table B-6 fall to approximately 0.3 before planting and after harvest. Similarly, cover coefficients for forests reach minimum values of 0.2 to 0.3 when leaf area indices approach zero. This suggests that monthly cover coefficients for can be given the value 0.3 when foliage is absent and 1.0 otherwise. Perennial crops, such as grass, hay, meadow, and pasture, crops grown in flooded soil, such as rice, and conifers can be given a cover coefficient of 1.0 year round.

Land Use/Cover		Hydrologic Condition	Soil Hydrologic Group A	B	C	D
Fallow	Bare Soil	-	77	86	91	94
Crop residue cover (CR)		Poor[a/]	76	85	90	93
		Good	74	83	88	90
Row Crops	Straight row (SR)	Poor	72	81	88	91
		Good	67	78	85	89
	SR + CR	Poor	71	80	87	90
		Good	64	75	82	85
	Contoured (C)	Poor	70	79	84	88
		Good	65	75	82	86
	C + CR	Poor	69	78	83	87
		Good	64	74	81	85
	Contoured & terraced (C&T)	Poor	66	74	80	82
		Good	62	71	78	81
	C&T + CR	Poor	65	73	79	81
		Good	61	70	77	80
Small Grains	SR	Poor	65	76	84	88
		Good	63	75	83	87
	SR + CR	Poor	64	75	83	86
		Good	60	72	80	84
	C	Poor	63	74	82	85
		Good	61	73	81	84
	C + CR	Poor	62	73	81	84
		Good	60	72	80	83
	C&T	Poor	61	72	79	82
		Good	59	70	78	81
	C&T + CR	Poor	60	71	78	81
		Good	58	69	77	80
Close-seeded or broadcast legumes or rotation meadow	SR	Poor	66	77	85	89
		Good	58	72	81	85
	C	Poor	64	75	83	85
		Good	55	69	78	83
	C&T	Poor	63	73	80	83
		Good	51	67	76	80

a/ Hydrologic condition is based on a combination of factors that affect infiltration and runoff, including (a) density and canopy of vegetative areas, (b) amount of year-round cover, (c) amount of close-seeded legumes in rotations, (d) percent of residue cover on the land surface (good ≥ 20%), and (e) degree of surface roughness.

Table B-2. Runoff Curve Numbers (Antecedent Moisture Condition II) for Cultivated Agricultural Land (Soil Conservation Service, 1986).

Land Use/Cover	Hydrologic Condition	Soil Hydrologic Group A	B	C	D
Pasture, grassland or range - continuous forage for grazing	Poor[a/]	68	79	86	89
	Fair	49	69	79	84
	Good	39	61	74	80
Meadow - continuous grass, protected from grazing, generally mowed for hay	-	30	58	71	78
Brush - brush/weeds/grass mixture with brush the major element	Poor[b/]	48	67	77	83
	Fair	35	56	70	77
	Good	30	48	65	73
Woods/grass combination (orchard or tree farm)[c/]	Poor	57	73	82	86
	Fair	43	65	76	82
	Good	32	58	72	79
Woods	Poor[/d]	45	66	77	83
	Fair	36	60	73	79
	Good	30	55	70	77
Farmsteads - buildings, lanes, driveways and surrounding lots	-	59	74	82	86

a/ Poor: < 50% ground cover or heavily grazed with no mulch; Fair: 50 to 75% ground cover and not heavily grazed; Good: > 75% ground cover and lightly or only occasionally grazed.

b/ Poor: < 50% ground cover; Fair: 50 to 75% ground cover; Good: > 75% ground cover.

c/ Estimated as 50% woods, 50% pasture.

d/ Poor: forest litter, small trees and brush are destroyed by heavy grazing or regular burning; Fair: woods are grazed but not burned and some forest litter covers the soil; Good: Woods are protected from grazing and litter and brush adequately cover the soil.

Table B-3. Runoff Curve Numbers (Antecedent Moisture Condition II) for other Rural Land (Soil Conservation Service, 1986).

Land Use/Cover	Hydrologic Condition	Soil Hydrologic Group A	B	C	D
Herbaceous - grass, weeds & low-growing brush; brush the minor component	Poor[a/]	-	80	87	93
	Fair	-	71	81	89
	Good	-	62	74	85
Oak/aspen - oak brush, aspen, mountain mahogany, bitter brush, maple and other brush	Poor	-	66	74	79
	Fair	-	48	57	63
	Good	-	30	41	48
Pinyon/juniper - pinyon, juniper or both; grass understory	Poor	-	75	85	89
	Fair	-	58	73	80
	Good	-	41	61	71
Sagebrush with grass understory	Poor	-	67	80	85
	Fair	-	51	63	70
	Good	-	35	47	55
Desert scrub - saltbush, greasewood, creosotebrush, blackbrush, bursage, palo verde, mesquite and cactus	Poor	63	77	85	88
	Fair	55	72	81	86
	Good	49	68	79	84

a/ Poor: < 30% ground cover (litter, grass and brush overstory); Fair: 30 to 70% ground cover; Good: > 70% ground cover.

Table B-4. Runoff Curve Numbers (Antecedent Moisture Condition II) for Arid and Semiarid Rangelands (Soil Conservation Service, 1986).

Land Use	Soil Hydrologic Group A	B	C	D
Open space (lawns, parks, golf courses, cemeteries, etc.):				
Poor condition (grass cover < 50%)	68	79	86	89
Fair condition (grass cover 50-75%)	49	69	79	84
Good condition (grass cover > 75%)	39	61	74	80
Impervious areas:				
Paved parking lots, roofs, driveways, etc.)	98	98	98	98
Streets and roads:				
Paved with curbs & storm sewers	98	98	98	98
Paved with open ditches	83	89	92	93
Gravel	76	85	89	91
Dirt	72	82	87	89
Western desert urban areas:				
Natural desert landscaping (pervious areas, only)	63	77	85	88
Artificial desert landscaping (impervious weed barrier, desert shrub with 1-2 in sand or gravel mulch and basin borders)	96	96	96	96

Table B-5. Runoff Curve Numbers (Antecedent Moisture Condition II) for Urban Areas (Soil Conservation Service, 1986).

Crop	% of Growing Season 0	10	20	30	40	50	60	70	80	90	100
Field corn	0.45	0.51	0.58	0.66	0.75	0.85	0.96	1.08	1.20	1.08	0.70
Grain sorgh	0.30	0.40	0.65	0.90	1.10	1.20	1.10	0.95	0.80	0.65	0.50
Wint wheat	1.08	1.19	1.29	1.35	1.40	1.38	1.36	1.23	1.10	0.75	0.40
Cotton	0.40	0.45	0.56	0.76	1.00	1.14	1.19	1.11	0.83	0.58	0.40
Sugar beets	0.30	0.35	0.41	0.56	0.73	0.90	1.08	1.26	1.44	1.30	1.10
Cantaloupe	0.30	0.30	0.32	0.35	0.46	0.70	1.05	1.22	1.13	0.82	0.44
Potatoes	0.30	0.40	0.62	0.87	1.06	1.24	1.40	1.50	1.50	1.40	1.26
Papago peas	0.30	0.40	0.66	0.89	1.04	1.16	1.26	1.25	0.63	0.28	0.16
Beans	0.30	0.35	0.58	1.05	1.07	0.94	0.80	0.66	0.53	0.43	0.36
Rice	1.00	1.06	1.13	1.24	1.38	1.55	1.58	1.57	1.47	1.27	1.00

Table B-6. **Evapotranspiration Cover Coefficients for Annual Crops - Measured as Ratio of Evapotranspiration to Lake Evaporation (Davis & Sorensen, 1969; cited in Novotny & Chesters, 1981).**

	Alfalfa	Pasture	Grapes	Citrus Orchards	Deciduous Orchards	Sugarcane
Jan	0.83	1.16	-	0.58	-	0.65
Feb	0.90	1.23	-	0.53	-	0.50
Mar	0.96	1.19	0.15	0.65	-	0.80
Apr	1.02	1.09	0.50	0.74	0.60	1.17
May	1.08	0.95	0.80	0.73	0.80	1.21
June	1.14	0.83	0.70	0.70	0.90	1.22
July	1.20	0.79	0.45	0.81	0.90	1.23
Aug	1.25	0.80	-	0.96	0.80	1.24
Sept	1.22	0.91	-	1.08	0.50	1.26
Oct	1.18	0.91	-	1.03	0.20	1.27
Nov	1.12	0.83	-	0.82	0.20	1.28
Dec	0.86	0.69	-	0.65	-	0.80

Table B-7. **Evapotranspiration Cover Coefficients for Perennial Crops - Measured as Ratio of Evapotranspiration to Lake Evaporation (Davis & Sorensen, 1969; cited in Novotny & Chesters, 1981).**

In urban areas, ground cover is a mixture of trees and grass. It follows that cover factors for pervious areas are weighted averages of the perennial crop, hardwood, and softwood cover factors. It may be difficult to determine the relative fractions of urban areas with these covers. Since these covers would have different values only during dormant seasons, it is reasonable to assume a constant month value of 1.0 for urban pervious surfaces and zero for impervious surfaces.

These approximate cover coefficients are given in Table B-8. Table B-9 list mean monthly values of daylight hours (H_t) for use in Equation A-31.

Cover	Dormant Season	Growing Season
Annual crops (foliage only in growing season)	0.3	1.0
Perennial crops (year-round foliage: grass, pasture, meadow, etc.)	1.0	1.0
Saturated crops (rice)	1.0	1.0
Hardwood (deciduous) forests & orchards	0.3	1.0
Softwood (conifer) forests & orchards	1.0	1.0
Disturbed areas & bare soil (barn yards, fallow, logging trails, construction and mining)	0.3	0.3
Urban areas (I = impervious fraction)	1 - I	1 - I

Table B-8. Approximate Values for Evapotranspiration Cover Coefficients.