AN ABSTRACT OF THE THESIS OF

Huaisheng Chen for the degree of Doctor of Philosophy in Forest Engineering presented on December 10, 1996. Title: Object Watershed Link Simulation (OWLS).

Abstract approved: Signature redacted for privacy.

Robert L. Beschta

Object Watershed Link Simulation (OWLS) is a physically based watershed model. In the OWLS model, a watershed is defined as a three-dimensional object with linkages between cells and their attributes (e.g., area, slope, soil type, etc.). A cell is defined as the linkages of edges and their attributes (e.g., length, slope, etc.) and an edge is defined as the linkage of nodes and their attributes (e.g., depth of soil, elevation). The watershed hydrologic components such as water depth, surface flow, etc., are features associated with the cells, edges and nodes of a watershed. Simulation of hydrologic processes across a watershed involves the calculation of flows and water balances for these cells, edges, and nodes and their linkages. Therefore, the OWLS model is a three-dimensional, object-linked, vector-based model.

OWLS includes four sub-models that focus on (1) Data Processing, (2) Geomorphology, (3) Hydrology and (4) Visualization. The Data Processing Model handles conversions of raw data from watershed surveys into OWLS format. It also handles missing data interpolation and extrapolation for air temperature, precipitation, and streamflow. The Geomorphologic Model handles the automatic watershed delineation for flowpaths, streams, and boundaries, as well as stream geometry and macropore geometry. The Hydrologic Model handles water balance, flow calculation and flow routing for the canopy, surface, subsurface and macropore system associated with each cell. The Visualization Model handles 3-D watershed projection, 2-D watershed projection, hydrograph presentation, and 3-D dynamic watershed animation for simulated flows and other hydrologic components of the Hydrologic Model.

The OWLS model was tested with data from the Bear Brook Watershed of Maine (BBWM). Results from parameter calibration and validation indicate that the model generally provided good estimation of streamflows for rain-based flood events and unstable estimations for rain-on-snow events or snowmelt-based events when air temperature was high.

Object Watershed Link Simulation (OWLS)

by

Huaisheng Chen

A THESIS

summitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented December 10, 1996
Commencement June 1997

Doctor of Philosophy thesis of Huaisheng Chen presented on December 10, 1996

APPROVED:

Signature redacted for privacy.

Major Professor, representing Forest Engineering

Signature redacted for privacy.

Signature redacted for privacy.

Head of Department of Forest Engineering

Signature redacted for privacy.

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Signature redacted for privacy. 1/29/97

Huaisheng Chen, AUTHOR

Acknowledgement

I have been enjoying working together with my major professor, Dr. Robert L. Beschta. Thanks for his acceptance of me to be his student, his valuable assistance and professional instruction in the whole process of this project and his effort to provide me with financial support throughout my degree. Also I would like to thank Dr. Parker J. Wigington in the US EPA Corvallis Lab., Dr. Steve Norton and his staff in the University of Maine, for their valuable assistance in the early stages of this project and the permission to use their data for the developing of the OWLS model. I would also like to thank the professors of my committee for their timely assistance and valuable comments. They are Dr. R.L.Beschta of Forest Engineering, Dr. Parker J. Wigington of EPA, Dr. Marvin R. Pyles of Forest Engineering, Dr. Wayne C. Huber of Civil Engineering, Dr. Peter C. Klingeman of Civil Engineering, and Dr. Chaur-Fong Chen of Bioresource Engineering.

My thanks also go to my wife Huijuan for her love and support of my study and research at home, to my parents, sister, parents-in-law and grandmother for their loves, encouragement and help, and to my son Daniel and daughter Amy for happy smiles.

Huaisheng Chen
Dec. 10, 1996

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF APPENDICES

Appendix

LIST OF APPENDIX FIGURES

LIST OF APPENDIX TABLES

LIST OF APPENDIX TABLES (Continued)

Object Watershed Link Simulation (OWLS)

Chapter One: *Introduction*

1.1. *Definition*

The *Object Watershed Link Simulation* (OWLS) Program is designed to physically and visually simulate the real time or short-term hydrological processes for small forested watersheds and to provide detailed information about watershed response to environmental changes.

1.2. *Conceptual Basis*

In order to attain the objective of this study, the design of the OWLS model employed a programming method more sophisticated than commonly used programming methods (e.g., FORTRAN programming). The *Object Orientation Programming* (OOP) Method (coded using C++) was used to construct the OWLS model. In OOP, an object is defined as a container of data type (or <u>class</u> in C++) which has some specific properties or features (or <u>data members</u>) and which also has certain types of functions (also called <u>member functions</u>). The philosophy of the OOP method is understandable if you consider the relation between cells and the human body: In order to understand the functionality of the human body, a doctor needs to know how many different kinds of cells a human has, what is in them, and how they function and relate to other cells. In addition, he/she needs to know how they are organized to form the organs and finally how the organ functions in the human body.

The OOP method used for watershed simulation is based on a similar philosophy: Starting from the basic components (objects) of the watershed(i.e., points, lines and cells), the properties that these objects have (e.g., elevation, length, slope, soil, vegetation ...) and the types of functions they perform (e.g., infiltration, surface flow ...)are established. Then, the relations between these objects (linkage) to form the "organs" of the watershed (e.g., flow path, stream network, canopy, surface, soil, macropore pipes ...) are specified. Finally, it is necessary to establish how these "organs" operate together (linkage) to reflect watershed behavior (e.g., streamflow, stream chemistry).

1.3. *System of Objects*

Within the OWLS model, there are numerous objects representing many different components of a watershed. Table 1-1 includes the selected objects used in OWLS; a complete list of objects is included in Appendix I.

Table 1-1. Example of objects in the OWLS model.

OWLS Modules	**Object Name (Induced Object)**	**Descriptions**	**Features (Additional Features)**	**Functions (Additional Functions)**
OWLSFacet	OWLSFacet	object for facet	*parent [OWLSPolyhedron] nNodes *nodeIdxs color [Color] area	draw fill print unitNormalMC unitNormalWC facetColor getFacetArea whichSide intersect reflect
	OWLSCell	cell object for watershed model	(*parent) [OWLSWatershed] (marked) (nEdges) (*edgesIdx) (*edgesWeight) (value) (info) [sCellInfo]	(save) (read) (print) (getCellInfo) (getWeight) (nodeInCell) (nodeOnCell) (getCellArea) (averageAspect) (unitNormalMC)
OWLSPoint	OWLSPoint	point object	x y z w	operator + operator - operator *
	OWLSGauge	gauge object for watershed	(name) (dataFile) (nRecords) (startTime) [OWLSTime] (endTime) [OWLSTime] (step)	
	OWLSNode	node object for watershed model	(d1) (d2)	(save) (read)

In the Table 1 - 1, OWLSFacet and OWLSPoint are object modules; each is a stand-alone object group within the model. There are many "induced" objects, for example, OWLSCell is induced from the OWLSFacet (a geometric object) and represents the unit area for watershed model. The OWLSCell is not only a geometric object, but also a watershed object with the additional features like *weights* for the edges, *value* for the cell, relations (pointers) to other watershed objects and functions like *save*, *read*, *getWeight*, etc. Similarly, the OWLSNode is a point object for watershed model with the addition of soil depths (*d1*, *d2* for two layers) and functions. The OWLSGauge is a watershed object for gauge stations (i.e., water gauge, rain gauge, and air temperature gauge). It has a name, time range, etc., which its parent object (OWLSPoint) does not include.

1.4. *System of Linkages*

In the OWLS program, objects are linked in basically three formats. Table 1 - 2 demonstrates the examples of these formats.

Table 1-2. Examples of object linkages in the OWLS model.

Main Object	**Internal Linked Object**	**Inherited Linked Objects**	**External Linked Objects**
OWLSWatershed		OWLSObject	OWLSNode OWLSEdge OWLSCell OWLSenTree OWLSBiTree OWLSPath OWLSPathNode
OWLSHydrology	OWLSFlow OWLSCloud	OWLSWatershed	OWLSRain OWLSTemp OWLSGauge OWLSSoil OWLSVegetation
OWLSStream			OWLSWatershed OWLSSegment OWLSStream

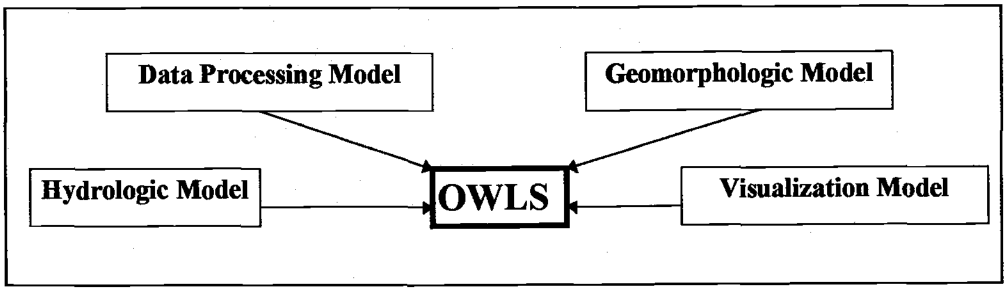


Figure 1-1. Object Watershed Link Simulation (OWLS) Models

(1) Internal linkage: One object is included within another. For example, the object OWLSFlow is included within object OWLSHydrology. It becomes one of the features of OWLSHydrology. When the object of OWLSHydrology is called in the OWLS model, its internal object OWLSFlow will be called. automatically. Features of OWLSFlow are automatically transferred to OWLSHydrology (in-to-out linkage).

(2) Inherited linkage: One object is "inherited" from another object. For example, the object OWLSHydrology is the inherited object from OWLSWatershed, which is also inherited from the general OWLS object OWLSObject. Parameters (features) of OWLSWatershed will automatically pass to OWLSHydrology (parent-to-children linkage).

(3) External linkage: One object contains a member acting as a gateway to another object. Such a member is also called a *pointer*, or an External linkage in OWLS' terminology. In such cases, some functions of the object can use parameters from another object through this linkage without complex analytic procedures. External linkage not only makes the cell-to-cell connection possible and is accomplished relatively easily, but also assists in the connection of flow paths and stream networks in the OWLS model.

A complete list of the linkages in the OWLS model is included in Appendix II.

1.5. *Entire Watershed*

The OWLS model is comprised of four primary models (or sub-models) (Figure 1-1). The *Data Processing Model* is a set of modules handling the conversion of raw data into OWLS' data format. The *Geomorphologic Model* is the combination of several modules, including a data conversion module which converts topographical data into 3-D vector data, a flowpath module to delineate catchment boundary, flow path and stream network, and so on. The *Hydrologic Model* is a group of modules simulating different hydrological aspects in the watershed processes, including a precipitation model, an interception model, a solar radiation model, an infiltration model, an evaportranspiration model, a macropore model, a surface water routing model, a soil water routing model, a channel water routing model, and so on. The *Visualization Model* is a group of modules whose functions are to display the digital data format onto a computer screen or as a printout. The OWLS 3-D visualization model is used to develop a three dimensional view of the watershed.

1.6. *Current State-of-the-Technology*

The distributed model concept can be traced back to mid-60's. Many of the early computer-based rainfall-runoff models recognized that the spatial variability of catchment characteristics needed to be accounted for but they did so only in a relatively crude functional manner (e.g. the infiltration function of the Stanford Watershed Model, Crawford and Linsley, 1966). Most current physically-based distributed models are based on the simplified mathematical formations of Freeze and Harlan (1969). This simplification has been essential for examining realistic problems due to the computational burden of simulating the fully three-dimensional dynamics of catchment hydrology. A great deal of this burden occurs in any treatment of partially saturated soil water systems because of the high nonlinearity of the process necessitating fine temporal and spatial discretization in any numerical scheme. Therefore, different simplifications have been adopted into current distributed hydrologic models to reduce the amount of calculations. For example, the Système Hydrologique Européen (SHE) model (Bathurst, 1986) treats unsaturated soil water flow as a principally vertical process forming a link between surface and saturated subsurface hydrologic components. Such an approximation makes physically-based simulations possible for very large scale catchments, although the validity of the effective parameter values that must be used with large scale catchments has been questioned (Beven, 1989). The Institute of Hydrology Distributed Model (IHDM, Beven et al., 1987) is an example of another style of model, which assumes the downslope flow components of partially saturated near-surface soils may be an important contributor to the storm hydrograph and that the subsurface system is approximated by a two dimensional, vertical slice, solution to the variably saturated flow equations for a number of independent hillslope planes. These simplifications have shown to have reasonable success for simulation of watershed discharges, in particular with regard to understanding the mechanisms controlling contaminant movement in the subsurface environment (Binley and Beven, 1992).

There is an increasing trend in the groundwater literature of undertaking fully three-dimensional modelling studies. There are still, however, few examples of three-dimensional catchment hydrology dynamic simulations even though this once computationally prohibitive exercise is now becoming a realistic option due to the now widespread availablity of fast computers, in particular those exploiting vector and parallel architectures. Binley and Beven (1992) presented results of a three-dimensional catchment hydrology simulation based on a numerical investigation of the response of a heterogeneous Darcian headwater using finite element method. Sophisticated layout for element nodal points, boundary requirement and vast amount of matrix calculation make this type of application slow in simulation and a lack of flexibility in application to other areas.

Since local topography has a strong influence on the site-soil water balance, there is a trend to develope a set of new approaches to simulate runoff generation by taking the digital terrain information into account. There have been many efforts involving the digital terrain analysis in the past 15 years; and these can be classified into two categories: (1) Raster-based and (2) Vector-based.

Most of the efforts on digital terrain analysis utilized a raster-based approach. Greysukh (1967) introduced a method of pixel classification by inspecting the eight-connected pixels adjacent to a cell and computing the sequence of elevation deviations from the central pixel. This stategy has been further improved by Puecker and Douglas (1975) who found that certain features were difficult to accurately extract and systematic errors of classification were often observed. Band (1986) and Douglas (1986) followed on this strategy and applied a standard binary thinning algorithm associated with a tree structure data representation and a steepest descent tracing method to obtain a connected stream network. Jenson (1985) used a similar method to identify potential drainage cells. Toriwaki and Fukumura (1978) introduced connectivity numbers and curvature coefficients for classificatioin of pixel elements which provide the one-pixel-wide ridge and ravines from raster terrain data. Beven and Kirkby (1979) constructed a semi-automated method of calculateing drainage area per unit contour length by manually constructing a set of intersecting contour and slope lines to form a set of connected triangular and quadrilateral hillslope areas. The constributing drainage areas are then accumulated across the downslope boundaries of each unit, and divided by the mean gradient at each contour segment. O'Loughlin (1986) and Moore et al. (1988) have further automated this approach by digitizing contour lines, then tracking the slope lines from equally spaced intervals along each contour to a ridge, peak or intersection with another slope line, resulting in a dense overlapping set of upslope drainage areas. Martz and Garbrecht (1993a, 1993b) developed the Digital Elevation Drainage Network Model (DEDNM) to extract the drainage network and watershed data from digital elevation model database.

Vector-based digital terrain analysis became possible after mid-80s when faster computer processing were available. O'Loughlin presented the basic argorithm for the TOPOG model in 1986, which later developed into the TAPES-C model (Moore and Grayson, 1991). Different from the Raster-based model, the TAPES-C model generates a set of attributes for terrain elements from a vector-based contour database. These attributes include area, upslope contribute area, upslope and downslope element indexs, center coordinate, downslope boundary midpoint, average slope, width of the upslope and downslope bounadry, aspect, and plan curvature. Since terrain elements (or flow net) are bounded by adjacent streamlines and contours, flux from one element can only pass to its downslope element through the downslope boundary. Thus, 2-D flow problem has been simplified into an 1-D problem (hillslope direction). Each element is represented by its midpoint on the upslope and downslope contour lines. The linkage between upslope and downslope element points forms a simplied network for a watershed. Vector-based digital terrain using Triangular Terrain Model (TIN) is another type of methodology. Jones et al. (1989) presents an argorithm to delineate the watershed information (stream, flowpath, and boundary) by tracing the path of steepest descent from a given starting point on a triangular terrain model. This argorithm has been adopted by the OWLS model (Chapter 2). Tachikawa et al. (1994) also presented a similar argorithm for the TIN-DEM data structure.

Beven and Wood (1983) and O'Loughlin (1981) attempted to derive the distributions of drainage area and slope gradient for hillslopes by approximating them with a set of idealized geometric forms including planar, cylindrical and conic sections. While this allows rapid production of the frequency distributions once the surfaces are fitted, these idealized forms probably do not capture the terrain form with much accuracy. A number of researchers have used such area-accumulation algorithms to directly parameterize TOPMODEL (Beven and Kirkby, 1979) using grid DEM (e.g. Band and Wood, 1986, 1988; Wood et al., 1988). The ability to automate the partition of watersheds into different subcatchments and hillslopes directly from DEM has given a significant boost to distributed watersed modeling strategies.

Many of the applications which couple terrain information to a hydrologic model utilize the TOPMODEL (e.g. Zhang, 1994; Band et al., 1991; Familiagetti and Wood, 1990; Robson et al., 1993; Durand et al., 1992, Beven et al., 1984, Charirat and Delleur, 1993). This model is based on using raster cells which are commonly in the form of square cells and have a discrete formation. Other researchers have also developed different models based on a similar raster data format (Wigmosta et al., 1994). Major difficulties that arise in this endeavor involve the translation of continuous concepts (e.g. flow, stream channels) into discrete terms (Band, 1993).

Common practices utilized in raster-based distributed hydrologic models include: (1) using the center of the element (cell) as the representation of the cell, with features like area, slope, length, width and associated hydrologic parameters, and (2) water is balanced at the cell center and the generated flow, as a point source, is distributed to downslope cells under the restriction of 8 possible directions to 8 neighbor elements (O'Callaghan and Mark, 1984). Costa-Cabral and Burges (1994) indicated this practice has large errors in the computed contributing areas and developed an alternative method called Digital Elevation Model Networks (DEMON). DEMON calculates the total contributing area for a rectangular grid of DEM pixels based on the plan-view area concept. The algorithm to find the plan-view area is similar to the flow path tracing algorithm (Jones et al., 1989). No references have been found that illustrate the coupling of DEMON with a hydrologic model.

There are numbers of studies illustrating the use of TAPES-C with a hydrologic model (Dawes and Short, 1994; Barling et al., 1994; Zhang and Montgomery, 1994; Goodrich and Woolhiser, 1994; Smith el al. 1994). The governing flow equations are solved at each nodal point in the element nodal network generated from the TAPES-C (Moore and Grayson, 1991). The concept later became the THALES model (Grayson and Moore et al., 1992a, 1992b), which introduced vector based terrain data into hydrologic modeling. It uses one of the following three approaches to determine the inflow discharge and cross-section area to an element with multiple tributary elements: (1) sum the tributary outflow cross-sectional areas to provide the inflow cross-sectional area to the element; (2) sum the tributary discharges to the element and calculate an equivalent flow cross-sectional area based on the properties of the element; and (3) assume flow is channelized and that only the upslope tributary element with the dominant discharge contributes directly to inflow and that the other elements become lateral inflow to the channel as it passes through the element. Since flows generated from the 1-D nodal points do not drain into a vector-based stream channel, instead into an "channelized element", the THALES model is basically a raster-based model.

As Band (1993) concluded in his review article: "future development of distributed watershed-modeling strategies will involve equal and simultaneous treatment and consideration of model development and the techniques to extract and distribute parameters from a combination of image and geographic processing techniques. At this stage, the digital terrain analysis involved in stream network extraction can also be extended to the parameterization of runoff-producing areas and can be considered an extension of the distributed simulation strategy."

1.7. *Contributions of the OWLS Model*

Comparing to other physically-based watershed hydrologic models, the OWLS model has new features that are not available in the reviewed models (Figure 1 - 3).

Table 1 - 3. Contributions of the OWLS model

No.	New Features	Significance, advantages, or scientific problems that were solved
1.	Object oriented model structure	1. Dynamic Memory Allocation, saves space and run faster; 2. Clearly structured, easy to add new functions and implement new simulation efforts; 3. Code reusable and expandable, reduces the size of the program.
2.	Three dimensional representation:	1. Three Dimensional watershed objects: nodes, edges, cells, stream, catchment boundary, flowpath. 2. Retains all topographical features throughout the model; 3. Enables automatic watershed delineation; 4. Enables terrain analysis and automate extraction of hydrologic parameters; 5. Enables 3-D visualizations.
3.	Vector-based hydrologic watershed modelling	1. No restriction on flow directions, flow is directed by the aspect of the cell (element); 2. Flows are generated from the cell area instead of from the cell center point; 3. Directly uses cell geometric parameters instead of being calculated from nodal points. 4. Stream channel network is represented by vector-based segments; 5. Water flows into channel segment through the riparian cells which are geometry-defined in vector-based model; 6. Water flows into stream segment instead of into channelized elements; 7. Directly uses segments geometric parameters instead of calculated from center nodes; 8. Enables automated watershed delineation: stream, boundary; 9. Enables automated extraction of hydrologic parameters: slope, length, width, aspect, upslope drainage area, upslope cell/segment, downslope cell/segment, neighbor cells, distance to the stream outlet, etc. All these are very difficult to obtaine and manipulate in raster-based models; 10. Enables the intergration of digital terrain information into watershed hydrologic model; 11. Enables the application of Equivalent Rectangle Simplification (ERS) (see 5).
4.	Equivalent Rectangle Simplification (ERS)	1. Converts the 2-D flow routing problem into 1-D without changing the element (cell) shape; 2. ERS can handle elements (cells) with different shapes and different sizes in hydrologic modeling. 3. Provides maximun flexibility to the distributed watershed hydrologic simulation: The hydrologic model can be coupled to watersheds with a rectangle grid from DEM, or triangle cells from TIN, or flow-tube cells from TAPES-C or mixtures of cells from the OWLS automatic delineation model. As long as each cell remains planar, the sizes and the shapes of the cells can be different from watershed to watershed, or within a watershed.

Table 1 - 3. Contributions of the OWLS model (Continued)

No.	New Features	Significance, advantages, or scientific problems that were solved
5.	Finite Different Approximation Throughout	1. 2-D surface flow and soil flow routing using kinematic wave approximation in association with ERS; 2. 2-D macropore flow routing using energy and continuity equation in association with ERS; 3. Unconditionally stable and convergent; 4. Simulation time steps can be varied without largely changing the simulation results or affecting stability of results.
6.	Unit compatible	Allows English or SI unit system for input and output.
7.	Watershed Macropore Flow Simulation	1. Identifies macropore flow from soil flow or surface flow. 2. Individual flow generation and routing machinism totally different for surface flow and soil flow. 3. Avoids unrealistic amplification of the hydrologic conductivity coefficient or surface water proportion as often occurs in other models that cover up the natural responses of the soil macropore system.
8.	Three-Dimensional Visualization	1. Enables sky-view of the watershed, as well as a 2-D map view; 2. Enables the watershed to be viewed from different angles; 3. Enables the watershed to be viewed at different time; 4. Enables the watershed to be viewed dynamically; 5. Enables the watershed characteristics to be visually presented: contour, soil, topography, flowpaths, flowpath tree, channel network, boundary; 6. Enables the simulation result to be visually presented: (1). Dynamic stream hydrology: stream discharge, stream flow velocity, segment width, segment water depth; (2). Dynamic watershed hydrology: distribution of cell total flow depths (variable source area), distribution of cell flow components (surface, soil and macropore), distribution of cell water components (canopy water, canopy snow depth, surface water, surface snow depth, soil water depth, soil moisture, macropore pipe water depth), distribution of cell vertical fluxs (canopy ET, surface ET, soil ET, infiltration).
9.	Two-Dimensional Visualization	1. Presents an intergration of system parameters; 2. Demonstates hydrologic inputs, simulated and observed discharge (hydrograph); 3. Dissects the flow components and water components conditions as a funtion of time over the watershed; 4. Provides assistance in model calibration; 5. Offers in-depth hydrologic information about the watershed.

Chapter Two: *Automatic Watershed Object Delineation*

The delineation of a watershed boundary is an important modelling problem for several reasons. These include:

A. A cell on the watershed boundary may have the flow going out-ward in stead of in-ward, and a cell outside of the watershed boundary may also have water going into the watershed. If the watershed boundary is determined from the a raster-based algorithm, its calculated area may be larger or smaller than it actually is. For many models, the watershed boundary is assumed known and seldom field-verified.

B. Similarly, the stream network may not be properly represented from digitized data. A digital line segment may not be hydrologically correct since the water from one side of the line may be able to pass through the line to the other side (like the line crossing through a slope of a cell). For many models, the watershed's stream network is also assumed to be known, so that any water arriving at a stream cell (for a raster based model) or stream line (for a vector based model) will become stream input no matter what the slope condition is. Raster-based models can identify the cell which may contain the stream, but not the stream itself. Both types of information are required by the vector-based OWLS model.

The OWLS' watershed delineation model obtains inputs from watershed topographical data (e.g. Digital Elevation Model Data or DEM from USGS) or a watershed vectorized database (node, edge and cells relational data). By selecting a predefined watershed outlet-node of interest, the OWLS model calculates the watershed boundary, flow path, and stream network in the vector-based data format. The major difference between the OWLS' watershed delineation model and other models as mentioned in Chapter 1 is that the OWLS' model is vector-based: watershed boundaries and streams are represented as point-and-line vectors. Each segment of the line contains information about their slope, direction, neighborhood cells, and upper drainage area for a stream/flowpath line. In addition, the OWLS model can also indicate not only the current stream network, but also the flowpaths of a fully extended or potential stream network.

2.1. *Flowpath*

A flowpath is the trail of water running downslope from a specific origin and along the surface of a watershed. Because water runs from a high elevation to a low elevation by the force of gravity, it will always choose the steepest slope as its path way. Therefore, the OWLS algorithm for determining the flowpath from a cell assumes that the path of steepest descending slope can approximate flow paths over a surface and will precisely indicate the direction in which flow will be initiated over a homogeneous surface.

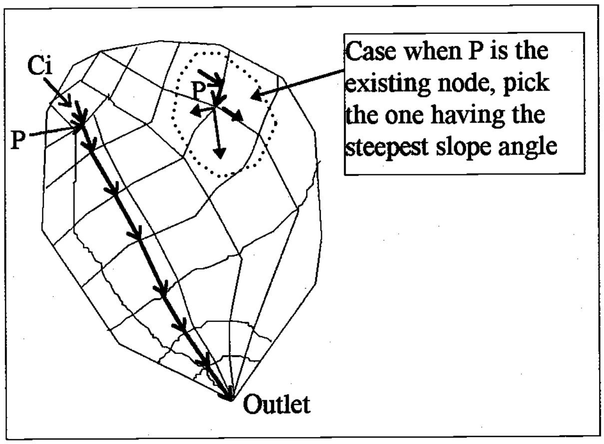


Figure 2 - 1. Flowpath from cells.

The OWLS' algorithm used to identify watershed flowpaths is modified from the algorithm used for triangle-based terrain model (Jones et al., 1990). It is described as follows:

(1) Take the center point (Ci) of a cell of a watershed as a start point;

(2) Find a point (P) on the cell's boundary so that it forms a vector with the center point (Ci->P) and which has the same direction as the cell's aspect (Figure 2-1).

(3) Record the line as the first segment of the flowpath from the cell (Ci->P);

(4) Take P as the next start point;

(5) If P is on an edge, repeat step (2) to (4) until P is on the study boundary;

(6) If P is on an existing node, repeat step (2) for all its adjacent cells (assume n cells) so that we have (P->P1, P->P2, ..., P->Pn). Pick the one who has the largest slope angle (like P->Pi) and record it as step (3). Then take Pi as the next start point, repeat step(2) to (4) until P is on the study boundary.

By traversing through every cell of a watershed, a cluster of hair-like flowpath lines is formed.

2.2. *Watershed Boundary*

The watershed boundary is dependent upon the selection of a watershed outlet point. To determine the watershed boundary from a given stream outlet, a reversed algorithm is used to subdivide the cells so that all cells within the boundary have in-ward flow (Figure 2 - 2):

(1) Take the given assumed stream outlet point (Oi) as the start point;

(2) Find the points (P1, P2, .. Pn) on the nearby cells (n cells) so that each of them forms the vector with the start point (P1->Oi, P2->Oi, ..., Pn->Oi) and which has the same direction as its corresponding cell's aspect;

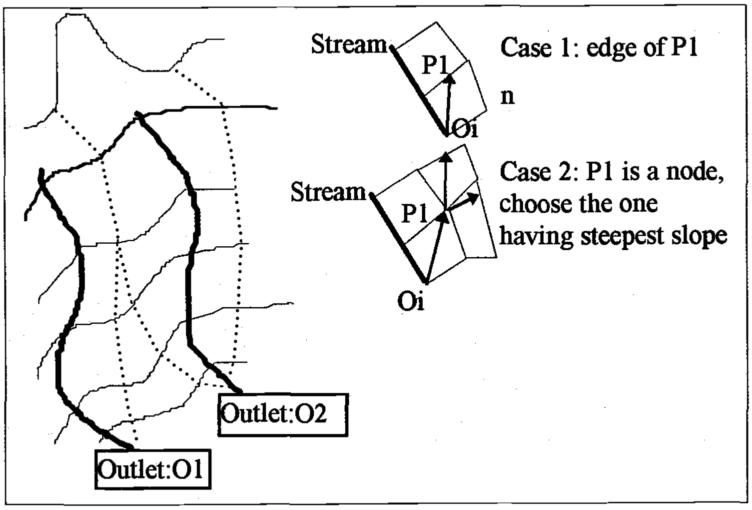


Figure 2 - 2. Schematic of catchment boundary algorithm.

(3) Pick the point with the largest possitive slope angle (assume Pi);

(4) If Pi is on an edge, split the cell into two parts with one part inside the catchment and the other part out;

(5) Take Pi as the start point and repeat step (2) to (4) until no Pi has a positive slope angle (i.e. until a topographic peak or ridge is encountered);

The above algorithm can usually only determine partial watershed boundaries (typically two sides). After completing this algorithm, the original cell map will have been altered by the addition of subdivided cells along the watershed boundary. However, by applying the flowpath algorithm to all cells, including subdivided cells, we can identify all the cells that have their flowpaths running through the stream outlet point (Oi). From the results of these operations, the actual boundary of the watershed can be established.

2.3. *Stream Network*

The term "stream" has a dynamic connotation, especially for small watersheds. Depending upon water inputs (rainfall/snowmelt) and soil conditions, stream length, width, number of branches, etc. can vary through time. In the OWLS program, a watershed's stream network is defined as the potential flow-collection pathway. Thus it is not necessary to have water present in a channel or pathway all the time.

A stream network is represented by a tree of nodes-and-paths, each node represents a stream cross-section and each path represents a stream segment (Figure 2 - 3). The algorithm for finding the stream network is relatively complex and is based on the results from the flowpath algorithm. Finding the stream network needs to accomplish two major tasks: (1) convert flowpaths into a flowpath tree and (2) trim single-source branches from the flow path tree.

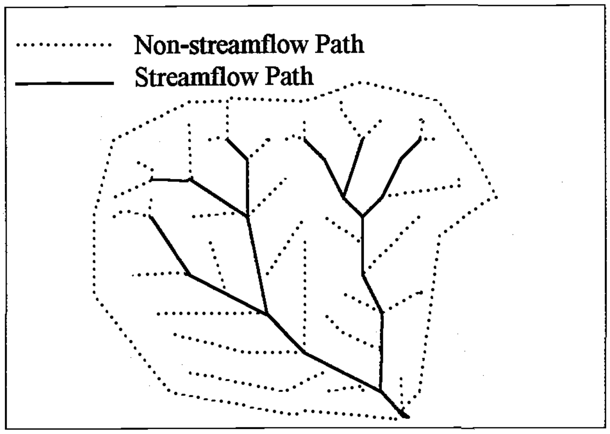


Figure 2 - 3. Stream network tree structure.

2.3.1. Convert flowpaths into a flowpath tree

In terms of tracing direction, a flowpath goes from up-to-down in elevation. But for the flowpath tree, the tracing direction is in the opposite direction. The basic concept of this algorithm is to individually add each flowpath to the flowpath tree. Many factors need to be considered at the point of joining a flowpath to the flowpath tree. Following are the major steps that are utilized in the OWLS program to convert flowpaths into a flowpath tree:

A. For all flowpaths passing through the watershed outlet, mark them as IN;

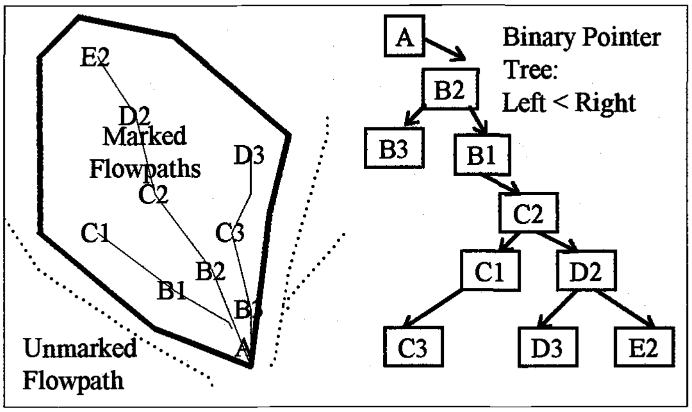


Figure 2 - 4. Marked flowpaths and binary pointer tree.

B. For fast calculation, create a binary pointer tree, in which each node points to the flowpath node of the flowpath tree, and the left-hand-side node always has a smaller elevation than the right-hand-side node (Figure 2 - 4). Before adding a new flowpath into the flowpath tree, we need to establish if a portion of the new flowpath has been represented in the flowpath tree to avoid overlapping. To do so, each node of the new flowpath is compared with nodes in the flowpath tree. This can be a very time comsuming task if it becomes necessary of search through all the nodes in the flowpath tree. However, utilizing the binary pointer tree, all nodes in the flowpath tree have been organized into certain order to fit the binary tree. This type of organization essentially accelerates the searching and comparing processes by going through only a few nodes in the binary tree.

C. Inserting a flowpath into the flowpath tree (see Figure 2 - 5 for simple demonstration) :

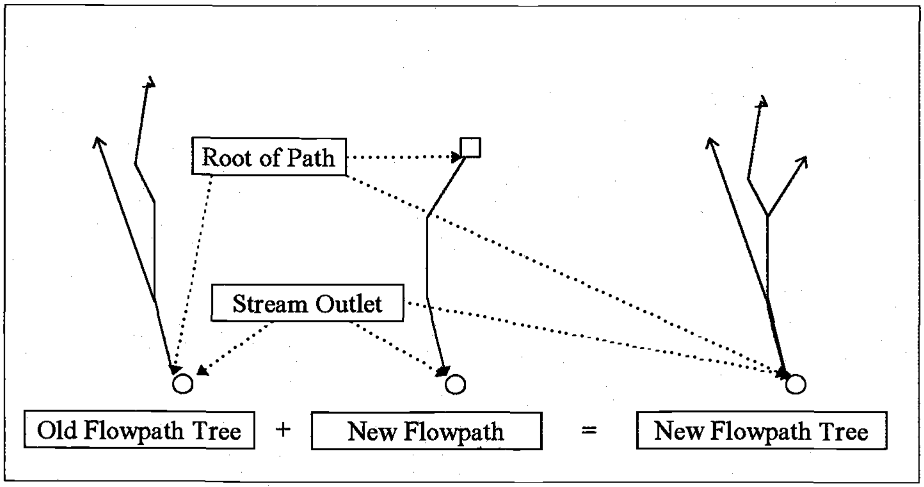


Figure 2 - 5. Simple flowpath insertion to flowpath tree.

1). For the new flowpath, which originates from the center of a cell (the root node) and passes downslope to the stream outlet, we need to create a pointer that points to the stream outlet node.

2). Check if this node is in the flowpath tree using the binary pointer tree.

2.1). If not, the program will initialize the flowpath tree and insert the stream outlet node as the root. Then all parential edges and nodes in the flowpath are inserted into the flowpath tree as child edges and child nodes. At the same time, initialize the binary pointer tree, define the root pointer, and create child pointers in the same manner as the flowpath tree. Then go on to step 4.

2.2). If yes, this means that the flowpath tree exists and the node is part of the tree. No action should be taken to the flowpath tree. If there is a parent node in the flowpath, move the pointer to the parent node and continue to step 2, otherwise, jump to step 4.

3). Check to determine if the node of the new flowpath is in the flowpath tree using the binary pointer tree.

3.1). If not, there will be several cases:

a) the node is on an edge of the flowpath tree.

a.1) when the node is the begining of a new branch of the flowpath tree (the parent of this node is neither in the flowpath tree nor on an edge of the flowpath tree), insert this node into the flowpath tree and create the branch as well as to the binary pointer tree.

a.2) when the node and its parent form an edge that overlaps the edge in the flowpath tree, then move the pointer to the parent node of the flowpath and continue to step 3.

b) the node is not on any edge of the flowpath tree, but instead is a new branch of the flowpath tree. Insert the rest of the flowpath into the flowpath tree and create a series of new pointers in the binary tree pointing to inserted nodes. Then continue to step 4.3.2) If yes, this means that the flowpath tree exists and the node is in the tree. There will be two cases:

b.1) the parent of this node forms an edge with this node is an existing edge in the flowpath tree. No adjustments of the flowpath tree are needed, however, move the pointer of the flowpath to the parent node (upslope node). If there is no more parent node in the flowpath, move on to step 4.

b.2) if the formed edge is not an existing edge in the flowpath tree, it is a new branch. Insert the node and the rest of the flowpath into the flowpath tree.

4). Move on to next available flowpath and start from step 3 until all marked flowpaths are evaluated and the conversion is finished.

When adding a flowpath to the flowpath tree, the characteristics of each flowpath node are calculated. Thus, when the flowpath tree is completed, we also have the detailed information about the flowpath node (e.g., upper-drainage area, elevation, length to the stream outlet, slope of the flowpath edge).

2.3.2. Trim the single-source branch of the flowpath tree.

To qualify for being a stream segment, a portion of the flowpath should receive water from at least two sources. However, to delineate individual flowpaths, an algorithm was developed to trim a single-source branch from the flowpath tree. The result is the potential stream network of the watershed; each node of this network may become a flowpath or stream when there is a sufficient supply of water to the system (see solid-line portion of Figure 2 - 3). The flowpath tree provides much information about the watershed surface and hydrologic system. Furthermore, we can select a filter to identify stream segments that meet specific criteria. For example, we can choose an upper-source area as a filter and then determine which stream segments have an upper drainage area meeting that criteria. Similarly, filters related to river-mile, slope, etc., can be used. From the stream network tree, we are also able to identify where the stream should be. By combining with the OWLS' hydrologic model, other filters such as water depth in channel segments can be used so that we are able to dynamically similate the stream network during a storm event.

2.4. Foundation for Watershed Hydrologic Simulation

Results from the automatic watershed object delineation become the vector-based watershed distributed object database. This database includes watershed boundary, stream segments, and network, cell geometry, and so on. It become the foundation of the hydrologic simulation in the OWLS model.

Chapter Three. *System of Equations*

Physically-based hydrologic models are designed to simulate water movements within both the hillslope and stream channel of a watershed. They can provide information on real-time flow for specific watershed objects, which in turn may be important for basin water chemistry simulation or other purposes. The structure of the physically-based hydrologic model developed for this study is represented in Figure 3-1 and consists of dozens of other functional models.

3.1. *Input Distributing Model*

Depending upon the availibility if data, the OWLS model provides two options to handling distributive watershed input data:

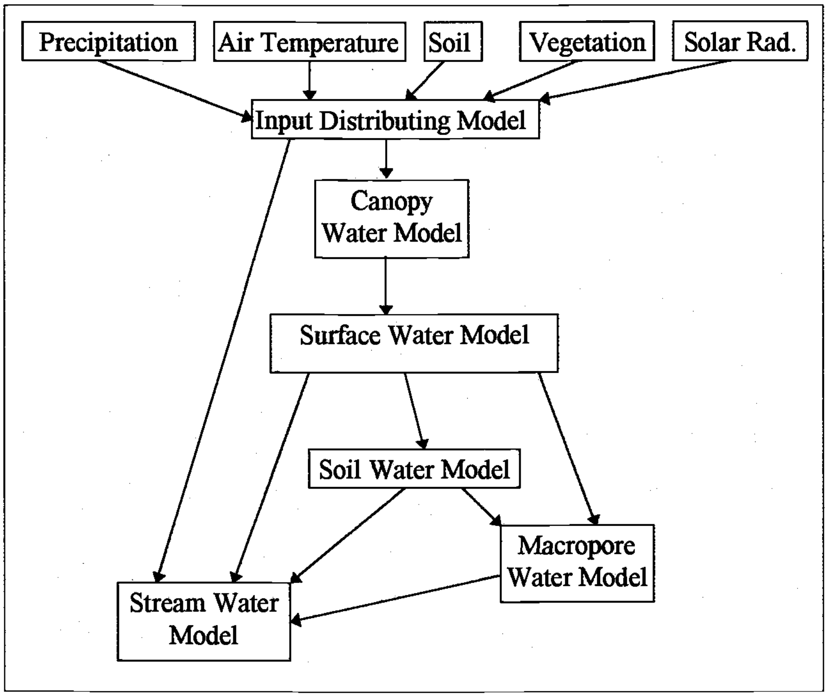


Figure 3 - 1. OWLS' physically-based hydrological model structure.

SIMPLE model option: Under this option, precipitation input, air temperature, soil characteristics (infiltration parameter, porosity and soil hydraulic conductivity) and vegetation characteristics (leaf area index, interception ratio, evapotranspiration ratio) will be considered as homogeneous over the watershed.

COMPLEX model option: When distributed watershed soil and vegetation characteristic data are available, the OWLS model is able to use this distributed data for its hydrologic simulation. Also, if a watershed has data from more than two meteorologic stations, the COMPLEX model option will use the precipitation distribution model and air temperature distribution model to create distributed precipitation and air temperature. Since many air temperature data are available in daily characteristic values (minimun, maximun and average), the OWLS model also includes an air temperature extension model (ATEM) to simulate the instantaneous air temperature.

Watershed evapotranspiration (ET) is essentially by the incoming solar radiation levels. The OWLS solar radiation model calculates solar radiation for each individual cell. Thus, different cells may receive different solar radiations depending upon the time of the day, and the slopes and aspects of the cells. Details of these distribution models are as follows:

3.1.1. Precipitation Distributing Model

The precipitation distribution model will be utilized only when there are at least three rain gauge stations available in the watershed. The model will perform linear space interpolation to distribute precipitation data from nearest three gauge stations to a cell (Figure 3-2).

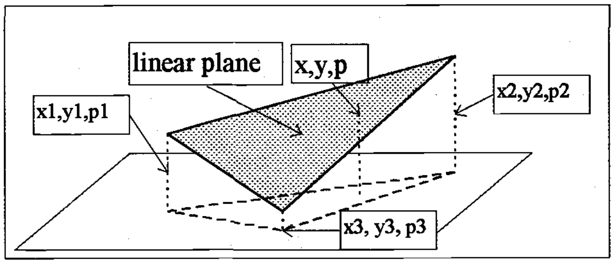


Figure 3 - 2. Linear space interpolation model

The mathematical equations to solve the interpolated precipitation at point (x, y) are:

$$S_1 = 0.5 \times \begin{vmatrix} x & y & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \qquad 3 - 1$$

$$S_2 = 0.5 \times \begin{vmatrix} x_1 & y_1 & 1 \\ x & y & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \qquad 3 - 2$$

$$S_3 = 0.5 \times \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x & y & 1 \end{vmatrix} \qquad 3 - 3$$

$$p = (p_1 \times S_1 + p_2 \times S_2 + p_3 \times S_3)/(S_1 + S_2 + S_3) \qquad 3 - 4$$

where, p_1, p_2, p_3 are the precipitation of rain gauge 1, 2, and 3 located at (x_1, y_2), (x_2, y_2) and (x_3, y_3); p is the precipitation of a cell which has a center at (x, y);

When a cell is located outside the triangular range of the gauges, the Linear Space Interpolation Model will perform linear extrapolation. The results from the extrapolation will then be identified for abnormal values. The method to determine the abnormal values is by comparing the estimated value to the normal data range (minimun to maximun) of all rain gauges. In some cases, the result from extrapolation may be too high (over a certain percentage of the maximun value) or too low (negative). The OWLS model will then adjust the high value into the value of the nearest gauge and the low value as zero precipitation.

3.1.2. Air Temperature Distribution Model

Air temperature in mountainous terrain is more dependent upon elevation than horizontal location. For a watershed having more than one temperature gauge, the Air Temperature Distribution Model will perform an ambient lapse rate (averageed -0.65^{0}C/100m) calculation (for only one gauge available), vertical linear interpolation (for two gauges available), or linear regression (for more than two gauges available) (Figure 3 - 3).

When more than two gauges available, the air temperature of a cell is calculated as:

$$T(z) = (a_T \times z) + b_T \qquad 3 - 5$$

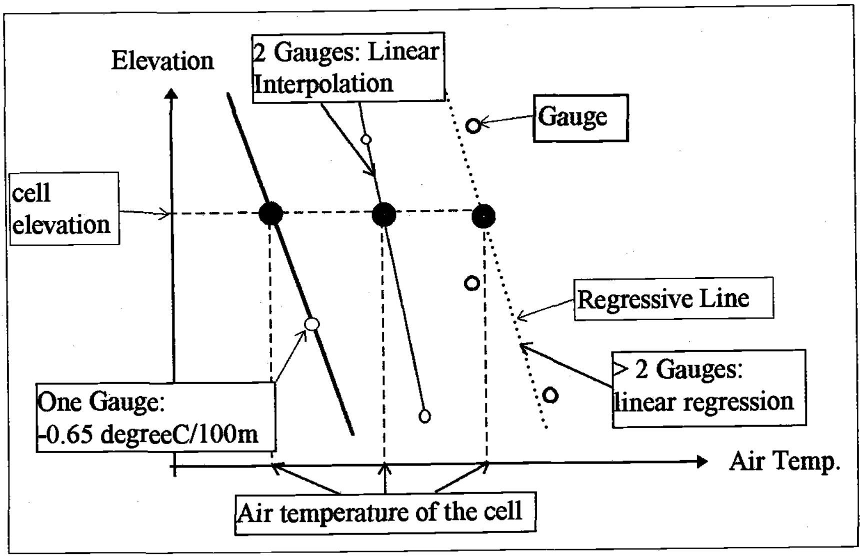


Figure 3 - 3. Air temperature distribution model.

where $T(z)$ is the air temperature at elevation z;

a_T and b_T are regression coefficients which are calculated as Equation 3 - 6 and 3 - 7;

$$a_T = \frac{\sum_{i=1}^{n}((T_i - \overline{T}) \times (z_i - \overline{z}))}{\sum_{i=1}^{n}(z_i - \overline{z})^2} \qquad 3 - 6$$

$$b_T = \overline{T} - (a_T \times \overline{z}) \qquad 3 - 7$$

where, n is the total number of air temperature gauges;

T_i is the air temperature (°C) at gauge i;

z_i is the elevation (m) of the gauge i;

3.1.3. Solar Radiation Model

The Solar Radiation Model consists of three portions: (a) Extraterrestial Solar Radiation; (b) Cloud Attenuation; and (c) Canopy Reduction (Lee, 1978; Black, 1956; Ross and Tooming, 1968 and Monteith, 1973):

$$\begin{array}{c} I = I_0 \times Z_a^{\sec(Z)} \times \cos(B) \times (0.8 - 0.34C - 0.46C^2) \times e^{-\gamma LAI/\cos(Z)} \\ |\leftarrow \mathit{Extraterrestial} \rightarrow| \; |\leftarrow \quad \mathit{Cloud} \quad \rightarrow| \; |\leftarrow \mathit{Canopy} \rightarrow| \end{array} \qquad 3-9$$

where,

Extraterrestial calculation adopts equations from Lee (1973), includes instantanuous solar constant, atmospheric effect and slope effect;

Cloud reduction calculation adopts equation from Black (1956);

Canopy reduction calculation adopts equation from a semi-empirical exponential formula proposed by Ross and Tooming (1968) which has been further theoretically proven by Monteith (1973);.

I is the solar radiation (W/m^2) received on the top of a hillslope surface (cell); for solar radiation recieved by the canopy, the last portion should be removed;

I_0 is the solar constant (W/m^2) calculated by Equation 3 - 10;

Z_a is the zenith path transmissivity (or atmosphere turbidity);

Z is the solar zenith (°) calculated from Equation 3 - 11 to 3 - 13;

B is the solar incidence (°) calculated from Equation 3 - 14 to 3 - 16;

C is the cloudiness measured as fraction of sky covered, in tenths;

LAI is the leaf-area-index (m^2/m^2) for the surface vegetation;

γ is the canopy reduction coefficient. Values of γ range from 0.21 to 0.6 depending on the canopy structure and solar elevation. In the OWLS model, $\gamma = 0.5$.

$$I_0 = I_a \times (1 + 0.033 \times \cos(\frac{360 \times (t_j + 10)}{365.25})) \qquad 3-10$$

here, I_a is the mean solar constant (=1367 W/m^2);

t_j is the Julian day;

$$Z = \arccos(\sin(L_a)\sin(S_d) + \cos(L_a)\cos(S_d)\cos(S_t)) \qquad 3-11$$

here, L_a is the latitude (°) of the location;

S_d is the solar declination (°) calculated by Equation 3 - 12;

S_t is the solar hour angle (°) calculated by Equation 3 - 13;

$$S_d = -23.5 \times \cos(360 \times \frac{t_j + 10}{365.25}) \qquad 3-12$$

$$S_t = \arccos(-\tan(L_a)\tan(S_d)) \qquad 3-13$$

$$B = \arccos(\cos(Z)\cos(S_c) + \sin(Z)\sin(S_c)\cos(S_{az} - S_{laz}))$$

3 - 14

here, S_c is the slope (°) of the cell;

S_{az} is the solar azimuth (°), which is calculated as following:

when solar hour $t_s < 12$ (morning):

$$S_{az} = arccos(cos(L_a)\, sin(S_d) + sin(L_a)\, cos(S_d)\, cos(S_t) / sin(Z)) \qquad 3-15$$

when solar hour $t_s >= 12$ (afternoon):

$$S_{az} = arccos(360 - cos(L_a)\, sin(S_d) + sin(L_a)\, cos(S_d)\, cos(S_t) / sin(Z)) \qquad 3-16$$

3.1.4. Air Temperature Extension Model

For many meteorological stations, mean, maximun and minimun air temperature are typically available. The OWLS model includes an Air Temperature Extension Model (ATEM) to use such information to provide temperature estimates needed for short-term hydrologic processes. The ATEM assumes that daily temperature changes are continuous and periodic, and utilizes a sine function to simulate this process (Figure 3 - 4).

The following equation represents the model used to simulate temperature patterns over time:

$$T(t) = T_{avg} + b \times \sin(\frac{(t - t_{max} + 6)\pi}{12}) \qquad 3-17$$

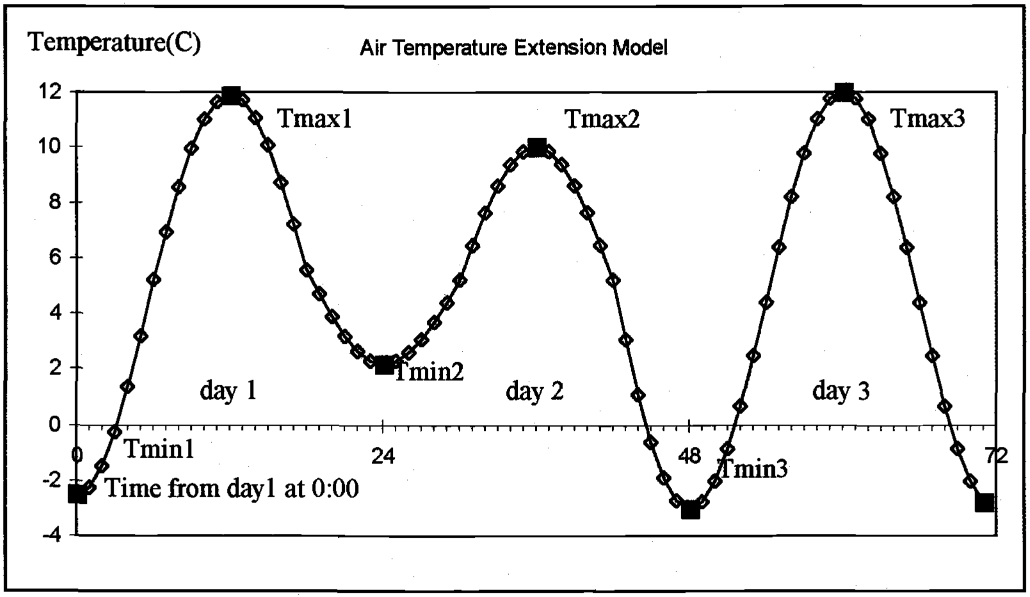


Figure 3 - 4. Air Temperature Extension Model.

where, T_{avg} is the averaged daily air temperature;

$T(t)$ is the temperature at time t;

b has different values under different time in a day:

$b = T_{avg} - T_{min}$ at night time (usually take $t < 6$ or $t > 18$);

$b = T_{max} - T_{avg}$ at day time ($6 \leq t \leq 18$);

t_{max} is the average time when daily maximun air temperature is recorded.

In order to maintain continuity, in the afternoon, T_{min} should take that of next day and T_{avg} is determined over a 2-day period.

3.2. *Equivalent Rectangle Simplification (ERS).*

The OWLS model uses an *Equivalent-Rectangle-Simplification* (ERS) method to establich a physical interrelation between cells for the routing of surface flows. Thus, the model is capable of handling a variety of cell patterns within a basin. The ERS method is used to simplify the geometry of a cell, which is represented as a polygon with n edges and n nodes, into a rectangle which has the same soil and vegetation, same area, same slope, same center location, and same total length (or total width, or width-to-length ratio) as the cell (Figure 3 - 5). Each edge of a cell has a weighting, which is determined by the relative area of a given cell providing water to that edge (Figure 3 - 6). This weighting was used to determine the amount of water that could cross a particular edge (zero when none, -9 identifies an upper edge that is receiving water from an upslope cell). By assuming that the physical performance of the cell can be approximated by that of its equivalent rectangle, hydrologic information can be calculated for the equivalent rectangle and then distributed to the edges by their relative weightings (e.g., discharge) or directly assigned to the edges (e.g., water depth).

The terminology "equivalent" means both cells have the same area, same slope, same soil and vegetation condition, same soil depth, some center location, same aspect and both are planar, so that they will have same amount of precipitation inputs, same solar radiation inputs, same infiltration rate, same surface water depth, same soil moisture content, same amount of flow generated from the surface, soil and macropore system. However, they can be different in shape and consequently the pattern of flow draining from each cell could be different.

An equivalent rectangle for an irregular cell is constructed so that it satisfies the above conditions. In order to implement a one-dimensional hydrologic calculation, the rectangle also needs to have two sides parallel to the aspect direction in addition to an upslope boundary and a downslope boundary.

There are an infinite number of rectangles possible to satisfy the requirements, but of these there are three types of rectangles are probably the most reasonable choices for an "equivalent rectangle". These are:

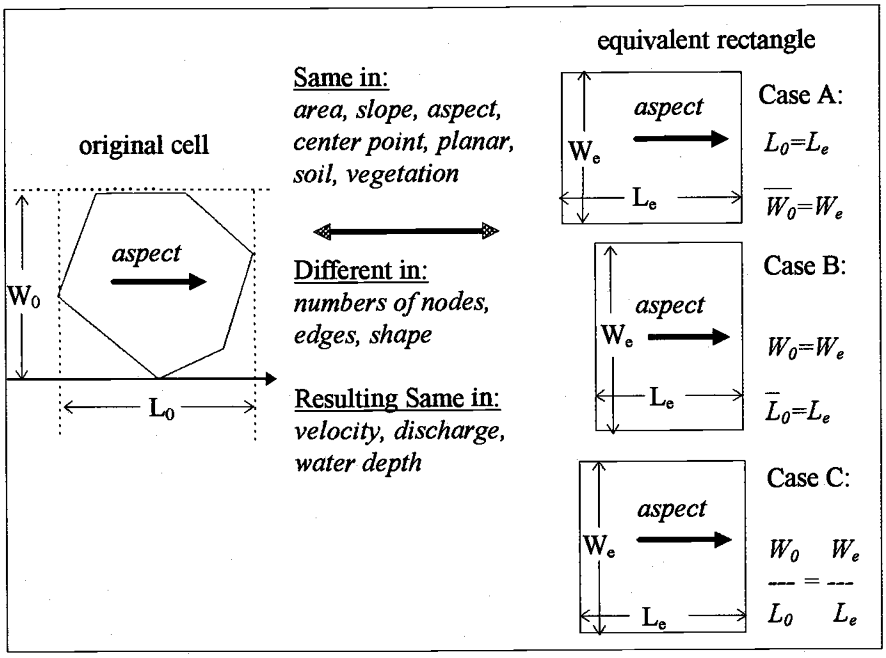
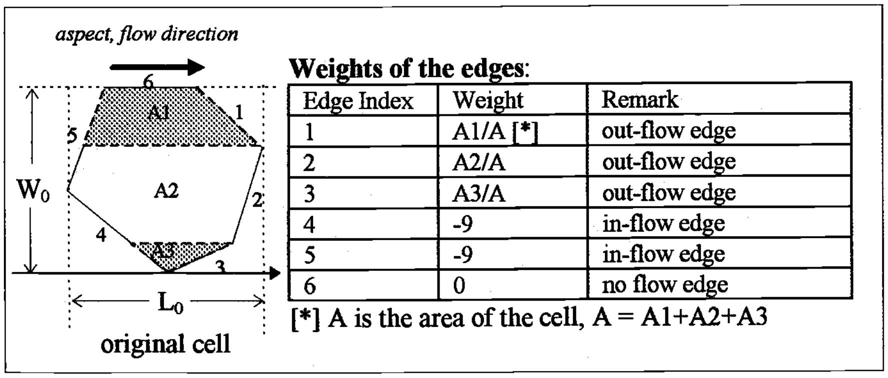


Figure 3 - 5. Equivalent rectangles.



Weights of the edges:

Edge Index	Weight	Remark
1	A1/A [*]	out-flow edge
2	A2/A	out-flow edge
3	A3/A	out-flow edge
4	-9	in-flow edge
5	-9	in-flow edge
6	0	no flow edge

[*] A is the area of the cell, A = A1+A2+A3

Figure 3 - 6. Edge weights of a cell.

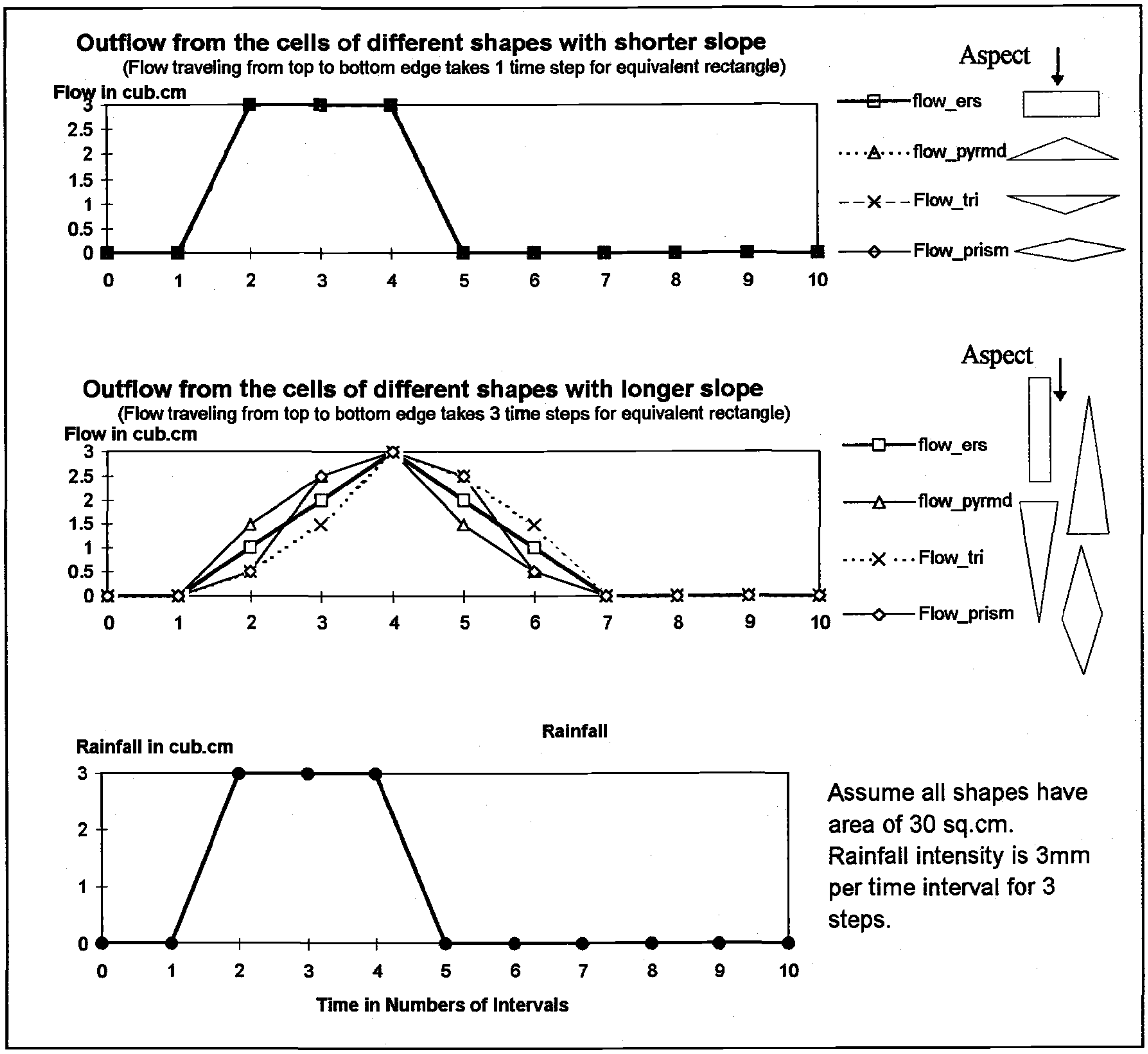
A. A rectangle having the length equal to the projected length of the cell on the slope direction.

B. A rectangle having the width equal to the projected width of the cell on the contour direction.

C. A rectangle having the same length:width ratio to the projected length:width ratio of the cell.

Then which rectangle is the best approximation to the cell hydrologically?

Figure 3 - 7 and 3 - 8 demonstrate an analysis of surface runoff routing for equivalent rectangles with type A (same length) and type B (same width) for several cell shapes (triangle and prism shapes were selected for ease of analysis). In both figures, an assumed rainfall event of 3 mm per time step with a duration of 3 time step has been applied at time steps 2, 3, and 4. The cells of different shapes are assumed to be planar and no diffusion occurs during flow routing along the surface. For both figures, there are two group of cells, one is shorter in slope length and the other is longer, which will requires


Outflow from the cells of different shapes with shorter slope
(Flow traveling from top to bottom edge takes 1 time step for equivalent rectangle)

Outflow from the cells of different shapes with longer slope
(Flow traveling from top to bottom edge takes 3 time steps for equivalent rectangle)

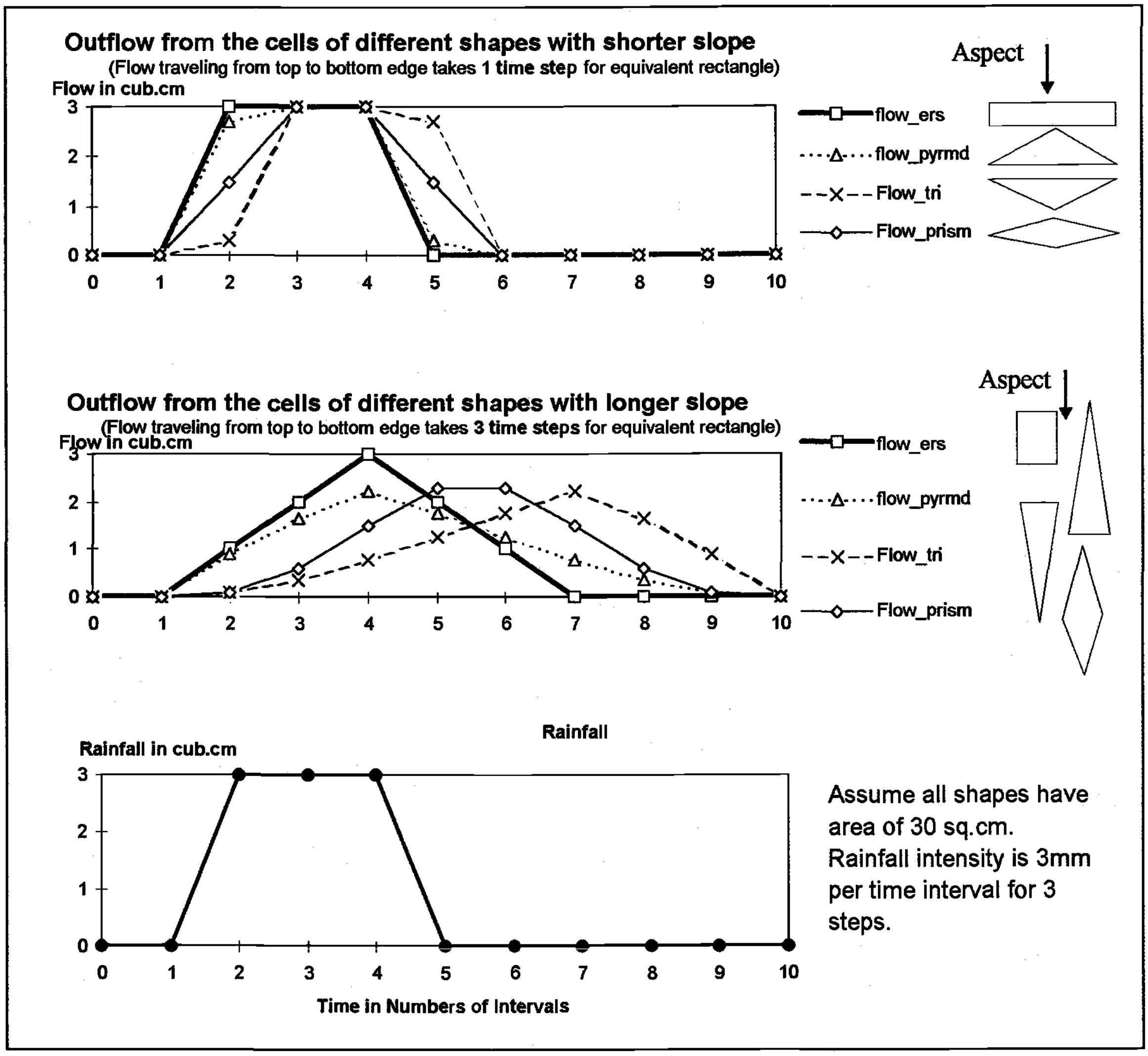
Rainfall

Assume all shapes have area of 30 sq.cm. Rainfall intensity is 3mm per time interval for 3 steps.

Figure 3 - 7. Equivalent Rectangle Simplification, Equal Length ERS.

more then one calculation time step to route the generated flow out of a cell. Each group has three cells with pyramid, triangle and prism shape respectively, representing the cells with wider downslope boundary, wider upslope boundary and wider center body. All the cells are assumed to be 10 cm^2 in area and are impermeable, each cell will expect to generate 3 cm^3 of flow from each time step during the rainfall period. Taking into account the time consumed by flow routing, "hydrographs" were then be calculated using a spread-sheet.

In Figure 3 - 7, all cells within a given group have the same length even though shapes are varied. For



Assume all shapes have area of 30 sq.cm. Rainfall intensity is 3mm per time interval for 3 steps.

Figure 3 - 8. Equivalent Rectangle Simplification, Equal Width ERS.

the group of cells with a shorter slope length, runoff responses are instaneous and all cells produce the same hydrograph. For the group of cells with relatively longer slopes, pyramid shaped cell tends to have a faster rising limb and slower falling limb; triangle shaped cell have reversed a runoff pattern; and prism shaped cell tends to smooth the hydrograph peak. Notice the duration of runoff for the different cell shapes are the same. The Equivalent Rectangle, however, produces flow in a linear manner and represents the average situation for the group of cells.

In Figure 3 - 8, all cells with different shapes have been constructed to have the same width. For the group of cells with shorter slope length, runoff responses are quick but varied. Let us assume that the equivalent rectangle has a slope length such that one calculation time step is required to drain all its water. Since other cells have different shapes, which consequently increases the length of the cells in order the have the same area, more then one calculation time step will be required to drain water from these cells. As shown in the Figure 3 - 8, pyramid shaped cells can be reasonable equivalent by the rectangle, but hydrographs from cells with triangle and prism shapes will be delayed nearly-one-step relative to the rectangle. For the group of cells with relative longer slopes, this advanced outflow phenomina of the rectangle becomes more obvious. In addition, flow from the rectangle tends to have a higher instantaneous peak than cells with any of the other shapes.

For a type C rectangle, which has the same width-to-height ratio, we may expect outflow patterns to occur between those found for type A and B cells. Flow advancing and a higher peak of the rectangular shape may also be expected. Therefore, we can conclude that equivalent rectangle should have the same length as the slope length of the cell it attempts to approximate. Even so, in the OWLS model, options for ERS characterizations are available among these three types so that they can be further evaluated. The default ERS in the OWLS model is equal length.

3.3. *Canopy Water Model*

In forested watersheds, the vegetation canopy (Figure 3 - 9) is the first layer of the terrestrial ecosystem interfacing with atmosphere. In terms of water balance, the canopy performs interception and evapotranspiration functions, which are basically acting

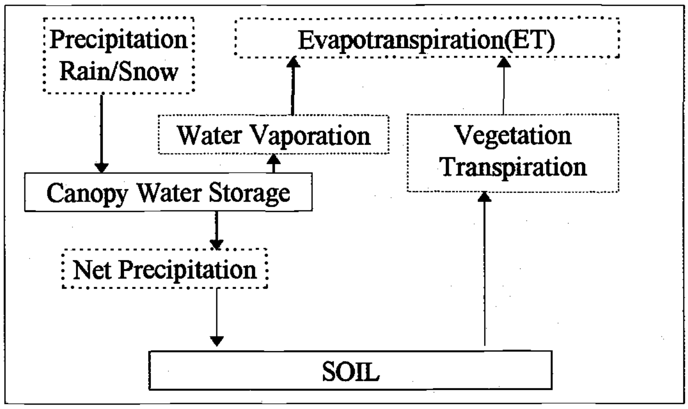


Figure 3 - 9. Canopy water model structure.

vertically; the OWLS model does not consider any horizontal water/vapor transfer within the canopy. For different vegetation species or different time periods, the ability of vegetation to influence the interception and evapotranspiration are also varied. The OWLS model attempts to simulate many of these temporal variations.

3.2.1. Interception

Canopy interception includes rainfall interception and snowfall interception. For snowfall interception, the OWLS model uses equivalent water depth to represent the snowpack on the canopy instead of actual snow depth. From information about the vegetation's leaf-area-index (LAI) for each species and for each month of a year, the interception capacity of a particular canopy for both rain or snow can be determined as a function of LAI:

$$S_r(t) = S_{r0} \times K_{canopy} \times LAI(t) / LAI_0 \qquad 3-18$$

where, $S_r(t)$ is the canopy interception capacity (m) at time t;

S_{r0} is the maximum canopy interception capacity (m) of the vegetation in a year;

$LAI(t)$ is the leaf-area-index (m^2/m^2) of the vegetation at time t;

LAI_0 is the maximum LAI (m^2/m^2) for the vegetation in a year;

K_{canopy} is the canopy density (m^2/m^2) in the cell of concern.

Actual interception by the canopy during a simulation is determined by amount of precipitation and the deficit of the canopy water storage $SC_d(t) = (SC_0 - SC(t-\Delta t))$, which is shown in Table 3 - 1.

Table 3 - 1. Actual interception amount determined by precipitation and storage deficit

Actual Interception Amount	$S_r(t) > SC_d(t)$	$S_r(t) <= SC_d(t)$
$S_r(t) > P(t)$	MIN{P(t), $S_d(t)$}	P(t)
$S_r(t) <= P(t)$	$SC_d(t)$	MAX{P(t), $SC_d(t)$}

* SC_d = deficit of canopy water storage (m);
SC = canopy water storage (m);

3.3.2. Evapotranspiration

Evapotranspiration (*ET*) amounts from different landscape layers are based on the values of potential evapotranspiration (ET_0). In the OWLS program, ET_0 is defined as the maximum water vapor flux volume (m) for a free water surface above the vegetation canopy. ET_0 is estimated using the equation derived by Jensen and Haise (1963):

$$ET_0(t) = (a_E \times T(t) + b_E) \times S_{ec}(t) \times C_s \times \Delta t \qquad 3-19$$

where, a_E and b_E are empirical constants. Jensen & Haise picked $a_E=0.025$ and $b_E = 0.078$;

$S_{ec}(t)$ is the solar radiation (W/m^2) received at the top of the canopy at time t, calculated from Equation 3 - 9 by excluding the canopy portion;

C_s is a constant of the latent heat of vaporization at 20°C ($1.471x10^{-6}$ m^3/W/hr), which converts solar radiation from W/m^2 into unit of equivalent depth of evaporation (Christiansen, 1966);

Δt is the time step of calculation (hr).

Evapotranspiration (ET) from the vegetation canopy includes two portions: Water Evaporation (ET_{re}) and Canopy Transpiration (ET_{rt}). ET_{re} indicates the amount of water evorporated from water held on the surface of canopy leaves. ET_{rt} indicates the amount of water transpired through canopy leaves. ET_{re} and ET_{rt} can be estimated by Equation 3 - 20a and 3 - 20b.

$$ET_{re}(t) = ET_0(t) \times K_{canopy} \times LAI(t) / LAI_0 \qquad 3-20a$$

$$ET_{rt}(t) = (ET_{r0} \times \Delta t + ET_0(t)) \times C_{et} \times K_{canopy} \times LAI(t) / LAI_0 \qquad 3-20b$$

where,

$ET_{re}(t)$ is the potential evaporation (m) from water on the leaf surface of a canopy at time t;

$ET_{rt}(t)$ is the canopy potential transpiration (m) at time t;

ET_{r0} is the minimum canopy transpiration rate (m/hr) which represents the night-time canopy transpiration when no solar radiation occurs ($ET_0=0$);

$ET_0(t)$ is the potential ET (m) at time t, calculated by Equation 3 - 20 under the condition of air temperature and solar radiation for the time and species of concern;

C_{et} is the vegetation ET ratio which represents the relation between potential ET and vegetation ET when canopy has maximum LAI and full coverage;

Δt is the calculation time step (hr).

Equation 3 - 20a estimates the potential ET_{re} instead of actual ET from the canopy. However, the actual amount of ET_{re} from a canopy is limited by the amount of water remaining in the canopy ($S_c(t-\Delta t)$). If ET_{re} is larger than $S_c(t-\Delta t)$, than all the water in the canopy will be evaporated, so the actual $ET_{re} = S_c(t-\Delta t)$ and the amount of water remaining in the canopy is zero: $S_c(t)=0$. Similary, Equation 3 - 20b estimates the potential ET_{rt}. The actual amount of ET_{rt} is limited by the supply of water from the soil (ST_{st}), which is a function of soil moisture condition:

$$ST_{st}(i,t) = ST_0 \times \theta_r(i,t) - c \times \sin(2\pi\theta_r(i,t)) \qquad 3-21$$

where, ST_{st} is the soil water supply ability (*m/hr*) to canopy transpiration;

ST_0 is the rate of water supply from soil to canopy when soil is saturated;

$\theta_r(i,t)$ is the volumetric relative soil moisture content;

c is a parameter representing the curvature of the maximun offset to the linear line.

If ET_{rt} is larger than $ST_{st}(i, t)$, than the actual $ET_{rt}(i, t) = ST_{st}(i, t)$, otherwise, the actual $ET_{rt}(i, t)$ will be the ET_{et} calculated from Equation 3 - 20b.

In addition to supply limitations, two conditions influence ET estimates in the OWLS program: (1) whenever there is a precipitation input, both ET_{re} and ET_{rt} are set to be zero since *ET* is assumed to be not significant during rainfall; (2) when there is intercepted water on the surface of canopy leaves, the actual canopy transpiration is zero, or $ET_{rt}(i, t) = 0$.

Given the above conditions, the equation of evapotranspiration from a canopy ($ET_r(i, t)$) is represented as:

$$ET_r(i,t) = ET_{re}(i,t) + ET_{et}(i,t) \qquad 3-22$$

3.3.3. Snowmelt

The OWLS model can also consider the snowmelt process in the canopy. The water equivalent of snow on the canopy is approximated by the interception model (Equation 3-18). A simple modified degree-day model is employed to calculate snowmelt amounts:

$$M_s(t) = D_f \times (T(t) - T_b) \qquad 3-23$$

where, $M_s(t)$ is the snowmelt at time t (m/hr);

D_f is the degree-day snowmelt factor (m/°C/hr), which is in the range of 3.6 to 7.3 mm/°C/day from an Iowa watershed (Haan, 1982). By considering the effective day length is 12 hours, the degree-day factor can be converted to a degree-hour factor with a range of from $3\text{x}10^{-4}$ to $6\text{x}10^{-4}$ m/°C/hr; In the OWLS model, this parameter will be calibrated.

$T(t)$ is the air temperature (°C) at time t;

T_b is the base air temperature (°C) when the snow start to melt.

After each flux of the canopy has been calculated, the Canopy Water Model undertakes a water balance to calculate the net rainfall ($P_n(T)$):

$$P_n(t) = P(t) + M_s(t) - ET_r(t) - I_r(t) \qquad 3 - 24$$

where, *P(t)* is the precipitation (m) at time t.

3.4. *Surface Water Model*

3.4.1. Infiltration

In the OWLS model, infiltration includes two parts: (1) infiltration from surface to soil and (2) seepage from soil surface to soil macropore system. Potential infiltration is calculated from a modified Horton Model (Equation 3 - 25). The actual infiltration amount is then determined by comparing potential infiltration and the available surface water, whichever is smaller.

$$f(i,t) = f_c(i) + (f_0(i) - f_c(i))e^{-kT_h(i,t)} \qquad 3 - 25$$

here, $f(i,t)$ is the potential infiltration rate (m/hr) at Horton time T_h;

$f_c\ (i)$ is the minimum infiltration rate (m/hr) when soil is saturated;

$f_0\ (i)$ is the maximum infiltration rate (m/hr) when soil is in field capacity (no gravitational water);

k is the infiltration coefficient (1/hr);

$T_h(i,t)$ is a equivalent time which is a function of the soil relative moisture content:

$$T_h(i,t) = -ln(1 - \theta_r(i,t)) / \alpha \qquad 3 - 26$$

$\theta_r(i,t)$ is the soil relative moisture content, which is related to the soil volumetric moisture content (θ) as $\theta_r = \theta / p_s$, p_s is the porosity;

α is a relational constant which can be obtained from soil experiment or calibrated in the model.

Figure 3 - 10 shows the value of α and its effect to the relation between relative soil moisture content and the equivalent time in Horton's equation.

Figure 3 - 11 shows the values of α and its effect to the relation between the soil moisture and infiltration rate.

Since there is lack of research on how moisture moves from a soil surface to the soil macropore system, the OWLS model simply assumes the seepage from surface to soil macropore system to be a function of infiltration:

$$f_{sm}(i,t) = C_{sm} \times f(t,t) \qquad 3 -27$$

here, C_{sm} is an empirical coefficient that needs to be calibrated in the model.

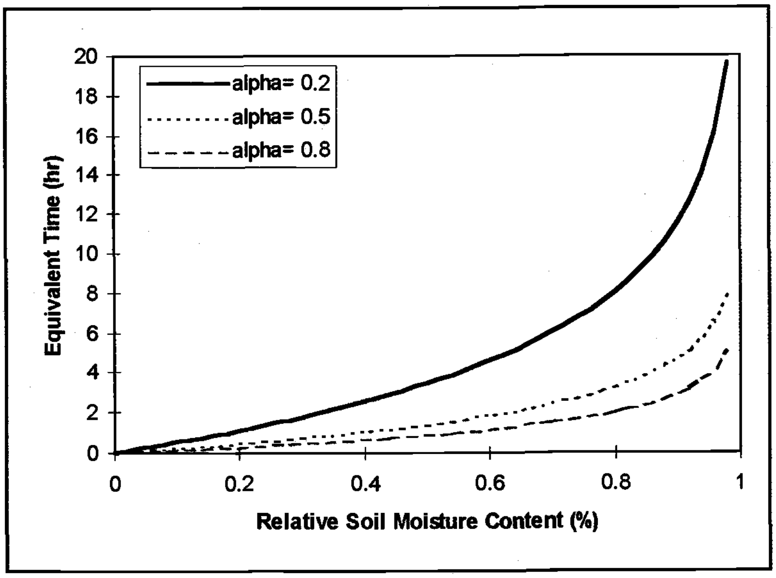


Figure 3 - 10. Relationship of soil moisture content and α to equilvant time

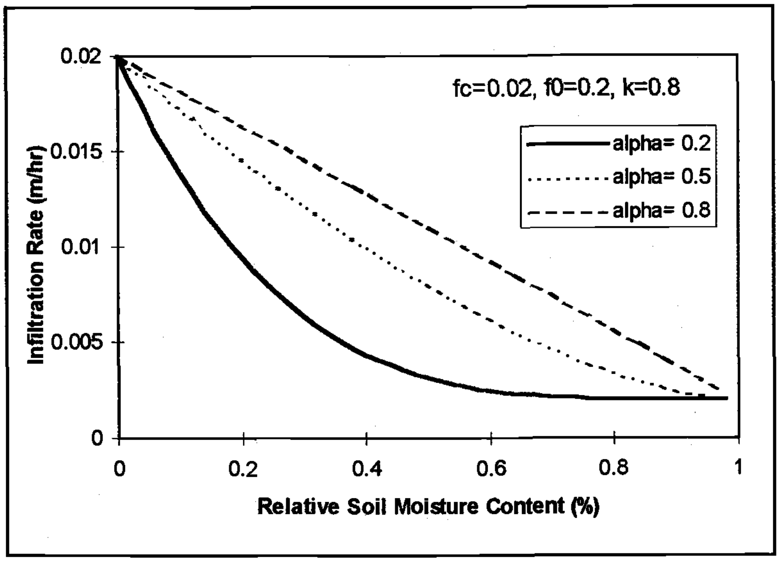


Figure 3 - 11. Relationship of soil moisture content and α to infiltration rate

3.4.2. Evaporation from the soil surface

In the OWLS model, evaporation from the soil surface (ET_s) will be set at zero during periods of precipitation. Similar to the ET calculation in the canopy, ET_s consists of two portions: ET from water on the surface (ET_{sw}) and ET from the soil surface (ET_{gs}). Both portions are calculated from the potential evapotranspiration (ET_0) with the consideration of canopy coverage and soil moisture condition:

$$ET_{sw}(i,t) = ET_0(t,LAI) \qquad 3 - 28a$$

$$ET_{gs}(i,t) = ET_0(t,LAI) \times C_{gs} \times \theta_r(i,t) \qquad 3 - 28b$$

where,

$ET_0(t, LAI)$ is the potential evaporation (m/hr) from free water under the canopy at time t calculated from Equation 3 - 9 by including the canopy portion;

C_{gs} is the soil evaporation ratio which represents the relation between potential ET and soil ET when a soil is saturated;

3.4.3. Surface Flow

Water movement on a sloping surface can be mathematically described by the St. Venent equations (Chow, 1988), including the continuity equation (3 - 29) and momentum equation (3 - 30).

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial l} = r \qquad 3 - 29$$

where, h is the depth of water on the surface (m);

q is the unit-width discharge (m^2/hr);

r is the vertical net incoming flux (m/hr);

l is the length of the slope (m);

t is the time (hr).

The OWLS model employs the Kinematic Wave form of the momentum equation:

$$S_f = S_0 \qquad 3 - 30$$

where, S_f is the friction slope;

S_o is the slope of the surface.

The surface flow rate is calculated by Manning's equation (Chow et al., 1988):

$$v = S_f^{\frac{1}{2}} h^{\frac{2}{3}} / n \qquad 3 - 31$$

where, *n* is the Manning's roughness of the slope surface.

By substituting the kinematic wave momentum equation (3 - 30) into Manning's equation (3 - 31) and rearranging, we obtain:

$$h = \left(\frac{n}{\sqrt{S_0}} \right)^{\frac{3}{5}} \times q^{\frac{3}{5}} = \alpha q^{\beta} \qquad 3 - 32$$

Here,

$$\alpha = \left(\frac{n}{\sqrt{S_0}} \right)^{\frac{3}{5}} \qquad 3 - 33a$$

$$\beta = \frac{3}{5} \qquad 3 - 33b$$

There are many different numerical methods in solving the St. Venent equations. In the OWLS model, the Nonlinear Kinematic Wave Scheme finite-difference method (Chow et al., 1988) is used for surface flow routing. Thus, the difference equation for continuity equation (3 - 29) becomes:

$$\frac{q(i,t) - q(up,t)}{\Delta l} + \frac{h(i,t) - h(i,t-\Delta t)}{\Delta t} = \frac{r(i,t) + r(i,t-\Delta t)}{2} \qquad 3 - 34$$

By substituting Equation 3 - 32 into Equation 3 - 34 and rearranging, the following equation is obtained:

$$\frac{\Delta t}{\Delta l} q(i,t) + \alpha(q(i,t))^{\beta} = \frac{\Delta t}{\Delta l} q(up,t) + \alpha(q(i,t-\Delta t))^{\beta} + \frac{\Delta t}{2}(r(i,t) + r(i,t-\Delta t))$$

3 - 35

where, *q(i, t)* and *q(up, t)* represent the unit-width discharge (m^2/s) from current cell and upper cell(s).

Since calculations are from up-hill cells to down-hill cells (Figure 3 - 12), discharge(s) from up-hill cell(s) are calculated by:

$$q(up,t) = \frac{\sum_{j=1}^{m} (Q(j,t) \times w_i(j))}{width_{eq}(i)} \qquad 3 - 36$$

here, $Q(j, t)$ is the calculated discharge (m^3/hr) from upper cell *j*;

$w_i(j)$ is the weight of edge *i* in cell *j*;

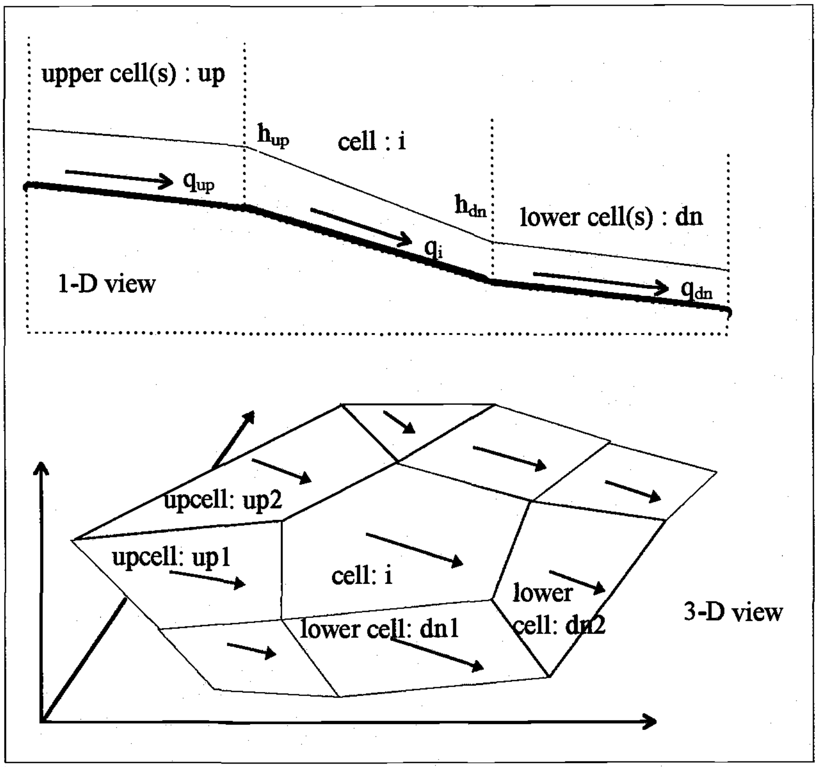


Figure 3 - 12. Inter-cell relations.

m is the total numbers of upper cells;

$width_{eq}(i)$ is the equivalent rectangle width (m) of cell *i*;

$h(i, t)$ and $h(i, t\text{-}\Delta t)$ are the surface water depth (m) for the cell from the current and last time calculations;

Δl is the length (m) of equivalent rectangle for cell *i*;

$r(i, t\text{-}\Delta t)$ and $r(i, t\text{-}\Delta t)$ is the net vertical incoming flux (m/hr) for cell i from the current and last calculation:

$$r(i,t) = P_n(i,t) + M_s(i,t) - f(i,t) - ET_s(i,t) - f_{sm}(i,t) \qquad 3\text{ - }37$$

here, $f(i, t)$ is the infiltration (m) at time t.

$f_{sm}(i, t)$ is the amount of water (m) flowing into soil macropore system at time t.

Equation 3 - 35 is a non-linear equation, and cannot be solved directly. However Newton's method (Chow et al., 1988) can be applied iteratively to obtain a numerical solution. The known right-hand side of Equation 3 - 35 at each finite-difference cell is:

$$C=\frac{\Delta t}{\Delta l}q(up,t)+\alpha(q(i,t-\Delta t))^{\beta}+\frac{\Delta t}{2}(r(i,t)+r(i,t-\Delta t)) \qquad 3-38$$

from which the residual error $R(q(i,t))$ is:

$$R(q(i,t))=\frac{\Delta t}{\Delta l}q(i,t)+\alpha(q(i,t))^{\beta}-C \qquad 3-39$$

The first derivative of $R(q(i,t))$ is:

$$R'(q(i,t))=\frac{\Delta t}{\Delta l}+\alpha\beta(q(i,t))^{\beta-1} \qquad 3-40$$

The object is to find $q(i,t)$ that forces $R(q(i,t))$ equal to 0.

Using Newton's method with iterations k = 1, 2, ...

$$(q(i,t))_{k}=(q(i,t))_{k-1}-\frac{(R(q(i,t)))_{k-1}}{(R'(q(i,t)))_{k-1}} \qquad 3-41$$

The convergence criterion for the iterative process is:

$$\left|R(q(i,t))_{k}\right|\leq\varepsilon \qquad 3-42$$

where ε is an error criterion.

In the OWLS program, the value of ε is defined by a user. The value of ε cannot be too large (>1.0) since it will cause the estimated value to be far from its TRUE solution even though a relatively large value of ε allows calculations to occur rapidly. Similarly, the value of ε cannot be too small ($<10^{-6}$) otherwise we may expect a costly computer running time associated with little improvement in accuracy. The determination of ε is a trial-and-error process and the trade-off between speed and accuracy needs to be balanced. In the OWLS model, ε is set at 0.01 and the number of iterations limited to 100.

In Equation 3 - 41, when $k = 1$, $q(i,t)_0$ is the first calculated value of surface unit discharge. The OWLS model obtains this value from the following linear function:

$$q(i,t)_{0}=\frac{\frac{\Delta t}{\Delta l}q(up,t)+\alpha\beta q(i,t-\Delta t)\left(\frac{q(i,t-\Delta t)+q(up,t)}{2}\right)^{\beta-1}+\frac{\Delta t}{2}(r(i,t)+r(i,t-\Delta t))}{\frac{\Delta t}{\Delta l}+\alpha\beta\left(\frac{q(i,t-\Delta t)+q(up,t)}{2}\right)^{\beta-1}}$$

3 - 43

The initial conditions for the surface flow discharge is defined as 0 in the OWLS model, which means there is no surface flow in the begining of a simulation. The boundary conditions for the surface flow is defined as $q(t) = 0$.

3.5. *Soil Water Model*

The vertical water movement into the soil is assumed to be instantaneous. Therefore, the infiltrated water instantly joins the subsurface water body and any evaporated water is also instantly removed from the subsurface water body.

The horizontal water movement in the soil is the major concern of this section. In the OWLS program, a one-layer, variable-effective-depth soil is used to simulate subsurface flow (Figure 3 - 13). The water content in the soil is not evenly distributed so that the depth of water in the soil changes with the soil moisture conditions. The simulation of subsurface flow utilizes the continuity equation (Equation 3 - 44) and Darcy's Law (Equation 3 - 45):

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial l} = r_g \qquad 3-44$$

$$f_g = D\frac{\partial H}{\partial l} = D(S_0 + \frac{\partial h}{\partial l}) \qquad 3-45$$

where, f_g is the flux (m/hr) in the soil;

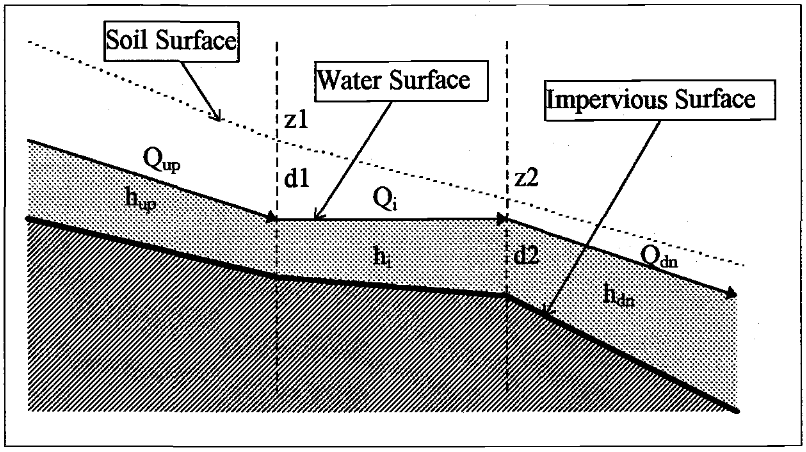


Figure 3 - 13. Soil water model.

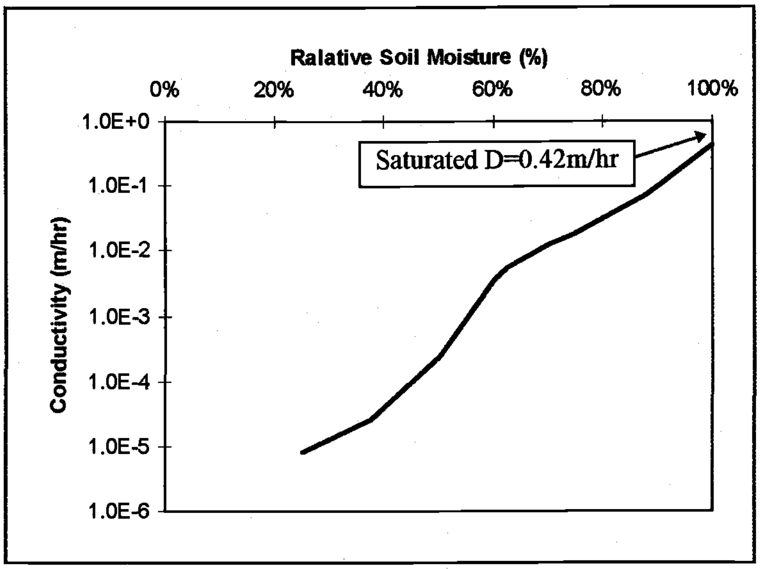


Figure 3 - 14. Unsaturated hydraulic conductivity for a sandy soil

D is the unsaturated soil hydraulic conductivity (m/hr), which should be provided as a list of relational data between volumetric relative soil moisture and the conductivity value for each type of soil. Figure 3 - 14 is an example computed from the data from Jury (1990) and Todd (1980) for a hypothetical sandy soil.

H is the water head (m) in the soil;

h is the soil water table depth (m).

Thus, the unit width discharge (q)is determined by:

$$q = p_s f_g h = Dp_s h(S_b + \frac{\partial h}{\partial l}) \approx Dp_s hS_b \qquad 3 - 46$$

where, p_s is the porosity of the soil; the change of water depth along the slope is assumed to be neglectible.

S_b is the bed-rock slope (°) of the soil which is:

$$S_b = S_0 + \frac{d_1 - d_2}{\Delta l} \qquad 3 - 47$$

here, d_1 and d_2 are the soil depth (m) at the upper and lower boundary of the cell;

S_0 is the slope (°) of the surface.

The net side incoming flux r (m/hr) in the continuity Equation 3 - 29 for subsurface water is calculated as:

$$r_g = f - ET_{gs} - f_{gm} \qquad 3-48$$

here, ET_{gs} is the soil evaporation rate (m/hr) calculated by Equation 3-28b;

f_{gm} is the flux of water (m/hr) moved from soil to macropore system, which is simplified as a function of soil water depth:

$$f_{gm} = C_{gm1} \times h^{C_{gm2}} \qquad 3-49$$

C_{gm1} and C_{gm2} are empirical parameters.

The finite-difference equation for Equation 3 - 44 is:

$$\frac{q(i,t)-q(up,t)}{\Delta l} + \frac{(h(i,t)-h(i,t-\Delta t))\times p_s}{\Delta t} = \frac{r_g(i,t)+r_g(i,t-\Delta t)}{2} \qquad 3-50$$

The cell equation for Equation 3 - 46 is:

$$q(i,t) = D \times p_s \times h(i,t) \times S_b \qquad 3-51$$

Substitute Equation 3 - 51 into the finite-difference equation of continuity Equation 3 - 50, we obtain:

$$\frac{D\times p_s}{\Delta l} S_b \times h(i,t) - \frac{q(up,t)}{\Delta l} + p_s \times (\frac{h(i,t)-h(i,t-\Delta t)}{\Delta t}) = \frac{r_g(i,t)+r_g(i,t-\Delta t)}{2} \qquad 3-52$$

Rearranging Equation 3 - 52 we obtain:

$$(\frac{D\times p_s}{\Delta l} S_b + \frac{p_s}{\Delta t}) \times h(i,t) = \frac{p_s}{\Delta t} \times h(i,t-\Delta t) + \frac{q(up,t)}{\Delta l} + \frac{r_g(i,t)+r_g(i,t-\Delta t)}{2} \qquad 3-53$$

Equation 3 - 53 can be further simplified as:

$$h(i,t) = \frac{p_s \times h(i,t-\Delta t)\times \Delta l + q(up,t)\times \Delta t + 0.5 \times (r_g(i,t)+r_g(i,t-\Delta t))\times \Delta l \times \Delta t}{(S_b \times D \times \Delta t + \Delta l)\times p_s} \qquad 3-54$$

Calculated result from Equation 3 - 54 is the depth of water in the soil of a cell, it will then used to compute the soil relative moisture content (θ_r):

$$\theta_r(i,t) = \frac{h(i,t)}{p_s \times d(i)} \qquad 3-55$$

p_s is the porosity (%) of the soil;

$d(i)$ is the depth of the soil (m) for the cell i;

When a soil is saturated, calculated value of θ_r will larger than 1.0. In this case, the OWLS model will calculate the extra amount of water as the ex-filtration of water from the soil to the surface of the cell. As a result, the soil water depth will be adjusted so that the relative soil moisture content become 100%, or equals to $p_s x d(i)$. The flow from the soil will be calculated using Equation 3 - 51.

3.6. *Macropore Water Model*

The OWLS program use a pipe-bundle model developed from the energy and continuity equations to simulate the movement of water in the soil macropore system (Figure 3 - 15). The energy equation (Gupta, 1989) for macropore pipe flow is:

$$z_1 + h_1 + \frac{v_1^2}{2g} = z_2 + h_2 + \frac{v_2^2}{2g} + h_f \qquad 3 - 56$$

where, z_1 and z_2 are the elevation (m) of the ends of a macropore pipe (1 is upper, 2 is lower);

h_1 and h_2 are the water pressure head (m) on the two ends;

v_1 and v_2 are the velocity (m/hr) at the end of the pipe;

g is the gravitational constant ($=1.27\text{x}10^8$ m/hr^2);

h_f is the friction loss inside the pipe, which can be determined by the Darcy-Weisbach equation (Gupta, 1989):

$$h_f = \frac{C_f l v^2}{2gd} \qquad 3 - 57$$

here, C_f is the friction factor ranged from 0.04 to 0.07 for turbulent flow in a rough pipe (dimemsionless) from the results of Nikuradse's experiment (Gupta, 1989). But for soil macropore pipe system, this value is will be calibrated in the OWLS model;

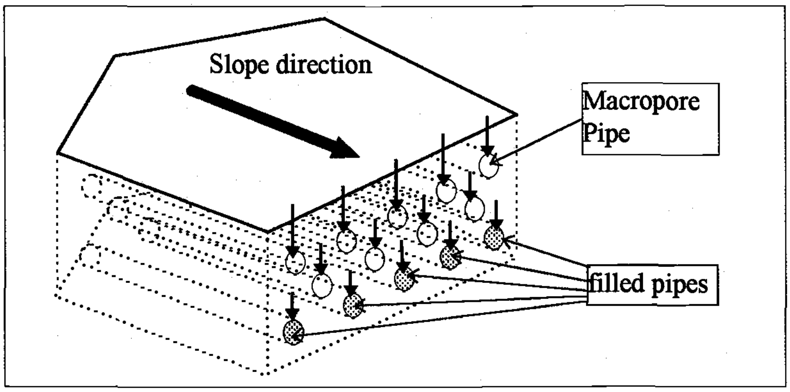


Figure 3 - 15. Macropore flow model.

l is the length of the macropore pipe and is assumed to be the equivalent length (m) of the cell;

v is the averaged velocity (m/hr) of flow in the pipe;

d is the diameter (m) of the pipe.

The continuity equation for the macropore pipe system is:

$$\frac{\partial A}{\partial t}+\frac{\partial Q}{\partial l}=R \qquad 3-58$$

where, A is the total area (m^2) of active macropore pipes, which is:

$$A=a\times N_w \qquad 3-59$$

a is the cross-section area (m^2) of a macropore pipe:

$$a=\pi\times\left(\frac{d}{2}\right)^2 \qquad 3-60$$

N_w is the numbers of macropore pipes filled with water, which is:

$$N_w=N_m\times\frac{V_{mw}}{V_m} \qquad 3-61$$

N_m is the total numbers of macropore pipes in the soil;

V_{mw} is the volume (m^3) of water in the macropore system;

V_m is the total volume (m^3) of the macropore pipe:

$$V_m=aN_m\Delta l \qquad 3-62$$

Q is the discharge (m^3/hr) from the macropore pipes of the cell;

R is the unit-length net incoming flux (m^2/hr) to macropore system:

$$R=(f_{sm}+f_{gm})/\Delta l \qquad 3-63$$

f_{sm} and f_{gm} are flow (m^3/hr) from surface and soil to the macropore system. f_{sm} is proportional to the infiltration rate; f_{gm} is calculated by:

$$f_{gm}=C_{gm}\times D_w\times(1-\frac{V_{mw}}{V_m}) \qquad 3-64$$

where, C_{gm} is the coefficient (1/hr) for soil to macropore flow;

D_w is the water depth (m) in the soil column.

By assuming the water depth is the same in both ends of the macropore pipe, the energy equation (3 - 56) becomes:

$$v=\sqrt{v_{up}{}^2+2g\Delta z-2gh_f} \qquad 3-65$$

The differential equation for the continuity equation (3 - 58) is

$$A(i,t)+\frac{\Delta t}{\Delta l}Q(i,t)=A(i,t-\Delta t)+\frac{\Delta t}{\Delta l}Q(up,t)+(f_{sm}(i,t)+f_{gm}(i,t))\Delta t \qquad 3-66$$

Where, *Q(i,t)* can be described by the energy equation (3 - 65) in the differential form as:

$$Q(i,t)=A(i,t)v(i,t)=\sqrt{\frac{Q(up,t)^2+2gA(i,t)^2\Delta z}{1+C_f\Delta l/d}} \qquad 3-67$$

Therefore, Equation 3 - 66 can be expressed as:

$$A(i,t)+\frac{\Delta t}{\Delta l}\sqrt{\frac{Q(up,t)^2+2gA(i,t)^2\Delta z}{1+C_f\Delta l/d}}= \qquad 3-68$$
$$A(i,t-\Delta t)+\frac{\Delta t}{\Delta l}Q(up,t)+(f_{sm}(i,t)+f_{gm}(i,t))\Delta t$$

which is a nonlinear differential equation. Using Newton's method, assume

$$C=A(i,t-\Delta t)+\frac{\Delta t}{\Delta l}Q(up,t)+(f_{sm}(i,t)+f_{gm}(i,t))\Delta t \qquad 3-69$$

and the residual function:

$$E=A(i,t)+\frac{\Delta t}{\Delta l}\sqrt{\frac{Q(up,t)^2+2gA(i,t)^2\Delta z}{1+C_f\Delta l/d}}-C \qquad 3-70$$

The first deviractive of E is

$$E'=1+\frac{\Delta t}{\Delta l}\frac{2g\Delta z}{1+C_f\Delta l/d}\Bigg/\sqrt{\frac{(Q(up,t)/A(i,t))^2+2g\Delta z}{1+C_f\Delta l/d}} \qquad 3-71$$

The object is to find *A(i,t)* that forces *E(A(i,t))* equal to *0*. Using Newton's method with iterations *k* = *1, 2, ...*

$$(A(i,t))_k=(A(i,t))_{k-1}-\frac{(E)_{k-1}}{(E')_{k-1}} \qquad 3-72$$

The covergence criterion for the iterative process is

$$|(E)_k|\le\varepsilon \qquad 3-73$$

where ε is an error criterion.

Boundary condition handling:

Where a cell is located by a ridge, the boundary condition is: $Q(up, t) = 0$. Thus, Equation 3 - 68 becomes:

$$A(i,t) = (A(i,t-\Delta t) + (f_{sm}(i,t) + f_{gm}(i,t))\Delta t) \Big/ (1 + \frac{\Delta t}{\Delta l}\sqrt{\frac{2g\Delta z}{1 + C_f \Delta l / d}}) \qquad 3-74$$

3.7. *Stream Water Model*

A stream network in the OWLS model is represented by a tree data structure, where the "root" of the tree is the stream outlet and the branches are the stream tributaries. The OWLS model applies a 1-D streamflow model to simulate the streamflow routing processes (Figure 3 - 16). The model uses the Kinematic Wave Method, and, similar to the surface flow routing, stream water movement is governed by the St. Venent's equations:

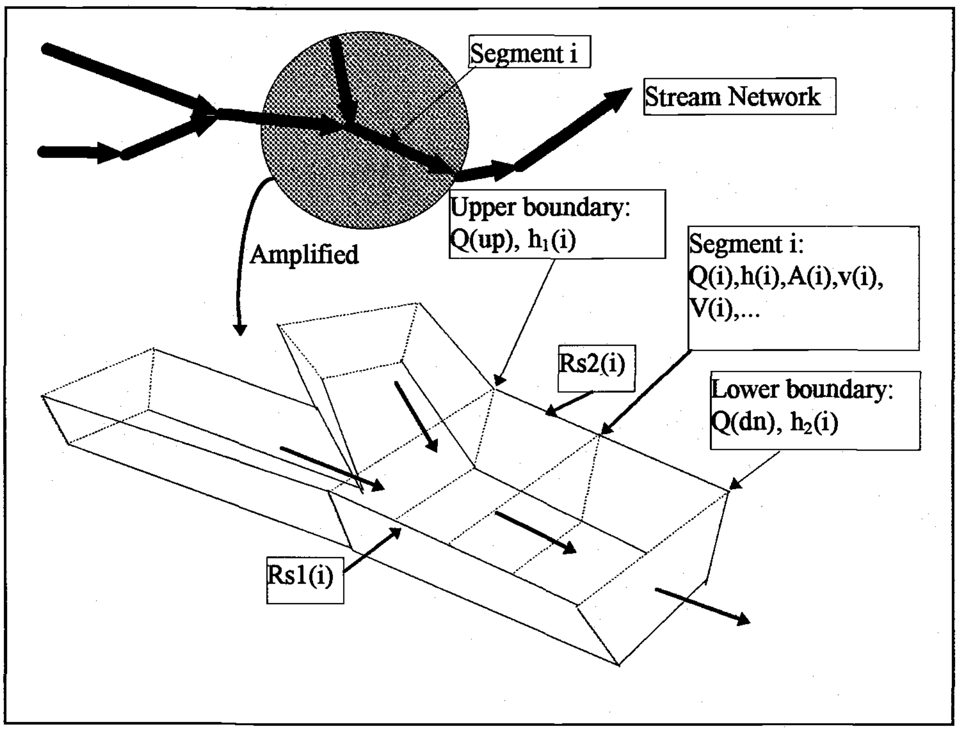


Figure 3 - 16. Stream water model

Continuity Equation: $$\frac{\partial A}{\partial t}+\frac{\partial Q}{\partial l}=I_s \qquad 3-75$$

Momentum Equation: $$S_f=S_0 \qquad 3-76$$

where,

A is the streamflow cross-section area (m^2) as a function of segment water level;

S_0 is the slope (°) of the stream segment.

Q is the stream discharge (m^3/hr), which can be solved by Manning's equation:

$$Q=AS_f^{\frac{1}{2}}R^{\frac{2}{3}}/n=AS_0^{\frac{1}{2}}\left(\frac{A}{P}\right)^{\frac{2}{3}}/n=n^{-1}S_0^{\frac{1}{2}}P^{-\frac{2}{3}}A^{\frac{5}{3}} \qquad 3-77$$

or in the other form of:

$$A=\alpha Q^{\beta} \qquad 3-78$$

where,

$$\alpha=\left(\frac{nP^{\frac{2}{3}}}{\sqrt{S_0}}\right)^{\frac{3}{5}}Q^{\frac{3}{5}} \qquad 3-79$$

$$\beta=\frac{3}{5} \qquad 3-80$$

S_f is the friction slope;

R is the hydraulic radius (m^2/m), which is a function of streamflow water level;

I_s is the regional net incoming flow to the stream segment per unit length (m^3/hr/m) and is determined by:

$$I_s=(P+R_{s1}+R_{s2}-ET_0)/l \qquad 3-81$$

R_{s1} and R_{s2} are the incoming flows (m^3/hr) from the cell on each side of the channel (see Figure 3 - 16).

The finite-difference equation for continuity equation is:

$$A(i,t)-A(i,t-\Delta t)=\frac{I_s(i,t)+I_s(i,t-\Delta t)}{2}\times\Delta t+\frac{\Delta t}{l(i)}\times(Q(up,t)-Q(i,t))$$

3 - 82

Substitute Equation 3 - 78 into 3 - 82 and rearrange, we have a nonlinear difference equation:

$$\alpha Q(i,t)^{\beta} + \frac{\Delta t}{l(i)} Q(i,t) = \alpha Q(i,t-\Delta t)^{\beta} + \frac{I_s(i,t) + I_s(i,t-\Delta t)}{2} \times \Delta t + \frac{\Delta t}{l(i)} \times Q(up,t) \qquad 3 - 83$$

Equation 3 - 83 can be solved using Newton's iterative method as indicated for the surface flow model (Equation 3 - 40 to 43).

Chapter Four: *Applications*

Development of the OWLS model is based on theoretical assumptions and physical laws. To evaluate the model, simulation results were compared to observed hydrologic responses for a specific watershed.

4.1. *Watershed Description*

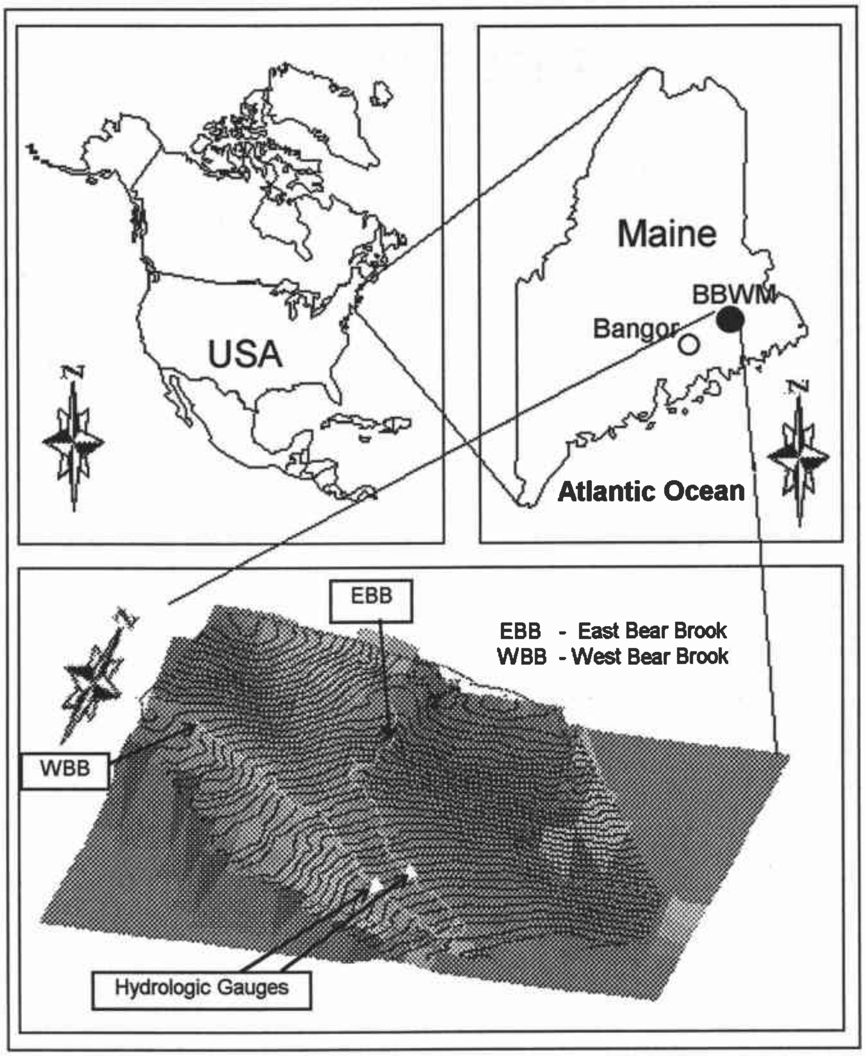


Figure 4 - 1. Bear Brook Watershed of Maine

The Bear Brook Watershed of Maine (BBWM) was selected as a test watershed for evaluating the OWLS model. The BBWM (Figure 4 - 1) is located in eastern Maine (44°52'15" Latitude, 68°06'25" Longitude), approximately 60 kilometers from the Atlantic coastline in the northeastern United States. The BBWM is a paired watershed study funded by U.S.EPA since 1987 as part of The Watershed Manipulation Project (WMP) within the National Acid

Precipitation Assessment Program (NAPAP). NAPAP was designed to assess the causes, effects, and strategies for controlling acidic precipitation.

The major purposes of the BBWM project are to:

(1) Identify and quantify the major processes that control surface water acidity, with a major emphasis on the role of excess sulfate and nitrate and the rate of cation supply through chemical weathering and cation desorption;

(2) Assess the quantitative and qualitative response at the watershed level to different (both increased and decreased) levels of acidic deposition;

(3) Evaluate the ability of existing models of water acidification to predict short- and long-term chemical variations in surgace water chemistry and to predict watershed soil response to increased and decreased loading of strong acids.

As a long-term research watershed, the BBWM includes bench-scale, micro-site, plot, and whole watershed investigations. The associated data bases are ideally suited for watershed hydrologic and chemical simulations at a watershed scale. Thus, it represented an ideal watershed to test the OWLS model.

The study site of the BBWM consists of two first order streams: Eastern Bear Brook (EBB) and West Bear Brook (WBB). On each stream, a catchment outlet was selected and gauged so that both streams have about the same catchment area (EBB=10.7 ha and WBB=10.2 ha). Since both streams are so close and facing the same slope direction, both watersheds are geographically similar and are ideal for a paired watershed study. Streamflow has been monitoring with a standard 120° V-notch weir. Flow data are sampled at 5-minute intervals. Both weirs are anchored on bedrock to ensure that they are stable and impermeable. For the six years of record (1987-1992), EBB has flowed an average of 44 weeks per year; WBB has been perennial. Both watersheds have a maximum discharge of about 0.01 mm/ha/sec or 0.15 m^3/s. Annual water yield relative to incoming precipitation for WBB ranges from 68 to 77% while EBB ranges from 62-68%. From 1987 to 1989, precipitation inputs and resulting discharge were very episodic with flows exceeding $0.09m^3/s$ at least once per year; from 1990 to 1992, discharge has been more moderated and flows rarely exceeding $0.03m^3/s$.

The soils in the two watersheds are thin spodosols developed from till. Soil series have been identified as Dixfield/Marlow in the lower portion of the watersheds, Tunbridge and a Tunbridge/Lyman complex in the middle portions, and a deep Tunbridge/Lyman vairant in the upper portions(Erickson and Wigington, 1987). The bedrock consists predominantly of metamorphosed and deformed pelites, with minor calc-silicate gneiss, and dikes and sills of granite (Norton, et. al., 1992). Folists are common near and at the summit. Minor, poorly-drained soils are present in the upper part of EBB and a small area in discharge region midway up the WBB. Areas supporting softwood stands are characterized by thin mineral soils or folists, whereas hardwoods are mostly present on well-drained, thick, mineral soils common on gentler and lower elevation slopes. The depth of the watershed soils range from 0 to 5 meters, typically 1 to 2 meters. The soils are heterogeneous in composition, containing a large variety of clasts not represented in the local bedrock. Fine-grained fluvial sediments are rare, and consist of pockets of sand seperated by a gravel- and cobble-paved stream bed. Organic debris dams are small and ephemeral.

Vegetation of the BBWM is dominated by hardwoods including American Beech, sugar maple, red maple, with minor amounts of yellow birch and white birch. The hardwood forest is successional following intensive logging prior to about 1945. The upper parts of the watershed have nearly pure softwood stands of red spruce, balsam fir, and hemlock, many of which are more than 100 years old. Softwoods occur dominantly on steeper slopes or where mineral soil is very thin or absent. Softwood, mixed, and hardwood stands cover approximately 25, 40, and 35% of the total watershed areas respectively.

The Climate at BBWM is cool and temperate, with a mild maritime influence. The mean annual temperature is about 4.9°C, with an observed range of +35°C to -30°C. Summer daily maxima temperatures commonly exceed 25°C and winter minima commonly reach -20°C. Precipitation for the period from 1987-1992 at the BBWM has average about 1400 mm per year but locally has ranged from 700 to 1900 mm over the last 10 years.Typically about 20% to 25% of the precipitation is snow. Snow cover may be continuous from late November to April, but more typically the snow pack is completely lost one or more times during winter.Therefore, soil frost may be non-existent or extend to depths approaching 1 meter.

Precipitation Chemistry has been one of the major components in the watershed study (Norton et al, 1995). However, the current version of OWLS is not formulated to simulate the watershed chemistry.

4.2. *Flow Simulation*

4.2.1. Data for the OWLS model

As a physically-based simulation model, detailed information about the watershed is considered highly desireable for running the OWLS model. However, given that detailed information may not always be available, the OWLS model is also designed to allow optional data inputs, for the user to guess-and-try, or even neglect some parameters that are not commonly available. Therefore, data for

OWLS flow simulation include three catagories: (1) Required data, (2) Optional data and (3) System Parameters.

4.2.1.1. *Required data*

Required data for the OWLS model include watershed terrain data (e.g., digital elevation data, land survey data), precipitation data (at least 1 hour in interval), geographical coordinates, air temperature data, soil survey data, and vegetation data. For more information about soil survey data inputs, see the USER'S MANUAL (Appendix III).

4.2.1.2. *Optional data*

Some data for the OWLS model are optional depending upon the application that user chooses. These data include streamflow, soil infiltration, macropore pipe system, channel geometry. The streamflow data are used only when calibration and validation are required. Other optional data can be estimated by the OWLS build-in model. See the USER'S MANUAL (Appendix III) for details.

4.2.1.3. *System Parameters*

There are two type of system parameters for the OWLS model: system control parameters and system model parameters. System control parameters are those used to determine the performance of the model, e.g., English vs SI unit, calculation time step. These parameter are choosen by the user and do not require calibration. System model parameters are those parameters required by the watershed model itself and directly involve the simulation of watershed processes, e.g.,infiltration coefficient, hydraulic conductivity. While many of these parameters have a physical interpretation and a centain range of values, their performance within a watershed model still needs to be determined. Therefore, they usually need to be calibrated.

4.2.1.3.1. System Control Parameters

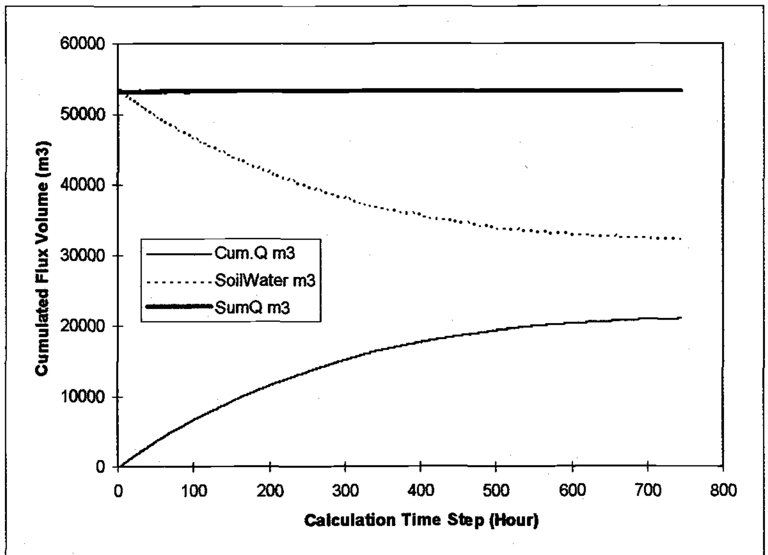
System Control Parameters include unit usage parameters, time domain parameters, output format option, file name definitions and switch parameters. see the USER'S MANUAL (Appendix III) for details.

4.2.3.1.2. System Model Parameters

Parameters that directly drive the hydrological process of the watershed are system model parameters. Some of these parameters are measureable and physically known. But many of them are not physically known, especially at a watershed scale. Thus, they need calibration so that the model can be adjusted in an attempt to represent local watershed. There are about 37 model parameters used in the OWLS hydrologic model for infiltration, soil hydraulic conductivity, snowmelting, evaportranspiration, surface flow routing, soil flow, macropore flow, channel flow routing, and channel geometry. See the USER'S MANUAL (Appendix III) for details.

4.2.2. Parameter Calibration

For each different watershed (especially differences in soil, geological condition), a set of parameters needs to be established so that the OWLS model can simulate hydrologic processes. The procedure for determining parameter values for a particular watershed is called parameter calibration (or parameter optimization). However, before undertaking parameter calibration, a system performance check was implemented to ensure continuity of mass during simulations. The system performance check should not be affected by the choice of parameters since they do not cause water to be "consumed"

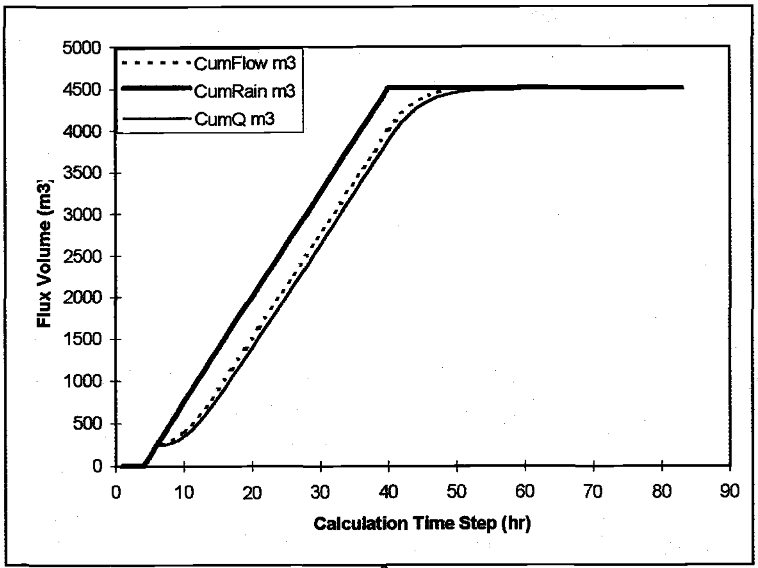


Remark: Basin area = 125137 m^2;
Total Error = 0.0003% per calculation time step;
SumQ = Sum of Cum.Q and SoilWater;
SoilWater = the soil water volume in the watershed;
Cum.Q = the accumulated flow in the outlet.

Figure 4 - 2. Model testing result I -- Soil Flow and Routing under No-Rain Condition for the EBB.

or "produced" within the model; they can only alter the distribution of water between different components in a catchment. The model has been tested and water-balanced under the following conditions:

(1) With high initial soil water content, no flow to macropores and no rainfall, the model was run to test the soil flow system and the overall water balance. Figure 4 - 2 is the testing results from the EBB watershed. In order to check the overall water balance, stream flow has to be accumulated so that summation of the accumulated stream flow and the soil water volume should be constant if the OWLS model produces mass-balance results. As shown in Figure 4 - 2, the model has demonstrated overall balance with a minor error of 0.0003% per calculation step caused from the floating point calculation and iteration;



Remark: Basin area = 125137 m^2;
Rain = 1mm/hr for period from 5 to 40;
Total error = 0.0006% per calculation time step;
CumQ = accumulative flow in the watershed outlet;
CumFlow = accumulative flow of the waters arriving to the stream channel from different parts of cells;
CumRain = The accomulative rainfall;

Figure 4 - 3. Model testing result II -- Surface Flow and Routing under Constant-Rain Condition for the EBB.

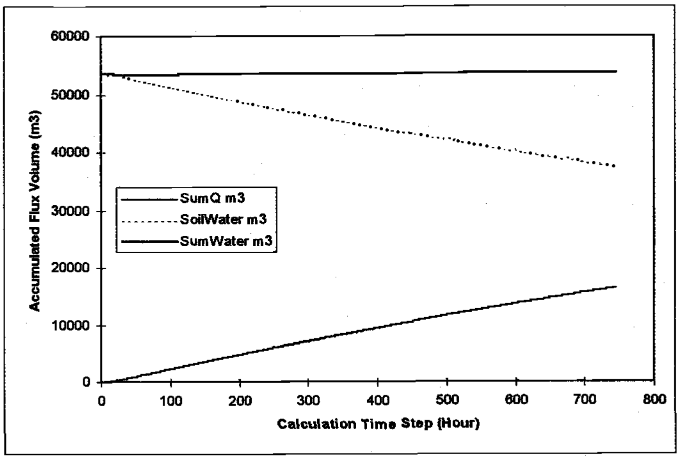
(2) With no infiltration, no soil water and no macropore pipe flow allowed, apply a constant rainfall and run the model to test the surface flow and overall water balance. Figure 4 - 3 shows the testing results from the EBB watershed. Under the above condiction, accumulative flow (either flow into the channel "CumFlow" or flow into the outlet "CumQ") should be the same as the accumulated rainfall in the whole hydrograph process. Figure 4 - 3 shows that during rainfall period (time step 5 to 40), the accumulated flows are smaller than accumulated rainfall. This is because the surface and channell routing delay the flow accumulation. But after the rain stop (time step 40), it takes about 8 to 10 hours for the flow to drain out. After that, accumulated flow are the same as the

accumulated rainfall (an error of 0.00006% per calculation step is caused by the floating point calculation and the iteration error control);

(3) With no soil water and surface flow allowed, apply a constant rainfall and run the model to test the macropore pipe system and overall water balance. Figure 4 - 4 shows the testing results. The summation of soil water volume and the accumulated outlet flow should be constant. The OWLS model produces an minor error of 0.000005% per calculation time step caused by the floating point calculation and the iterations;

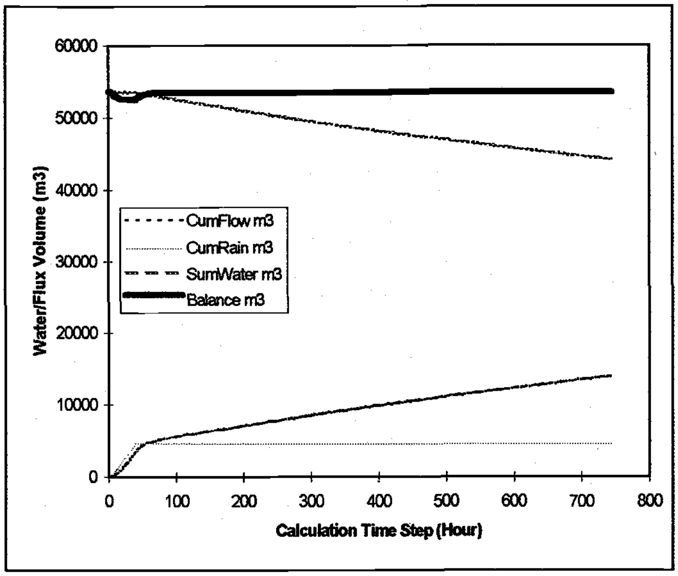
(4) With known initial soil water content, apply a known amount of rainfall for a specified period, run the model to test the water balance of the whole system. Figure 4 - 5 is the testing results, which proves that the OWLS model produces a water balanced results with a minor error of 0.0002% per calculation time step caused by the floating point calculation and iterations.

After testing the model and confirming its capability of



Basin area = 125137 m^2;
Total Error = 0.000005% per calculation time step;
SumQ = the accumulated.flow in the outlet;
SumWater = the total water balance in the watershed;
SoilWater = the soil water volume.

Figure 4 - 4. Model Testing Result III -- Macropore Pipe Flow and Routing under No-Rain Condition for the EBB.



Rainfall = 1mm/hr for period from 5 to 40;
Basin area = 125137 m^2;
Error = 0.0002% per calculation time step
CumFlow = the accumulated flows to the stream channel;
CumRain = the accumulated rainfall;
SumWater = the total water storage
Balance is the water volume after removal of rainfall effect (= SumWater + CumFlow - CumRain), which should approximate to the initial water storage.

Figure 4 - 5. Model Testing Result IV -- All Flow and Routing under Rain Condition for the EBB.

satisfying the above conditions, calibration was undertaken. Calibrating parameters is a time consuming process since there are so many of them and there are many methods of calibration. One may use an optimizing program to evaluate the results of simulation and adjust parameters accordingly. But as the numbers of parameters increase, the options for multiple parameter adjustments increase dramatically. Automatic calibration may introduce unexpected results. Therefore, as with many other complex models, parameter calibration involved multiple trial-and-error run; professional judgement being used to decide which parameters to adjust and to what extent. The strategy for determining the model parameters was threefold:

(1) Pick initial values for parameters as rational as possible by doing some simply math calculations;

(2) Pick a low-flow period that is followed by a peakflow at the beginning of a simulation and determine the initial type parameter first (like initMoisture);

(3) Test run the model and evaluate results as follows:

a. timing of peakflow: adjust the watershed surface roughness and channel Manning's roughness to change the peak timing;

b. rising limb of hydrograph: for faster rising, allow more surface flow or macropore pipe flow and vice versa;

c. falling limb of hydrograph: for slower recession, slow the macropore pipe flow by increasing its friction coefficient of the macropore system, and increase infiltration. To rise the base flow, increase the soil water supply to the macropore system or the soil conductivity.

d. peak height: for higher peak, increase the surface water portion and/or macropore flow by reducing the infiltration rate, reduce the water lost via tree interception or ET, or reduce infiltration.

e. base flow: for persistant base flow, reduce the soil hydraulic conductivity when the recession limb is too steep or increase the initial soil moiture condition when the base flow curve falls belows the observed one.

Other parameters can also be adjusted, such as the Horton's parameters, macropore pipe radius and parameters, channel geometric parameters, ET, and others. Some of these parameters may have little effect on short-term hydrological processes and thus may not need to be adjusted. Table 4 - 1 lists the eight most sensitive parameters to the simulated hydrograph. These parameters are most offen adjusted in model calibration.

In the BBWM watershed, we chose the Eastern Bear Brook (EBB) and two time periods for parameter calibratizon. The time periods were early summer (May 1 to June 1, 1989) and late fall (Oct. 15 to Nov. 28, 1989).

Table 4 - 1. Most Sensitive Parameters in the OWLS model

Parameter	**Description**	**Data Range**
infiltration0Adjust	Dimensionless, adjust factor for maximun infiltration rate. For distributed model, different soils have different lab-tested maximun infiltration rates, the "infiltration0Adjust" parameter is a ratio to adjust all the soil maximun infiltration rates simultaneously to approximate the field condition. This parameter has great effect the proportion of water distribution amoung surface, soil and macropore system.	0.1 ~ 10.
infiltrationCAdjust	Dimensionless, adjust factor for maximun infiltration rate. This parameter has great effect the proportion of water distribution amoung surface, soil and macropore system.	0.1 ~ 10.
surfaceMacroporeConst	Dimensionless, adjust factor for amount of surface water directly drain into the soil macropore system. It is a proportion factor to the amount of soil infiltration. It has great effect to the peak flow and basically control the flow subdivision between surface flow and macropore pipe flow.	1 ~ 5.
soilMacroporeConst	Dimensionless, adjust factor for amount of soil water directly drain into the macropore system. It is a proportion factor to the amount of soil water depth, relative soil moisture content. It has great effect to the base flow and basically control the base flow subdivision between soil matrix flow and macropore pipe flow.	$0 \sim 10^{-3}$
conductivityAdjust	Dimensionless, adjust factor for soil conductivity rate. For distributed model, different soils have different lab-tested unsaturated conductivities, the "conductivityAdjust" parameter is a ratio to simultaneously adjust the hydraulic conductivities for all the soils to approximate the field condition. This parameter has great effect on the soil base flow.	1 ~ 1000
frictionCoeff	Dimensionless, the friction coeffience for macropore pipe system. It has great effect on the falling-limb of a peak flow hydrograph.	1 ~ 100
roughness	Dimensionless, the Manning's roughness for watershed surface. It has great effect on the peak timing and smoothness of the hydrograph.	0.1 ~ 1.0
Manning	Dimensionless, the Manning's roughness for channel segments. It has large effect on the peak timing and smoothness of the hydrograph.	0.05 ~ 0.5
snowMelt_Df	m/°C/hr, the degree-day snowmelt factor. It has great effect on the snowmelt-caused flow event, both peak and event period.	$10^{-4} \sim 10^{-2}$

Figure 4 - 6 illustrates the calibration results from the early summer of 1989. The top text section lists the major parameters and the values that were used during this simulation. It shows the node index of the watershed outlet, which is 233 for the EBB, and the number of nodes(94), edges(210), and cells(117) for the simulation area and 474 nodes, 743 edges and 464 cells for the study area. The area of the simulation watershed is 125,100m^2. The calculation time interval for this simulation was 1 hour. The interval for saving results is 1 step, which is equal to 1 hour here. A total 745 hours have been simulated. Parameter values are also listed.

The top and bottom banner of Figure 4 - 6 illustrates the time scale for a simulation. Each cross-over line represents one day (which can be redefined by an user). All curves shown in the chart are plotted for every calculation step, or 1 hour.

From top to bottom, the first chart in Figure 4 - 6 is the air temperature, showing the hourly temperature fluctuation during the simulation period. The unit is °C and the center line is 0 °C.

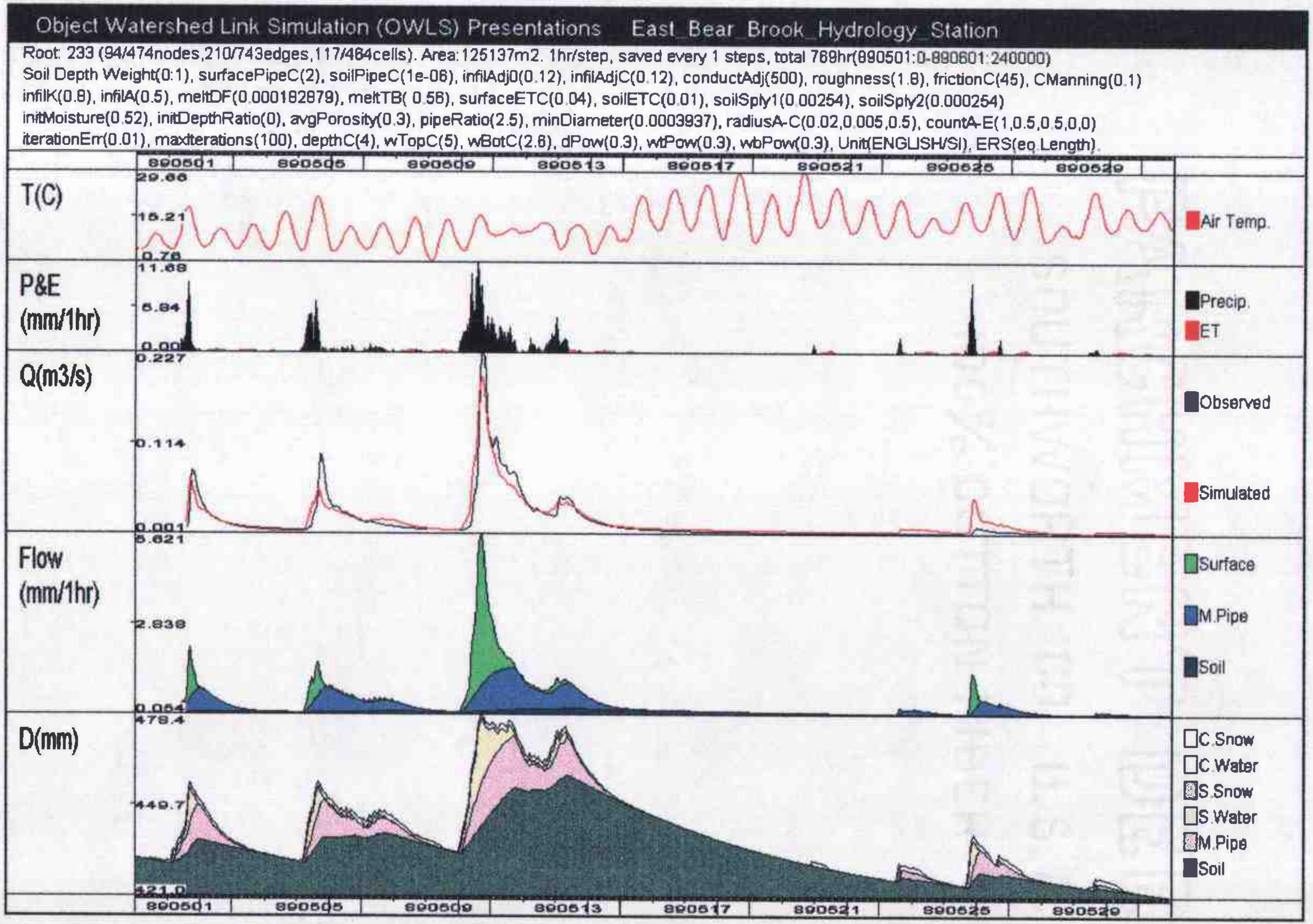
The second chart is the precipitation and simulated evaportranspiration (mm).

The third chart is the hydrograph for both observed and simulated flow at the stream outlet (m^3/s). There are four storm events in this month. Simulated flow peaks generally match except for the smallest event, which is over-predicted. The simulated rising and falling limb have the similar slope as the observed flow for three largest events. The baseflow is also well-estimated.

The fourth chart is the flow composition. It is represented in flow depth calculated from the division of the summation of different flows arriving to the stream channel for that interval of time by the watershed drainage area. There are three sources of water flow into the stream channel segments: (1) overland flow from the surface of riparian cells, (2) macropore flow from soil macropore system of the riparian cells and (3) soil matrix flow from the soil of the riparian cells. This chart indicates that the major contribution to peakflow is from surface flow and macropore pipe flow; soil flow had little effect. The higher proportion of contributions from surface flow in the Bear Brook is the result of shallow soil and rocky terrain in the riparian area.

The fifth chart represents the water depth for different layers in the watershed. In this example, it includes water associated with the canopy, surface, soil and macropores. For the canopy and soil surface, water has two phases: liquid water and snow. This chart indicates that the majority water in the watershed is stored in the soil; macropore and surface water can only provide temporary storage. Surface water can exist only for a very short time, while macropore water can be detained somewhat longer. Water on the canopy will be evaporated soon after a rainfall event.

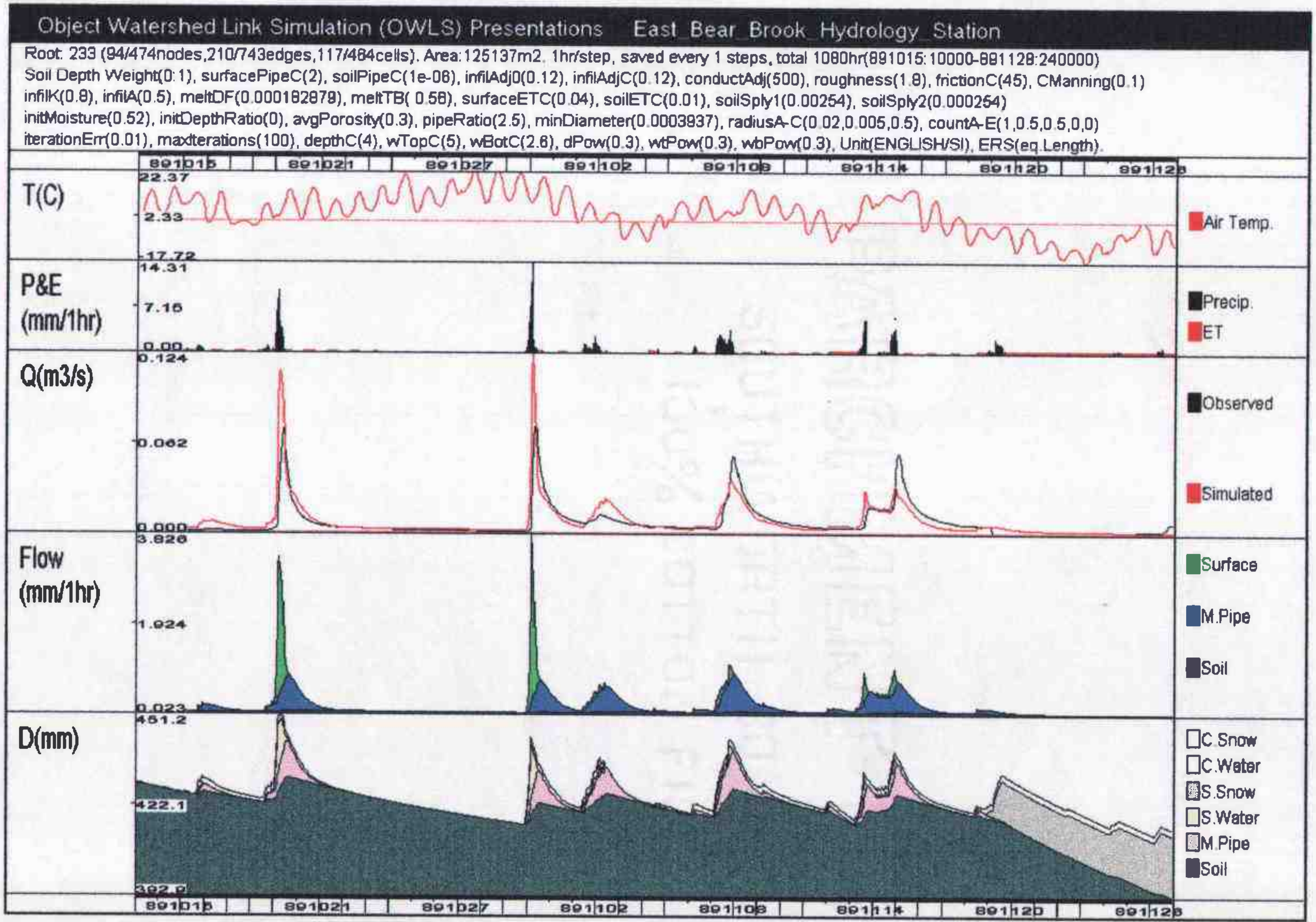
Figure 4 - 7 illustrates the calibration results for the late fall of 1989. Since the air temperatures are close to 0 degrees C, it is difficult to correctly simulate rain or snow from the



Object Watershed Link Simulation (OWLS) Presentations East_Bear_Brook_Hydrology_Station

Root: 233 (94/474nodes,210/743edges,117/464cells). Area:125137m2. 1hr/step, saved every 1 steps, total 769hr(890501:0-890601:240000)
Soil Depth Weight(0:1), surfacePipeC(2), soilPipeC(1e-06), infilAdj0(0.12), infilAdjC(0.12), conductAdj(500), roughness(1.8), frictionC(45), CManning(0.1)
infilK(0.8), infilA(0.5), meltDF(0.000182879), meltTB(0.56), surfaceETC(0.04), soilETC(0.01), soilSply1(0.00254), soilSply2(0.000254)
initMoisture(0.52), initDepthRatio(0), avgPorosity(0.3), pipeRatio(2.5), minDiameter(0.0003937), radiusA-C(0.02,0.005,0.5), countA-E(1,0.5,0.5,0,0)
iterationErr(0.01), maxIterations(100), depthC(4), wTopC(5), wBotC(2.8), dPow(0.3), wtPow(0.3), wbPow(0.3), Unit(ENGLISH/SI), ERS(eq Length).

Figure 4 - 6. Calibration results from the EBB watershed, 5/1/89 - 6/1/89



precipitation records. Thus, in the hydrograph, some hydrograph peaks are over-predicted and some are under-estimated. Even so, the occurances, runoff volumes and timing of simulated hydrographs generally match the observed hydrographs.

Following the completion of the calibration, a set of parameters for particular watershed were available for simulation. Values of the calibrated parameters for the East Bear Brook watershed are listed in Table 4 - 2.

Table 4 - 2. Calibrated parameters for BBWM from the East Bear Brook

No.	**Parameters**	**Value**	**Unit(English)**
		Canopy Parameters	
1	canopyMinETRate	0.01	inch/hr
		Surface Parameters	
2	ET_a	0.025	dimensionless
3	ET_b	0.078	dimensionless
4	roughness	1.8	dimensionless
5	snowMelt_Df	0.004	in/hr/dF
6	snowMelt_Tb	33	dF
7	underCanopyETConstant	0.04	dimensionless
		Soil Parameters	
8	conductivityAdjust	500	dimensionless
9	infiltration_a	0.5	dimensionless
10	infiltration_k	0.8	1/hr
11	infiltration0Adjust	0.12	dimensionless
12	infiltrationCAdjust	0.12	dimensionless
13	layerWeight1	0	dimensionless
14	layerWeight2	1	dimensionless
15	soilETConstant	0.01	dimensionless
16	soilMoisture	0.52	dimensionless
17	soilWaterSupplyC1	0.1	inch/hr
18	soilWaterSupplyC2	0.01	inch/hr
		Macropore Parameters	
19	countA	1	dimensionless
20	countB	0.5	dimensionless
21	countC	0.5	dimensionless
22	countD	0	dimensionless
23	countE	0	dimensionless
24	frictionCoeff	45	dimensionless
25	minDiameter	0.000394	inch
26	pipeRatio	2.5	dimensionless
27	radiusA	0.02	inch

Table 4 - 2. Calibrated parameters for BBWM from the East Bear Brook (Continued)

No.	Parameters	Value	Unit(English)
28	radiusB	0.005	inch
29	radiusC	0.5	dimensionless
30	soilMacroporeConst	0.000001	dimensionless
31	surfaceMacroporeConst	2.0	dimensionless
	Stream Parameters		
32	depthConstant	4	dimensionless
33	depthPow	0.3	dimensionless
34	initialStreamDepthRatio	0	dimensionless
35	Manning	0.1	dimensionless
36	widthBotConstant	2.6	dimensionless
37	widthBotPow	0.3	dimensionless
38	widthTopConstant	5	dimensionless
39	widthTopPow	0.3	dimensionless
	General Parameters		
40	avgTempDifference	7	hour
41	iterativeErr	0.01	dimensionless
42	maxIterations	100	dimensionless
43	maxTempTime	14	hour
44	snowSeasonBeginMMDD	1101	mmdd
45	snowSeasonEndMMDD	531	mmdd
46	turbidity	0.8	dimensionless

* Detailed explaination, please see the USER'S MANUAL (Appendix III)

4.2.3. Model Validation

Model validation is a process of varifying the correctness of model parameters for a watershed. The traditional method of accomplishing this is:

A. Pick a data series for a period of time which has not been used by the model for parameter calibration;

B. Simulate the streamflow for the precipitation event of that period;

C. Compare the results of simulation to the observed ones;

D. If the results are within a specific error range, then the model is validated for that watershed.

Since BBWM is comprised of paired watersheds, besides using the traditional model validation procedure to the EBB watershed, the model was also applied to the WBB watershed to see

Table 4 - 3. Simulated events in BBWM for the calibration and validation of the OWLS model

Basin	**Period for Events**	**Figure Name**	**Remark**
East Bear Brook Watershed	5/1/89 - 6/1/89	Figure 4 - 6	For Calibration
	10/15/89 - 11/28/89	Figure 4 - 7	For Calibration
	3/27/90 - 4/25/90	Figure 4 - 8	For Validation
	2/26/91 - 3/29/91	Figure 4 - 9	For Validation
West Bear Brook Watershed	5/1/89 - 6/1/89	Figure 4 - 10	For Validation
	10/15/89 - 11/28/89	Figure 4 - 11	For Validation
	3/27/90 - 4/25/90	Figure 4 - 12	For Validation
	2/26/91 - 3/29/91	Figure 4 - 13	For Validation

if the model could satisfactorly simulate its flows. Table 4 - 3 lists the events that were used for the validation as well as the calibration of the OWLS model.

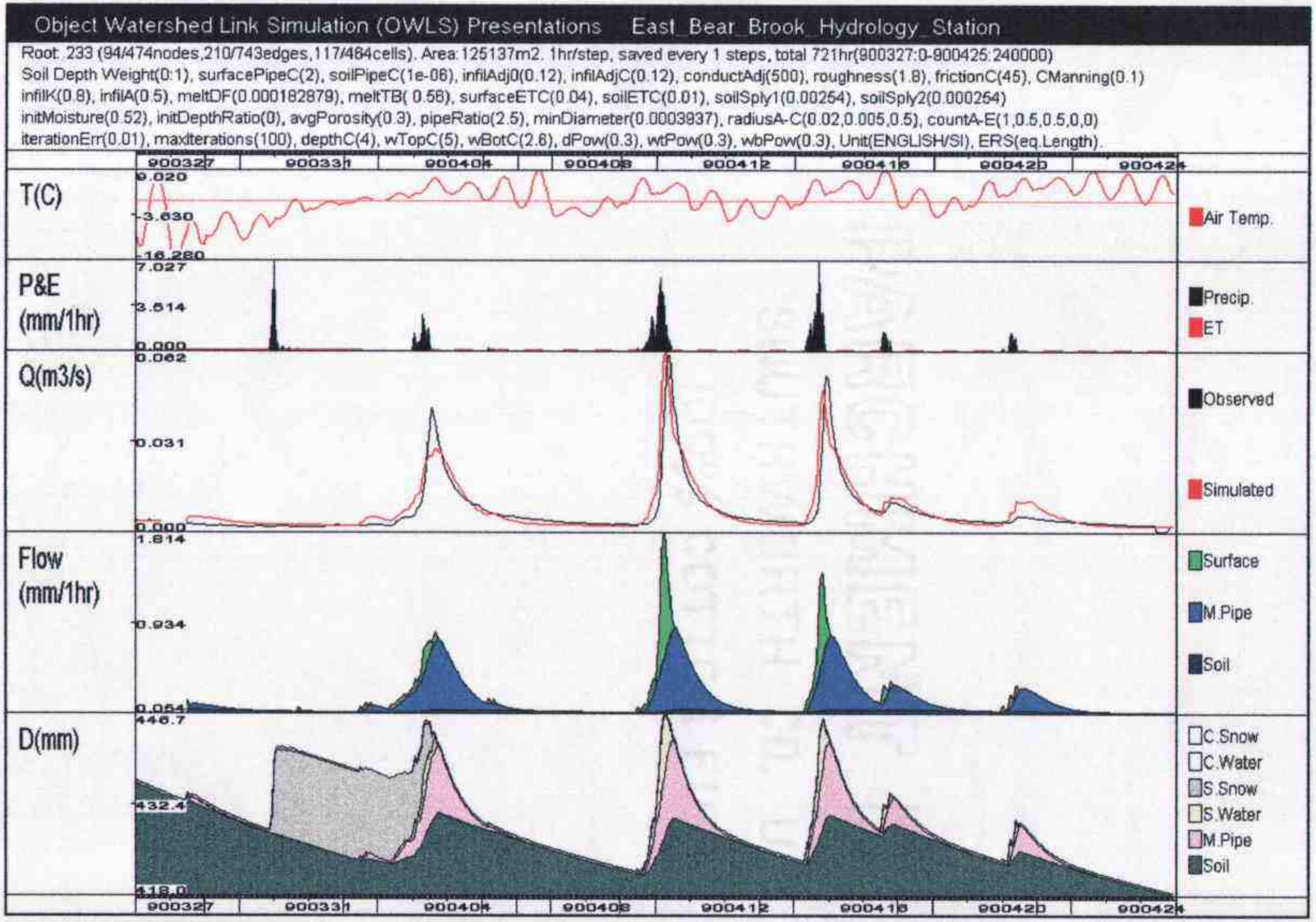
Figure 4 - 8 is the result of validation during the period from March 27 to April 25, 1990 in the EBB watershed. Three major runoff events in this period; snow fall occurred early in the simulation. For the all these major runoff events, the OWLS model provides good estimations in both flood volume and peak timing.

Figure 4 - 9 illustrates the results of validation for winter conditions in the EBB watershed (Febuary 26 to March 29, 1991). Within this period, there are five flood events with rainfall and/or snowfall; The OWLS model has good simulation for the highest peak and its volume and fine results for the volumes and timing-shifting for the other events.

Figure 4 - 10 is the result of validation for the period May 1 to June 1 of 1989 in the WBB watershed. This is the time period used in parameter caliberation in the EBB. The simulation in the WBB shows under-estimations for all the peak flows, but timing and base flow are good-estimated for all the flood events.

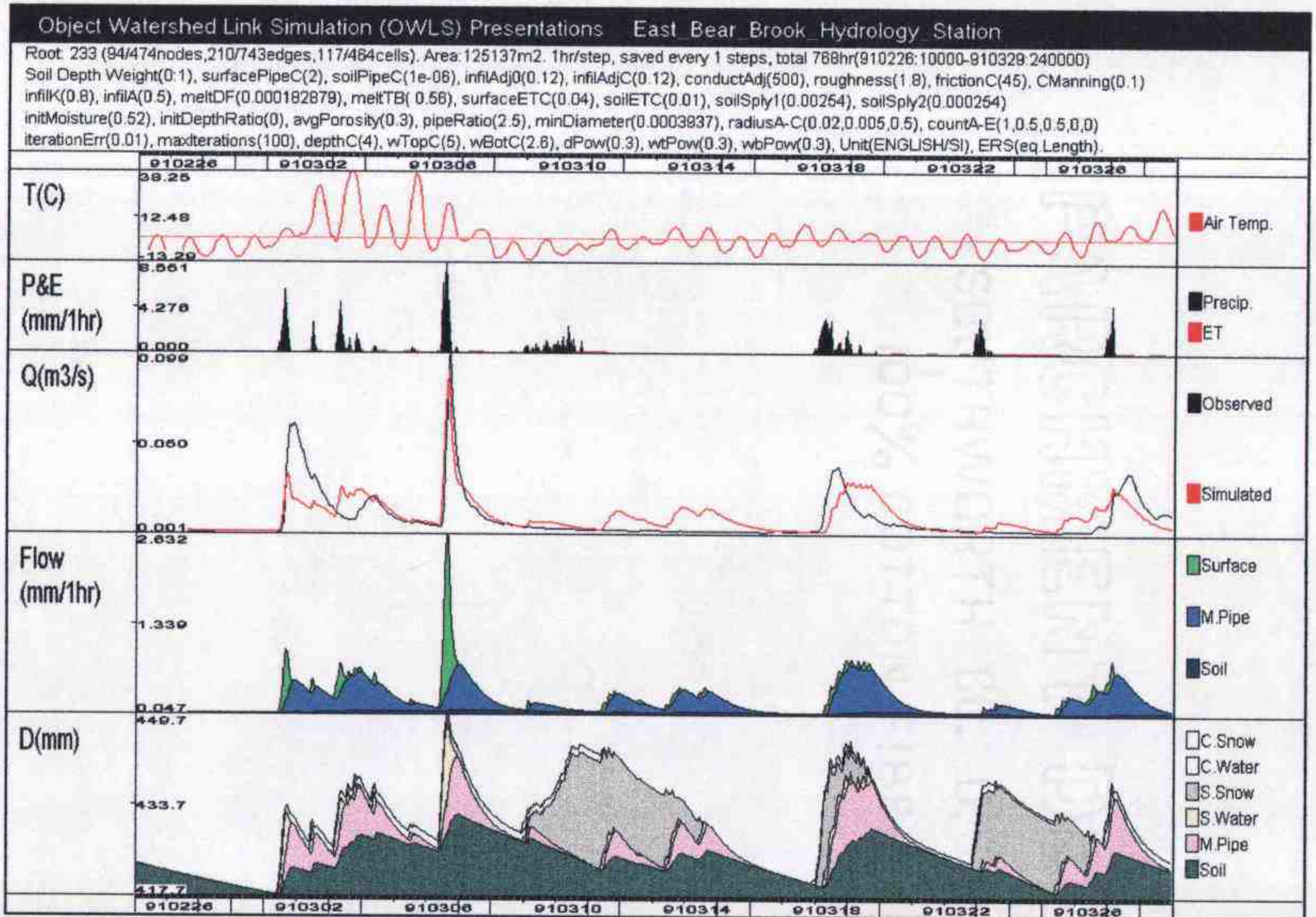
Figure 4 - 11 is the result of validation for the period of October 15 to November 28 of 1989 in the WBB watershed. The OWLS model tends to over-estimate the peak flows for the first two large events and produce good estimations for other smaller events.

Figure 4 - 12 is the result of validation for the period of March 27 to April 25 of 1990 in the WBB watershed. The OWLS model under-estimates the first event and produces good estimations for all other flood events.



Root: 233 (94/474nodes,210/743edges,117/484cells). Area:125137m2. 1hr/step, saved every 1 steps, total 721hr(900327:0-900425:240000)
Soil Depth Weight(0:1), surfacePipeC(2), soilPipeC(1e-06), infilAdj0(0.12), infilAdjC(0.12), conductAdj(500), roughness(1.8), frictionC(45), CManning(0.1)
infilK(0.8), infilA(0.5), meltDF(0.000182879), meltTB(0.56), surfaceETC(0.04), soilETC(0.01), soilSply1(0.00254), soilSply2(0.000254)
initMoisture(0.52), initDepthRatio(0), avgPorosity(0.3), pipeRatio(2.5), minDiameter(0.0003937), radiusA-C(0.02,0.005,0.5), countA-E(1,0.5,0.5,0,0)
iterationErr(0.01), maxIterations(100), depthC(4), wTopC(5), wBotC(2.6), dPow(0.3), wtPow(0.3), wbPow(0.3), Unit(ENGLISH/SI), ERS(eq.Length).

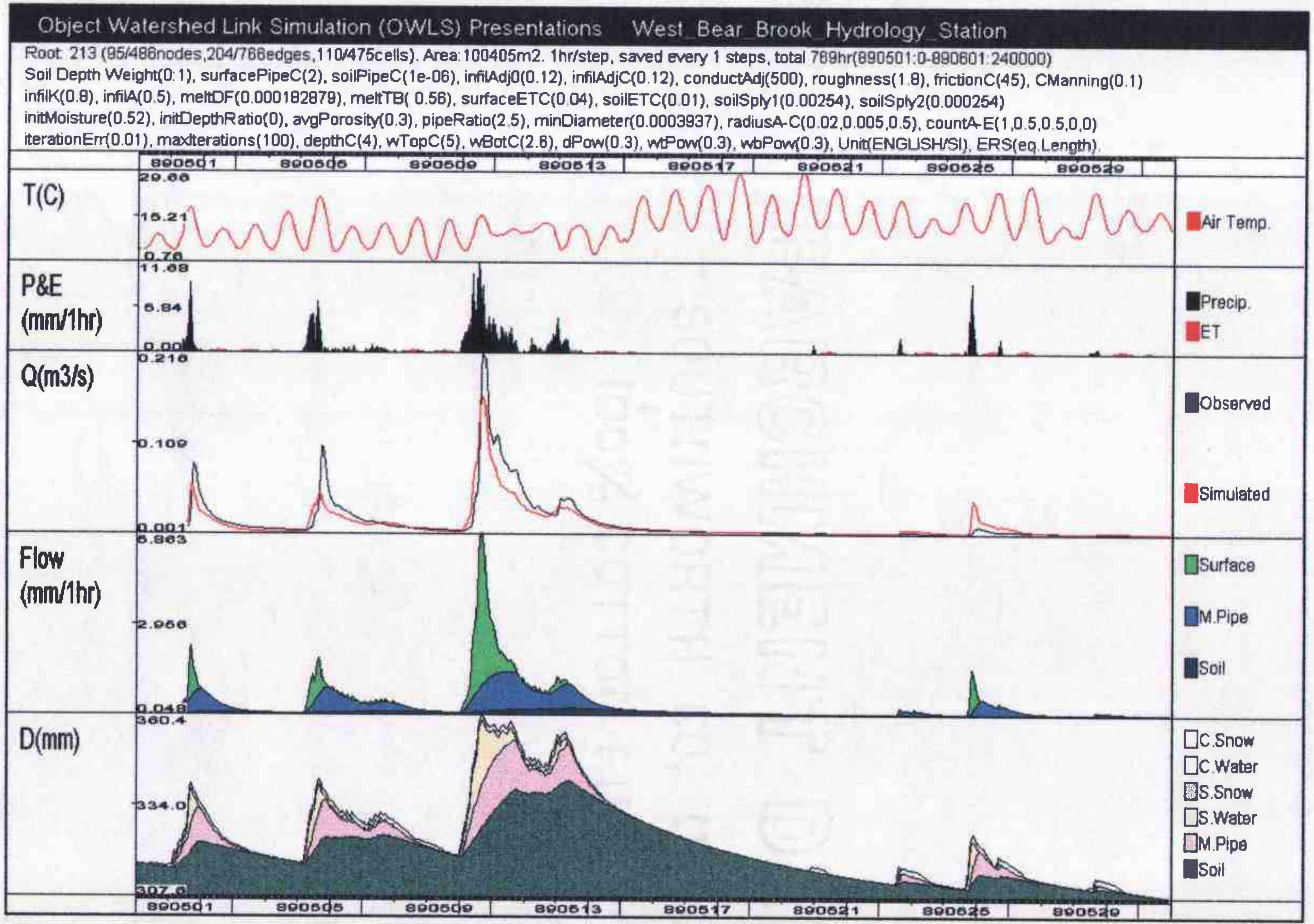
Figure 4 - 8. Validation results from the EBB watershed, 3/27/90 - 4/25/90.



Object Watershed Link Simulation (OWLS) Presentations East_Bear_Brook_Hydrology_Station

Root: 233 (94/474nodes,210/743edges,117/464cells). Area:125137m2. 1hr/step, saved every 1 steps, total 768hr(910226:10000-910329:240000)
Soil Depth Weight(0:1), surfacePipeC(2), soilPipeC(1e-06), infilAdj0(0.12), infilAdjC(0.12), conductAdj(500), roughness(1.8), frictionC(45), CManning(0.1)
infilK(0.8), infilA(0.5), meltDF(0.000182879), meltTB(0.56), surfaceETC(0.04), soilETC(0.01), soilSply1(0.00254), soilSply2(0.000254)
initMoisture(0.52), initDepthRatio(0), avgPorosity(0.3), pipeRatio(2.5), minDiameter(0.0003937), radiusA-C(0.02,0.005,0.5), countA-E(1,0.5,0.5,0,0)
iterationErr(0.01), maxIterations(100), depthC(4), wTopC(5), wBotC(2.6), dPow(0.3), wtPow(0.3), wbPow(0.3), Unit(ENGLISH/SI), ERS(eq.Length).

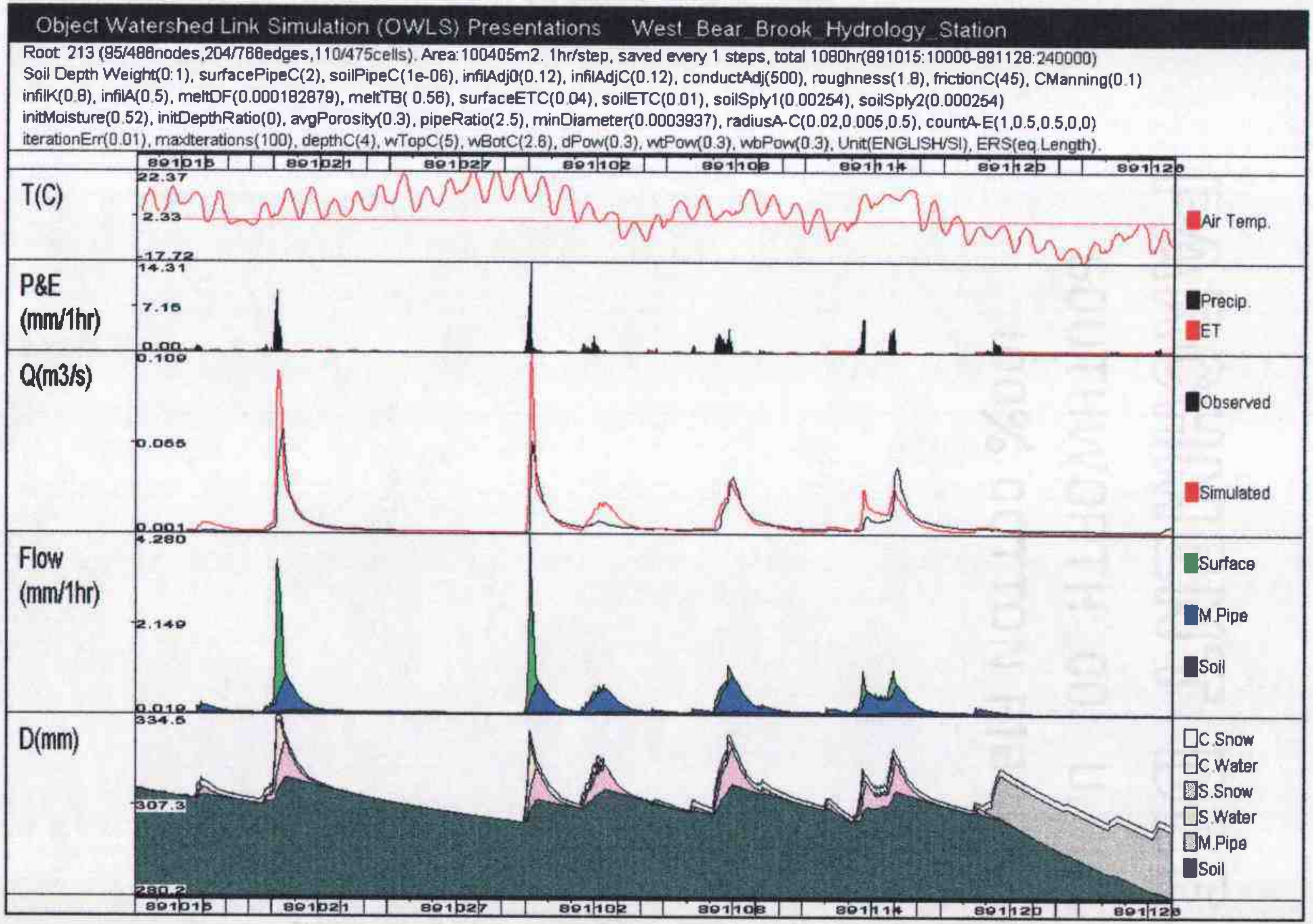
Figure 4 - 9. Validation results from the EBB watershed, 2/26/91 - 3/29/91.



Object Watershed Link Simulation (OWLS) Presentations West_Bear_Brook_Hydrology_Station

Root: 213 (95/486nodes,204/766edges,110/475cells). Area:100405m2. 1hr/step, saved every 1 steps, total 769hr(890501:0-890601:240000)
Soil Depth Weight(0:1), surfacePipeC(2), soilPipeC(1e-06), infilAdj0(0.12), infilAdjC(0.12), conductAdj(500), roughness(1.8), frictionC(45), CManning(0.1)
infilK(0.8), infilA(0.5), meltDF(0.000182879), meltTB(0.56), surfaceETC(0.04), soilETC(0.01), soilSply1(0.00254), soilSply2(0.000254)
initMoisture(0.52), initDepthRatio(0), avgPorosity(0.3), pipeRatio(2.5), minDiameter(0.0003937), radiusA-C(0.02,0.005,0.5), countA-E(1,0.5,0.5,0,0)
iterationErr(0.01), maxIterations(100), depthC(4), wTopC(5), wBotC(2.6), dPow(0.3), wtPow(0.3), wbPow(0.3), Unit(ENGLISH/SI), ERS(eq.Length).

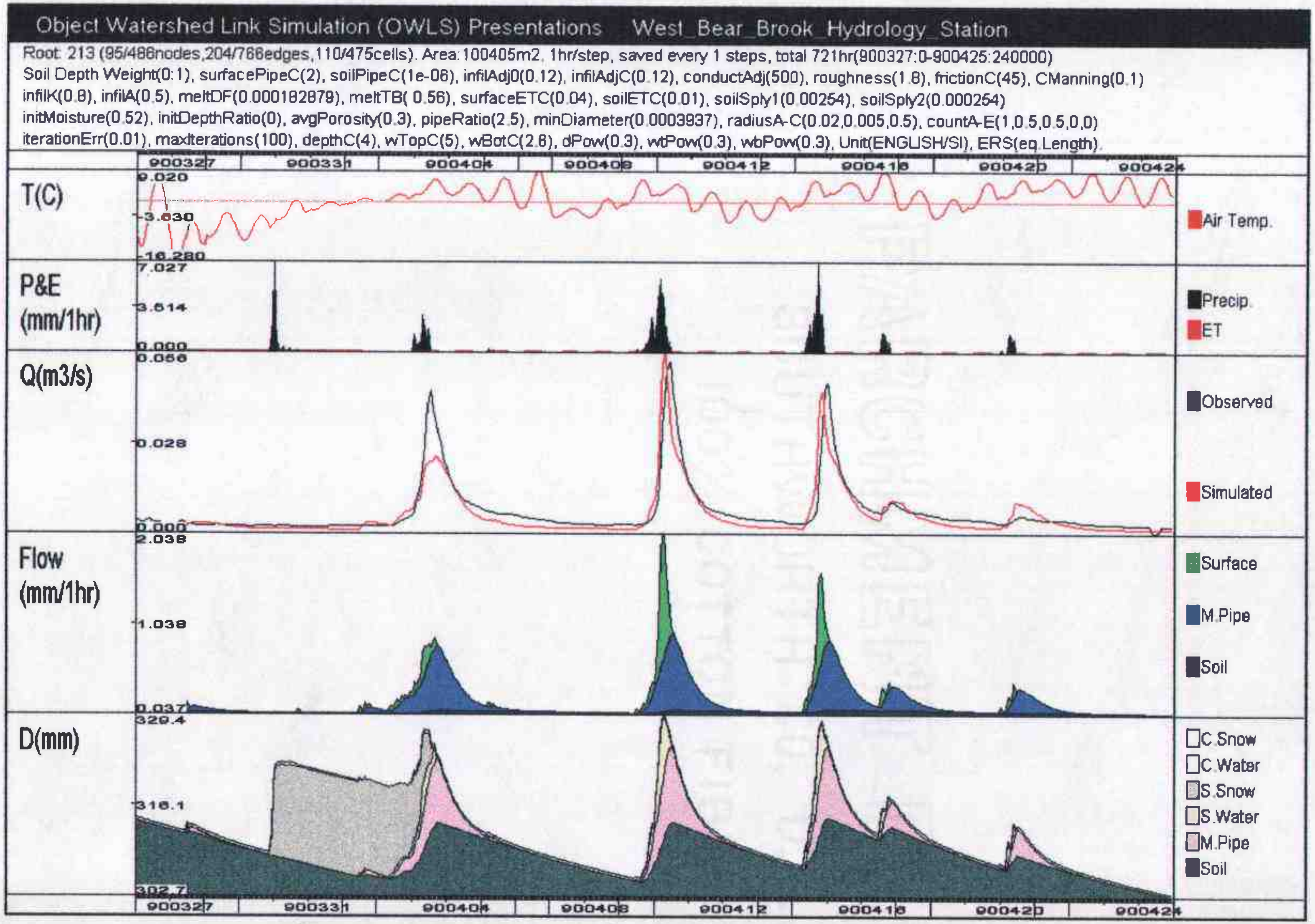
Figure 4 - 10. Validation results from the WBB watershed, 5/1/89 - 6/1/89



Object Watershed Link Simulation (OWLS) Presentations West_Bear_Brook_Hydrology_Station

Root 213 (95/488nodes,204/788edges,110/475cells). Area:100405m2. 1hr/step, saved every 1 steps, total 1080hr(891015:10000-891128:240000)
Soil Depth Weight(0:1), surfacePipeC(2), soilPipeC(1e-06), infilAdj0(0.12), infilAdjC(0.12), conductAdj(500), roughness(1.8), frictionC(45), CManning(0.1)
infilK(0.8), infilA(0.5), meltDF(0.000182879), meltTB(0.56), surfaceETC(0.04), soilETC(0.01), soilSply1(0.00254), soilSply2(0.000254)
initMoisture(0.52), initDepthRatio(0), avgPorosity(0.3), pipeRatio(2.5), minDiameter(0.0003937), radiusA-C(0.02,0.005,0.5), countA-E(1,0.5,0.5,0,0)
iterationErr(0.01), maxIterations(100), depthC(4), wTopC(5), wBotC(2.6), dPow(0.3), wtPow(0.3), wbPow(0.3), Unit(ENGLISH/SI), ERS(eq.Length).

Figure 4 - 11. Validation results from the WBB watershed, 10/15/89 - 11/28/89



Object Watershed Link Simulation (OWLS) Presentations West_Bear_Brook_Hydrology_Station

Root: 213 (95/486nodes,204/766edges,110/475cells). Area:100405m2. 1hr/step, saved every 1 steps, total 721hr(900327:0-900425:240000)
Soil Depth Weight(0:1), surfacePipeC(2), soilPipeC(1e-06), infilAdj0(0.12), infilAdjC(0.12), conductAdj(500), roughness(1.8), frictionC(45), CManning(0.1)
infilK(0.8), infilA(0.5), meltDF(0.000182879), meltTB(0.56), surfaceETC(0.04), soilETC(0.01), soilSply1(0.00254), soilSply2(0.000254)
initMoisture(0.52), initDepthRatio(0), avgPorosity(0.3), pipeRatio(2.5), minDiameter(0.0003937), radiusA-C(0.02,0.005,0.5), countA-E(1,0.5,0.5,0,0)
iterationErr(0.01), maxIterations(100), depthC(4), wTopC(5), wBotC(2.6), dPow(0.3), wtPow(0.3), wbPow(0.3), Unit(ENGLISH/SI), ERS(eq.Length).

Figure 4 - 12. Validation results from the WBB watershed, 3/27/90 - 4/25/90.

Figure 4 - 13 is the result of validation for the period of Feb. 26 to March 29, 1991 in the WBB watershed. Precipitation is rain or snow. In this period, there are totally five events, the OWLS model under-estimated the first event. It is possible that the first event is a rain on snow event but the initial condition of the model did not reflect it. There are four other flood events. The OWLS model produced good estimation for the highest peak event and fine estimation in volumes and timing-shifting for the other events.

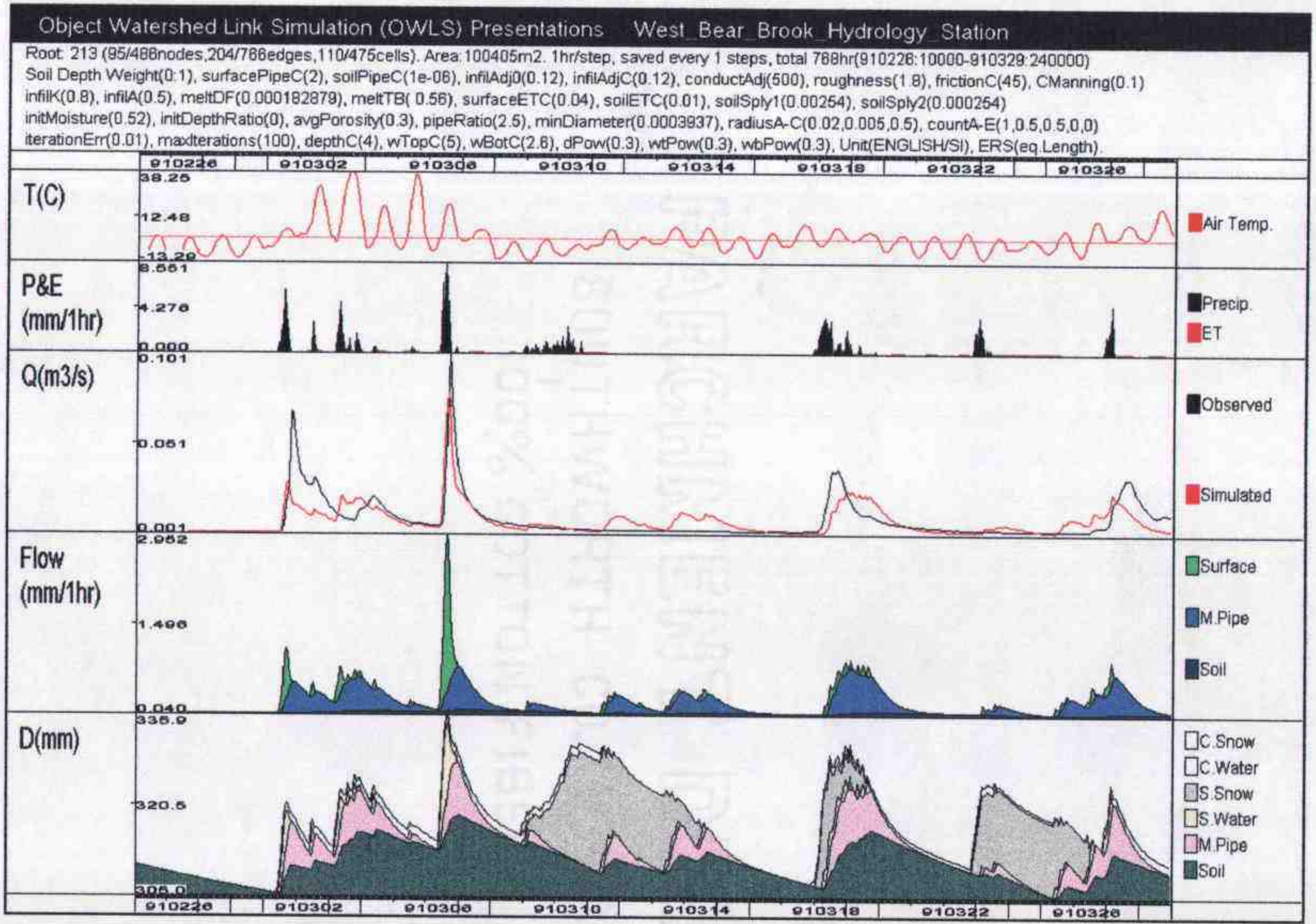
4.3. *Conclusion of the Hydrologic Simulation*

As indicated by the results of the previous simulations, the OWLS hydrologic model appears to provide good flow estimations for rain-based events. However, the model could not provide good runoff estimations when air temperature fluctuated around 0°C and when high air temperature occurred during snowmelt. Other factors may also cause errors in the simulations. One of these might be the simplication of the model (under simple mode) whereby precipitation, air temperature, vegetation and soil characteristics are considered the same for all cells. We might expect better and more realistic simulation results if the following information was available and the model simulated the watershed under the complex mode:

(1) detailed physical information about different types of vegetation;
(2) detailed physical information about different types of soils;
(3) data from more then three meteorological stations;
(4) precipitation data which includes information about rain or snow;
(5) data for actual cloud cover;
(6) detailed air temperature data.

Some parameters of the OWLS model are still unknown and can only be guessed, like the parameters for macropores pipe system. Supplimentary information from future field survey and lab studies represents an important need for the continuing development of the OWLS model.

While the OWLS model produces lots of information about watershed's hydrologic processes, it also generates lots of questions. For details about usage of the model, please see the USER'S MANUAL (Appendix III).



Object Watershed Link Simulation (OWLS) Presentations West_Bear_Brook_Hydrology_Station

Root: 213 (95/488nodes,204/788edges,110/475cells). Area:100405m2. 1hr/step, saved every 1 steps, total 788hr(910226:10000-910329:240000)
Soil Depth Weight(0:1), surfacePipeC(2), soilPipeC(1e-06), infilAdj0(0.12), infilAdjC(0.12), conductAdj(500), roughness(1.8), frictionC(45), CManning(0.1)
infilK(0.8), infilA(0.5), meltDF(0.000182879), meltTB(0.56), surfaceETC(0.04), soilETC(0.01), soilSply1(0.00254), soilSply2(0.000254)
initMoisture(0.52), initDepthRatio(0), avgPorosity(0.3), pipeRatio(2.5), minDiameter(0.0003937), radiusA-C(0.02,0.005,0.5), countA-E(1,0.5,0.5,0,0)
iterationErr(0.01), maxIterations(100), depthC(4), wTopC(5), wBotC(2.8), dPow(0.3), wtPow(0.3), wbPow(0.3), Unit(ENGLISH/SI), ERS(eq.Length).

Figure 4 - 13. Validation results from the WBB watershed, 2/26/91 - 3/29/91.

Chapter Five: *Visualization of Hydrological Processes*

The OWLS model includes the 3-D graphical module to visualize the watershed topography, soil characteristics and simulated flow paths, stream channel and hydrologic components.

5.1. *Basic Theory*

In order to three-dimensionally visualize a watershed, the watershed has to be subdivided into many smaller pieces (cells) with associated nodes and edges. The three-dimensional coordinates used by the OWLS model are shown in Figure 5 - 1.

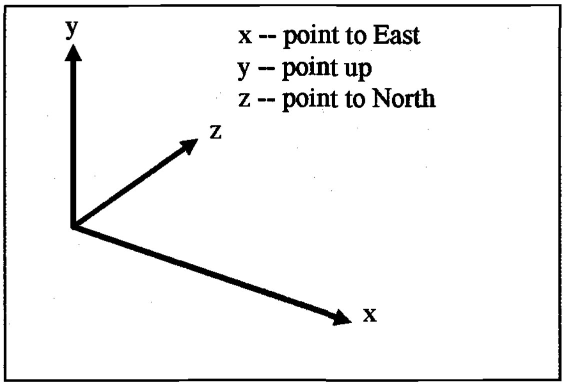


Figure 5 - 1. Coordinate system for the OWLS.

Unlike traditional 3-D coordinate systems, the OWLS model uses “y” as an upward direction and “z” as a forward direction. The reason to do so is that the OWLS 3-D system is an advanced system from previous 2-D’s. For a 2-D system, the "y" axis points upward, so for the 3-D simply add an axis towards the paper as "z".

In the OWLS visualization model, each point (P) is represented by three-dimensional coordinate values P{x, y, z}; each line (L) is represented by two or more points L{P1, P2, ...}; each facet (F) is represented by an clockwise-ordered points F{P4, P2, P6, ...}; an area or a 3-D object, or a watershed, or a study area (A) is represented by a set of facets A{F1, F2, ...}. The process of representing an area so that it can be seen on the computer screen as a three-dimensional object is to mathematically transfer all the points of the area from Pi{x, y, z} onto the screen coordinates Si{x, y}. There are four coordinate systems involving in this transformation (Figure 5 - 2):

(1) Model Coordinate System: a 3-D coordinate system of actual field data, i.e., data of location and elevation from a field survey of the study watershed;

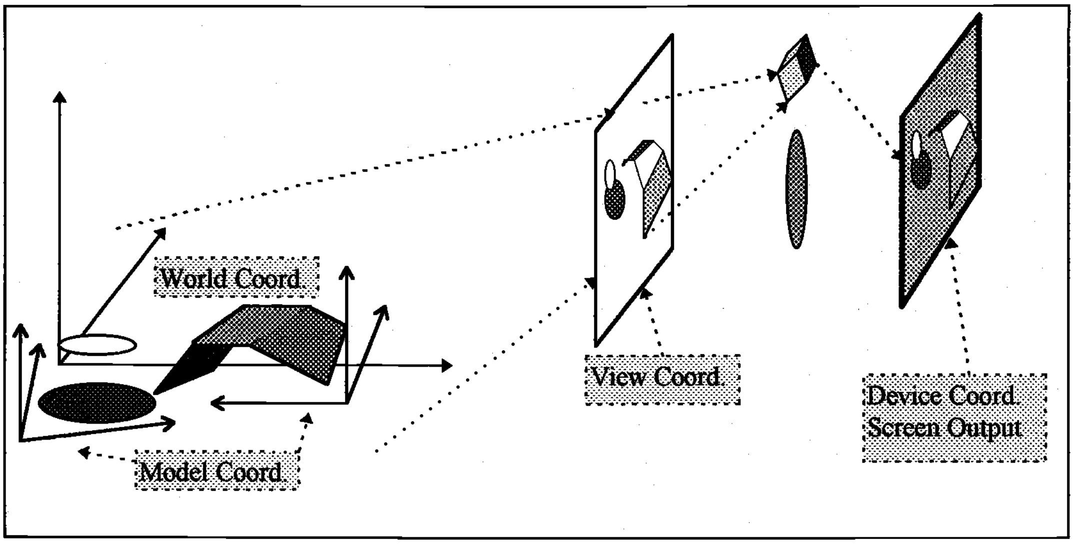


Figure 5 - 2. Coordinate systems of the OWLS visualization model.

(2) World Coordinate System: a 3-D coordinate system of the entire space. In some cases, there may be two or more study areas next to each other and each has its own Model Coordinate System. The World Coordinate System is used to unify them.

(3) View Coordinate System: a 2-D coordinate system of the viewer. When a camera is used for taking pictures, the image seen from the camera is a portion of the whole "world." It is 2-dimensional. The view coordinate system is the same concept as that of a camera. It projects all 3-D objects within its view range into 2-D;

(4) Device Coordinate System: a coordinate system for the output device, i.e., computer window screen, printer, or computer memory. This coordinate system provides the physical dimensions that a graph will show.

The transformation of a point from Model Coordinate System to Device Coordinate System involves many standard matrix transformations as well as some graphical techniques. Algorithms like space sorting, searching, back surface removal, color painting, polygon filling and so on are also supporting programs in the OWLS model. The final result, just like taking pictures, is the display of a 3-D study area (or watershed) on the computer screen.

5.2. *The Visualization Model*

The OWLS visualization model includes a 3-D model and 2-D model. The 3-D model is designed to visualize topographical, geographical, and hydrologic components of a watershed. The 2-D model is designed to display simulation results from the OWLS hydrologic model. Both models have an Platform Independent Graphics Interface (PIGI) module to bridge the OWLS model with graphic library from different computer platforms (operating systems). This feature makes it possible to run under different operating systems with little modification. The current version of the OWLS visualization model was developed and tested under both MS Windows 3.1 and Windows 95 operating systems. It allows a user to view a watershed from different angles. Information that the OWLS visualization model can present is as follows:

5.2.1. *3-D Visualization*

The 3-D visualization model is able to present a 3-D view of a watershed. It can also present the 2-D map layout after defining the view direction as directly overhead. The dynamic animation for both watershed and hydrologic processes can also be represented with this model:

(1). Basin Topographic Visualization: including contoured, meshed, or shaded display of the study area;

(2). Basin Characterization Visualization: including soil type, soil depth, flow path, drainage area, stream network, watershed boundary, and channel river-meter;

(3). Basin Animation: including bird-viewing, time-animation;

(4). Dynamic Watershed Hydrologic Simulation: a visual animation of the hydrologic processes in a watershed, in both space and time. The hydrologic component that can be dynamically visualized by the OWLS mode includes:

- (4.1) Watershed Cells Hydrologic Components:
    - (4.1.1) Canopy: intercepted water depth, intercepted snow depth, ET, netrain;
    - (4.1.2) Surface: water depth, snow depth, infiltration, ET, surface flow;
    - (4.1.3) Subsurface: soil moisture, water depth, ET, subsurface flow;
    - (4.1.4) Macropore Pipe System: water depth, incoming flow from soil, incoming flow from surface, total incoming flow from other system, macropore flow;
    - (4.1.5) General: precipitation, total water depth, total outgoing flow;
- (4.2) Stream Segments Hydrologic Components:

(4.2.1) Flow: flow velocity, discharge;

(4.2.2) Channel Geometry: channel width, channel water depth;

(4.3) Hydrograph:

(4.3.1) Flows: both measured and simulated. The simulated flow curve will proceed with the time;

(4.3.2) Precipitation and evaportraspiration;

(4.3.3) Air Temperature.

5.2.2. *2-D Visualization*

The 2-D visualization model handles graphical outputs from the OWLS hydrologic model. It displays the simulated hydrologic components as a function of time as well as the basin information and the parameters that were used for the model. It is a special design for the hydrologic model, and is especially usefull for parameter calibration and presentation of results. Hydrographs shown in the previous chapter were created from this model and they include the following information:

(1) Text information: basin name, size, simulation and system parameters. The importance of these parameters decreases from top to bottom;

(2) Air temperature curve, simulated from the daily characterization data (minimum, average and maximum);

(3) Precipitation as observed and evaportranspiration as simulated;

(4) Hydrographs for both simulated and observed flows;

(5) Simulated flow components, including surface flow, macropore pipe flow, and soil flow;

(6) Water in different vertical components of the basin, including canopy intercepted water, canopy intercepted snow, surface water, surface snow, macropore pipe water and soil water.

5.3. *Examples*

There are many combinations of outputs from the OWLS visualization model and some of their outputs have been presented in Chapter Four. However, Table 5 - 1 explains a list of figures (Figure 5 - 3 to 11) as additional examples of outputs from the OWLS visualization model.

Table 5-1. List of example figures from the visualization model

Figure #	Figure Name	Explanations
5 - 3	Contour of the BBWM watershed	Steady view for detail exam of the watershed topography.
5 - 4	Flowpath in the BBWM watershed	The simulated flow paths tell the trail that water will go through.
5 - 5	Delineated watershed boundary, stream and digitized stream	The simulated watershed boundary and stream, associated with the digitized stream from field survey. The simulated stream matches the digital stream, and as a surplus, extending the stream for possible channels.
5 - 6	Flowpath tree and drainage area of the EBB watershed	Gray scale tells the differences of drainage area. Lighter means smaller drainage area and vise verse.
5 - 7	Dynamic Watershed Hydrologic Simulation (DWHS) I: Total water depth and discharge at the EBB watershed at flow peak	A frame from the DWHS when stream flow is in peak. Total water depth is identified by the color of the cells. The more the red, the deeper the water. Discharge is identified by the color of the channel. Notice that the with of the channel has been amplified 10 times.
5 - 8	Dynamic Watershed Hydrologic Simulation II: Total water depth and discharge in the EBB watershed at recession limb	A frame after the peak flow shown in Figure (5 - 8). Compare this figure to (5 - 8) and notice the color changes for both cells and channel segments.
5 - 9	Dynamic Watershed Hydrologic Simulation (DWHS) III: Total cell flow and discharge in the EBB watershed at flow peak	A frame from the simulated results at the EBB watershed. Total Cell Flow includes surface, subsurface and macropore flow draining out from the cell. For cells not next to the stream channel, the flow from a cell goes to or passes through the cells below it. Notice their spatial distribution identified by their color.
5 - 10	Dynamic Watershed Hydrologic Simulation (DWHS) IV: Total macropore flow and velocity in the EBB watershed	A frame from the EBB watershed. Macropore flow is acting as a very important hydrologic component in the watershed hydrology. The flow velocity in the channel can also be identified by the color of the channel.
5 - 11	Dynamic Watershed Hydrologic Simulation (DWHS) V: Soil moisture content, discharge and stream depth in the EBB watershed in 3-D	A frame from the EBB watershed. 3-D view of the DWHS. Cell colors represent the relative soil moisture content. Stream colors identify the discharge. The height of the stream can also be visualized by scaling the depth 1000 times.

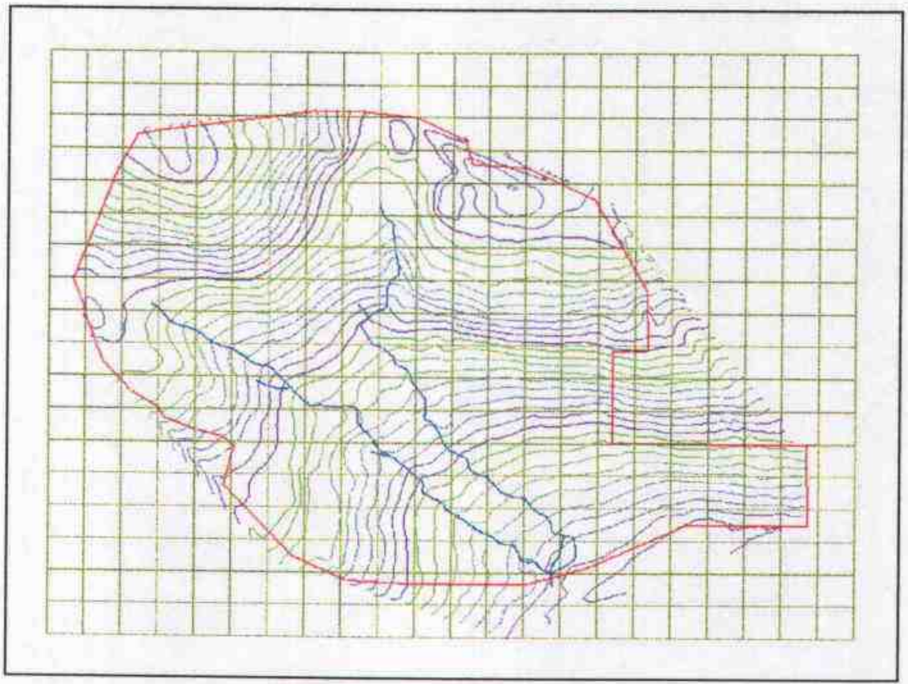


Figure 5 - 3. Contour of the BBWM watershed

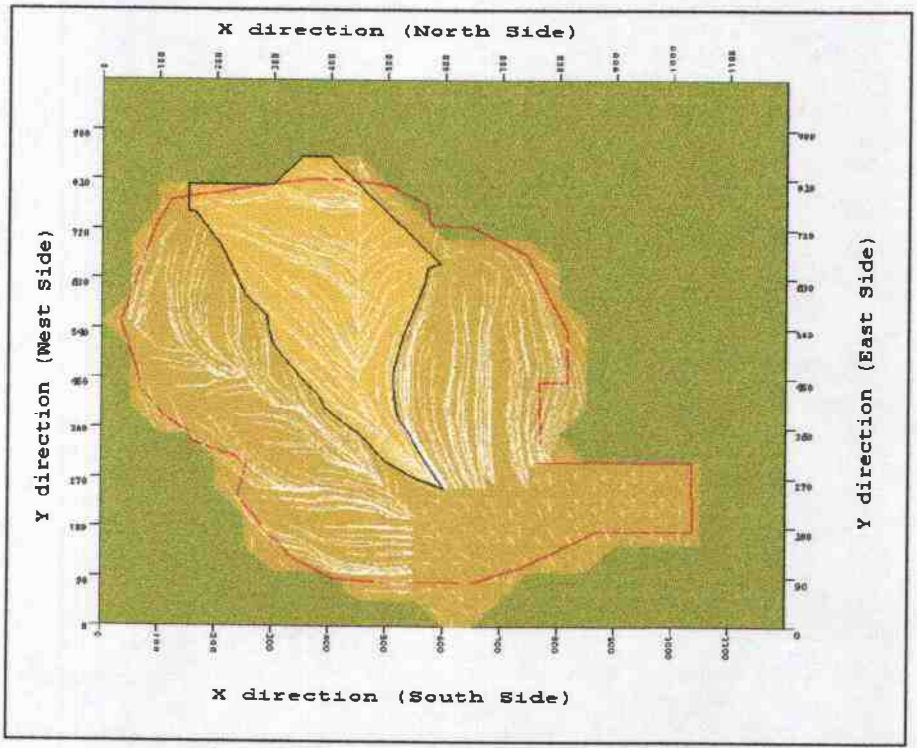


Figure 5 - 4. Flowpath of the BBWM watershed

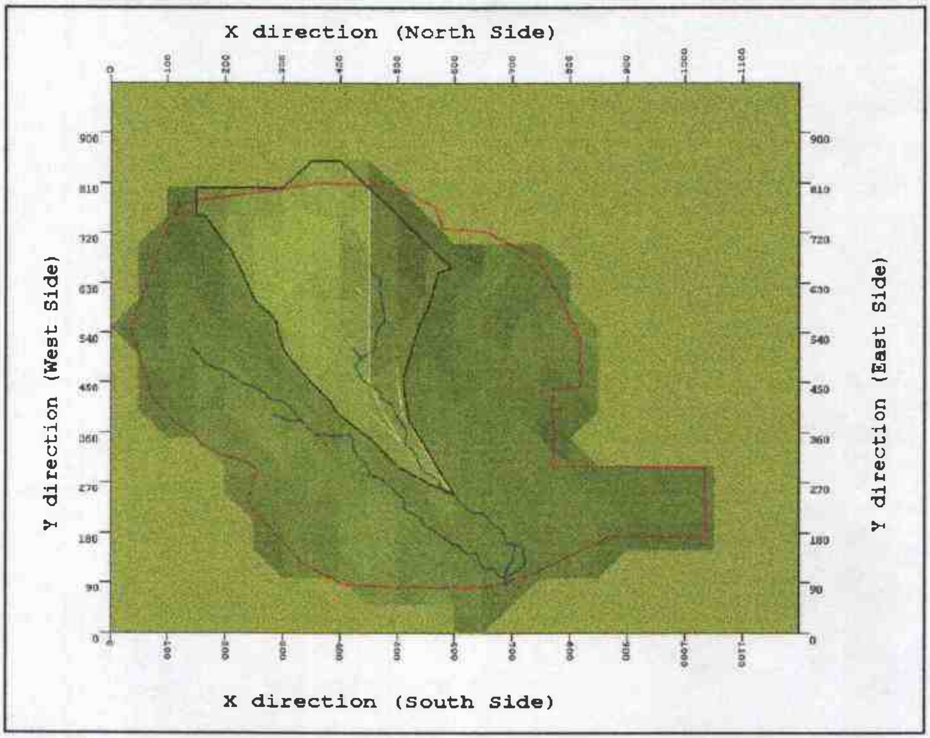


Figure 5 - 5. Delineated watershed boundary, stream and digitized stream

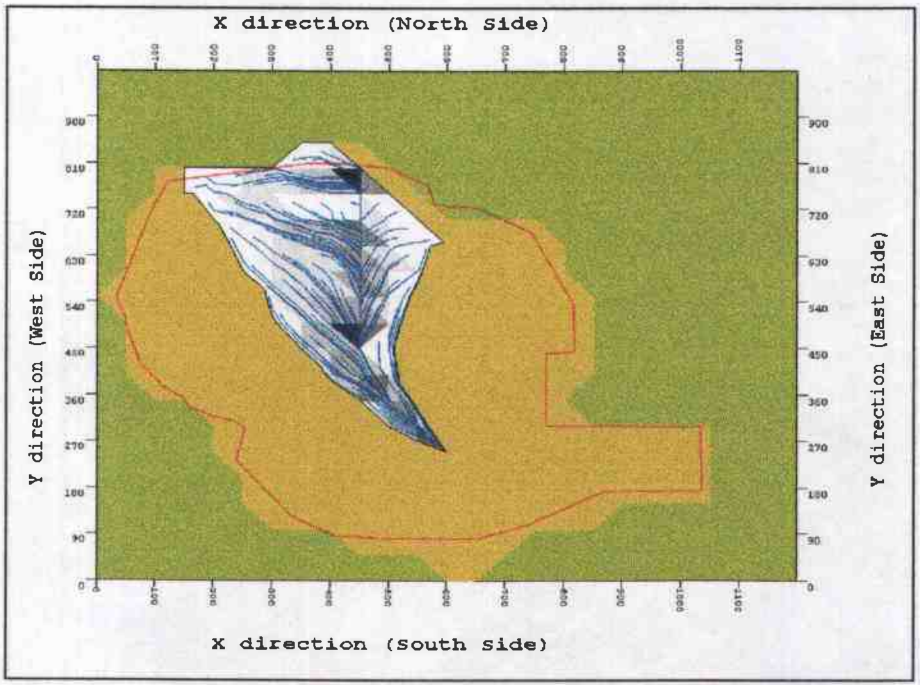


Figure 5 - 6. Flowpath tree and drainage area of the EBB watershed

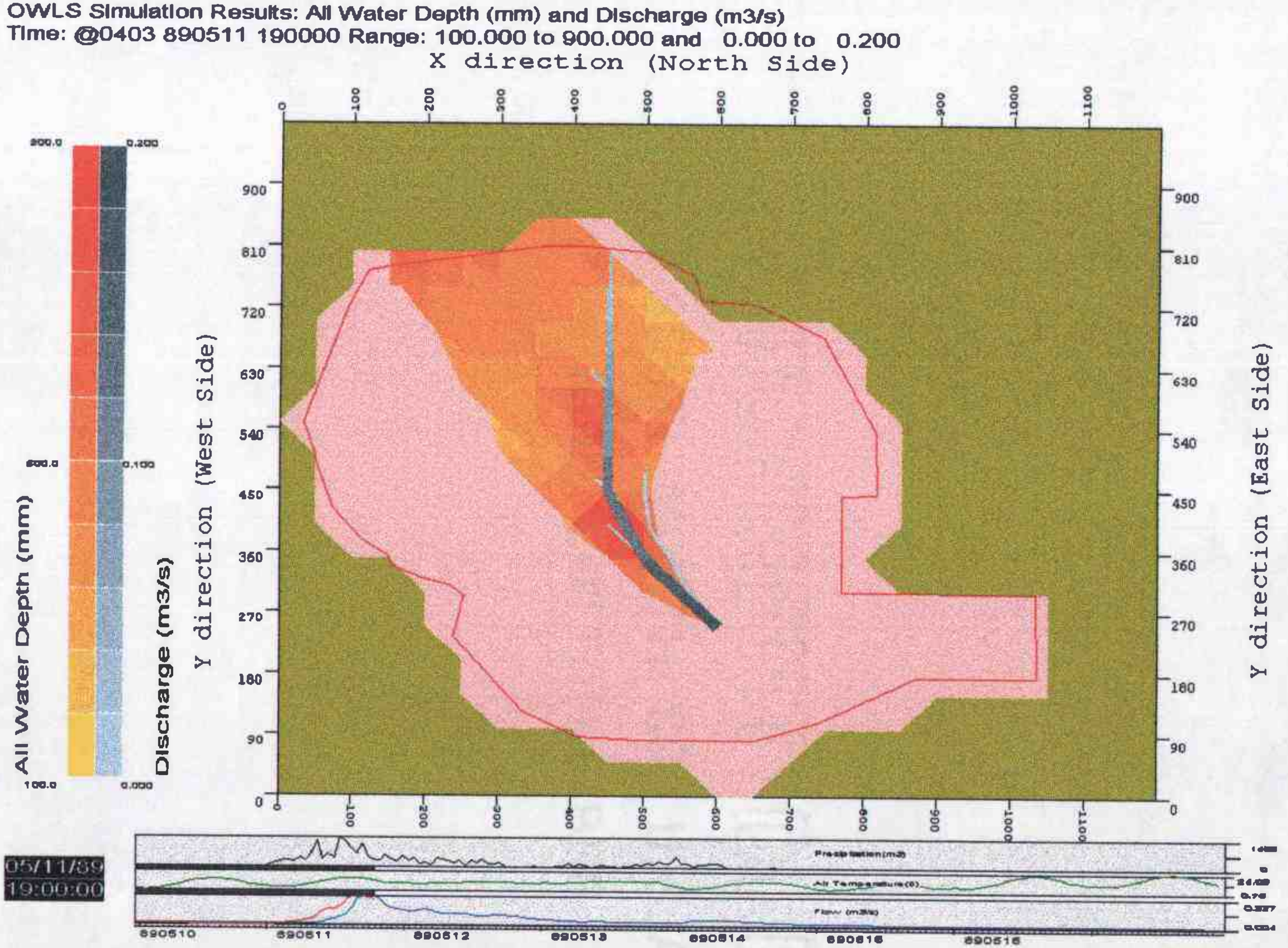


Figure 5 - 7. Dynamic Watershed Hydrologic Simulation (DWHS) I: Total water depth and discharge in the EBB watershed at flow peak

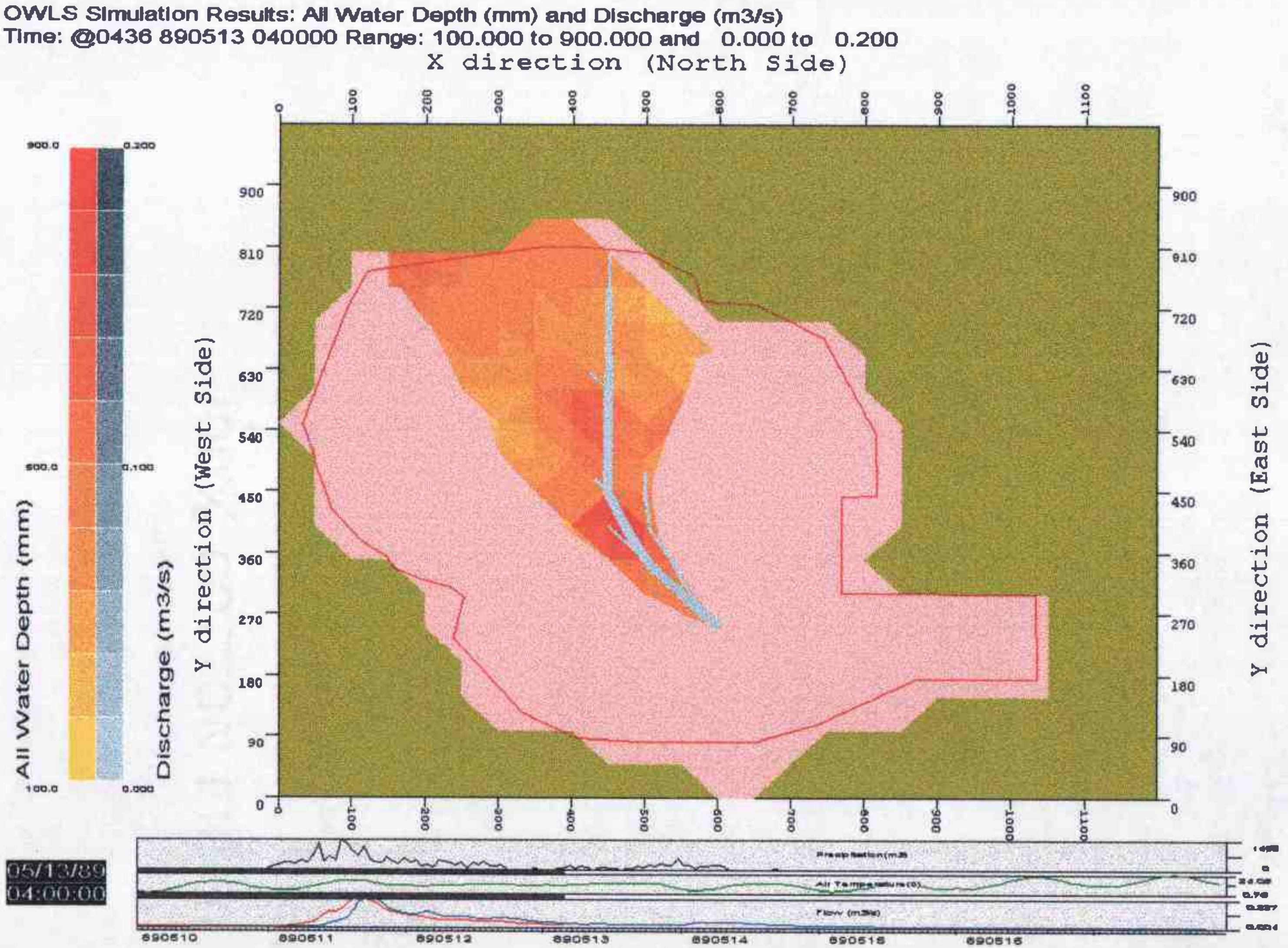


Figure 5 - 8. Dynamic Watershed Hydrologic Simulation (DWHS) II: Total water depth and discharge in the EBB watershed at recession limb

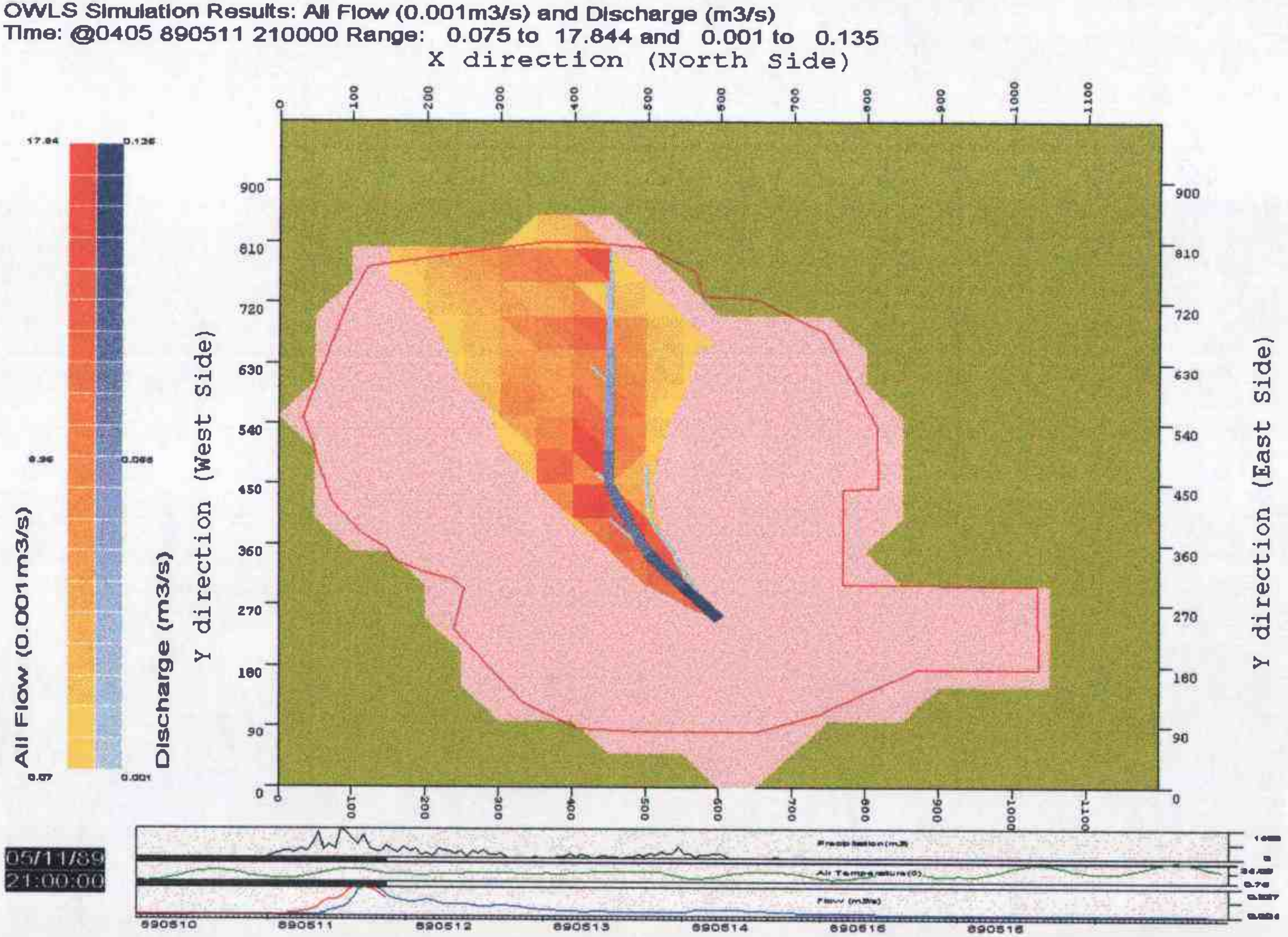


Figure 5 - 9. Dynamic Watershed Hydrologic Simulation (DWHS) III: Total cell flow and discharge in the EBB watershed at flow peak.

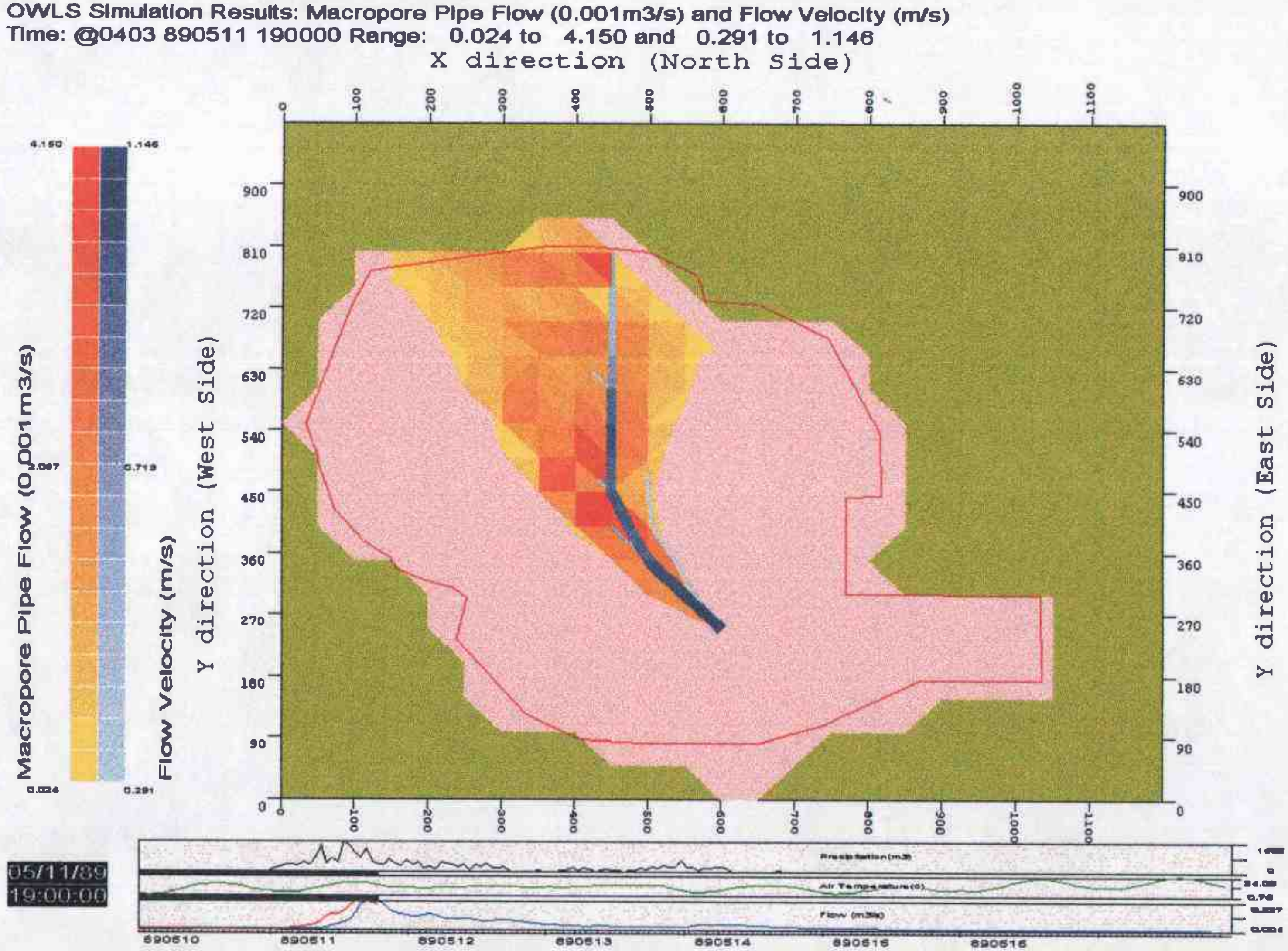


Figure 5 - 10. Dynamic Watershed Hydrologic Simulation (DWHS) IV: Total macropore flow and velocity in the EBB watershed.

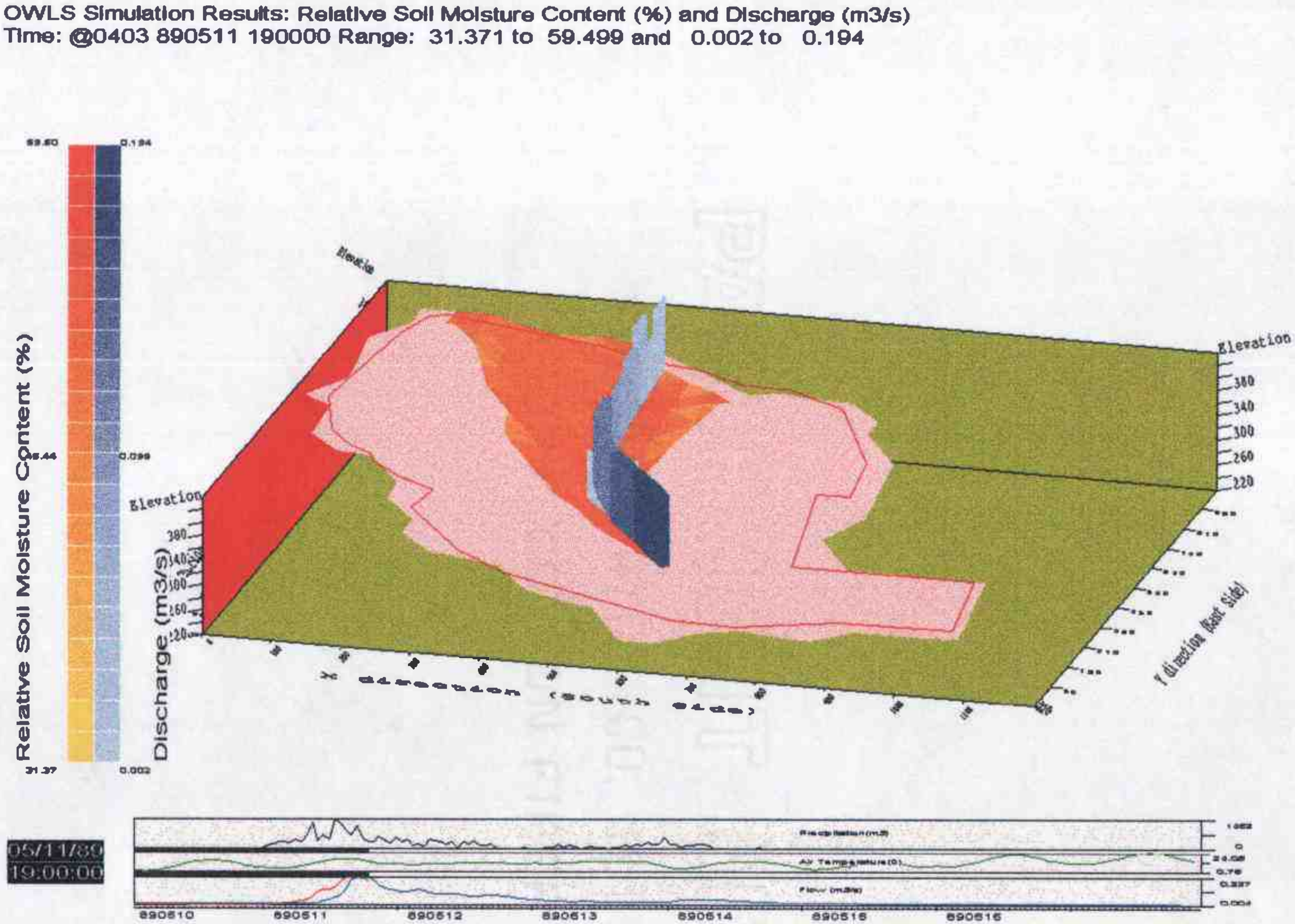


Figure 5 - 11. Dynamic Watershed Hydrologic Simulation (DWHS) V: Soil moisture content, discharge and stream depth (1/1000) in the EBB watershed in 3-D.

Chapter Six: *Conclusion and Future Development*

The Object Watershed Link System (OWLS) is a physically-based hydrologic model. The following characterizations identify its main features relative to other watershed models:

1. It is coded in C++ object orientated programming (OOP), containing modules and codes that are reusable and expandable. Also, it utilizes a dynamic memory allocation mechanism to reduce the usage of computer memory when running. This makes it possible to run a large program in a personal computer. In addition, the object-orientated programming makes the platform independent graphic interface (PIGI) possible in the OWLS, which provides great potential for the future development of the model.

2. It is structured as objects and linkages; all watershed components, hydrologic components, and even time and measurement units are represented as objects and are linked with each other. Each object not only identifies itself from others, but also carries characteristic data to wherever it goes. It provides a higher efficiency for running the model.

3. Automatic watershed delineation is vector-based. It identifies the watershed boundary, possible stream channel, and their hydrologic characteristics. It provides a harmonious simulation base for the hydrologic model.

4. The OWLS model is a vector-based and true 3-D hydrologic simulation model. A watershed is a linkage of three dimensional cells, edges and nodes. Thus, hydrologic components are nested into these 3-D cells, edges and nodes; thus, the hydrologic processes become a dynamic linkage among them.

5. The OWLS model is designed to handle any kind of cell geometry or their combination in terms of watershed topography or hydrology. Because of the OOP and dynamic memory management, the OWLS model defines the object of a cell having undetermined numbers of nodes and edges; this can be accomplished while data are being input. Therefore, each cell of a watershed can be a triangle, a rectangle, or an x-edges polygon, there are no restriction on cell characteristics. To handle different cell geometries, the Equivalent-Rectangle Simplification (ERS) procedure unifies all cells by finding a hydrological equivalent-rectangle for each and calculates the weights for each edges.

6. The hydrologic model is constructed over the 3-D watershed, from each cell, to each segment of the potential stream. In the vertical dimension, there are three and half layers: the canopy layer, the

surface layer, the soil layer and the half -- macropore pipe layer, which is nested into the soil layer and invisible. Hydrologic processes considered in the OWLS model includes:

(1) Interception: in canopy layer, calculated from the leaf area index of a particular season using a water balance method;

(2) Evaporation: in canopy and surface layer, defined as water vaporized from the water surface of the related layer, calculated from solar radiation, air temperature, and canopy coverage.

(3) Transpiration: in canopy and soil layer, defined as water vaporized from the layer itself. Calculated from potential Evaporation plus the restriction from the available soil moisture and the soil water supply capacity.

(4) Snow melting: in canopy and surface layer, calculated from the simple degree-day function. No energy budget has been taken into account.

(5) Infiltration: in surface layer, calculated by modified Horton's equation with relative soil moisture content to define the Hortonean time.

(6) Macropore surface water entry: in surface layer, calculated as proportional to infiltration.

(7) Macropore soil water entry: in soil layer, calculated as a function of soil water depth and the relative soil moisture content.

(8) Surface overland flow: calculated by the kinematic wave finite-difference approximation using Manning's equation.

(9) Subsurface flow in the soil: calculated by the kinematic wave finite-difference approximation using Darcy's equation.

(10) Macropore flow of the macropore pipe system in the soil: calculated by multiple-pipe finite-difference approximation derived from the energy balance equation;

(11) Hillslope flow routing: all horizontal flows including surface, subsurface and macropore flow are routed by utilizing the edge weights calculated from the ERS method in combination with the kinematic wave finite differential calculations. A three-dimensional routing has been converted into a virtually 1-D flow routing procedure. This technique dramatically reduces the complexity of the flow routing model and increases the flexibility of the model as well as the calculation speed.

(12) Stream flow routing: since information about the stream segments has been calculated from the automatic watershed delineation model and the stream geometric model, the kinematic wave method has been utilized for channel flow routing.

7. The visualization model represents a significant component of the OWLS watershed model. Specially designed for watershed hydrologic simulation and animation, this built-in but relatively stand-alone model provides more information than ever before. The OWLS model also provides data outputs in text format for custom graphics.

8. The simulation results from the OWLS model not only provided valuable information about the sources of flow generation from a watershed, but also dynamically visualized the concept of Variable Source Area (VSA) from a watershed in a three dimensional aspect. Source area concept in the OWLS model has been expanded: Flow not only comes from the surface of riparian cells, but also from soil macropore and soil matrix.

9. Tremendous amounts of distributed-and-dynamic flow data generated from the OWLS model provide a strong fundation for applications in other fields of studies. For example, by setting the pollution source in the watershed, one can calculate the transportation of the pollutants in the watershed using the data from the OWLS model.

Future development for improving the OWLS model should include the following aspects:

(1) Model testing by applying the OWLS model in other watershed areas. More applications should bring model coefficients closer to reality; more testing could discover unforseen problems (or bugs) currently embedded in the model;

(2) Laboratory testing and field survey to support, verify, modify ,or even rewrite the theories, assumptions, and equations that have been developed and used in the OWLS model. The model has approached many uncertain areas in the hydrology field. Additional data from the field or lab testing will able to enhance the model;

(3) Extending the application of the OWLS model to include watershed sedimentation, water chemistry and pollution processes;

(4) Adding additional features into the stream channel model simulating in-stream woody debris so that it can be used to address more complicated problems for stream ecology;

(5) Expanding the model into different area, e.g. agriculture fields, roads, and urban areas.

(6) Improving the User Interface of the OWLS model so that it can be easier to use.

Bibliography

Band, L.E. 1986. Topographic Partition of Watersheds with Digital Elevation Models. *Water Resources Research* 22:15-24.

Band, L.E. 1993. Extraction of Channel Networks and Topographic Parameters from Digital Elevation Data. IN: K. Beven and M.J.Kirkby. *Channel Network Hydrology*. John Wiley & Sons Ltd.

Band, L.E., J.P.Patterson, R.Nemani, and S.W.Running. 1991. Ecosystem Processes at the Watershed Scale: 2: Incorporating Hillslope Hydrology. *Agricultural and Forest Meteorology*, review.

Barling, R.D., I.D.Moore, and R.G.Grayson. 1994. A quasi-dynamic wetness index for characterizing the spatial distribution of zones of surface saturation and soil water content. *Water Resource Research* 30(4):1029-1044.

Bathurst, J.C. 1986. Physically-based Distributed Modelling of an Upland Catchment Using the Système Hydrologique Européen. *Journal of Hydrology* 87:79-102.

Beven, K.J. 1989. Changing Ideas in Hydrology. The Case of Physically Based Models. *Journal of Hydrology* 105:157-172.

Beven, K.J., and E.F.Wood. 1983. Catchment Geomorphology and the Dynamics of Runoff Contributing Areas. *Journal of Hydrology* 65:139-158.

Beven, K.J., A. Calver, and E.M.Morris 1987. The Institute of Hydrology Distributed Model. *Institute of Hydrology Report*. 98, Institute of Hydrology, Oxon, U.K.

Beven, K.J., M.J.Kirkby, N.Schofield, and A.F.Tagg. 1984. Testing a Physically-Based Flood Forestcasting Model (TOPMODEL) for Thre U.K. Catchments. *Journal of Hydrology* 69:119-143.

Beven, K.L., and M.J.Kirkby. 1979. A Physically based, variable contributing area model of basin hydrology. *Hydrological Sciences Bulletin* 24:43-69.

Binley, A., and K. Beven. 1992. Three-Dimensional Modelling of Hillslope Hydrology. IN: Beven, K.J. and I.D.Moore (ed.) *Terrain Analysis and Distributed Modelling in Hydrology*. John Wiley & Sons.

Black, J.N. 1956. The Distribution of Solar Radiation over the Earth's Surface. *Archieves for Meteorology, Geophysis , and Bioclimatology*. B.7:165-189.

Chairat, S., and J.W. Delleur. 1993. Effects of the Topographic Index Distribution on Predicted Runoff Using GRASS. *Water Resources Bulletin* 29(6):1029-1034.

Chow, V. T., D.R. Maidment, and L.W. Mays. 1988. *Applied Hydrology*. McGraw-Hill Book Company. 627pp.

Christiansen, J.E. 1966. Estimating pan evaporation and evapotranspiration from climatic data. *ASCE Irrig. and Drain. Spec. Conf. Proc.*, Nov. 24, pp193-231.

Crawford, N.H., and R.K.Linsley, 1966. Digital Simulation in Hydrology: Stanford Watershed Model IV. Tech. Rep. 39, Depart. Civil Eng., Stanford Univ., Stanford, CA, USA, 210pp.

Darcy, H. 1856. Les Fontaines Publiques de la Ville de Dijon. Dalmont, Paris.

Dawes, W.R., and D. Short. 1994. The Significance of Topology for Modeling the Surface Hydrology of Fluvial Landscapes. *Water Resource Research* 30(4):1045-1055.

Douglas, D.H. 1986. Experiments to Locate Ridges and Channels to Create a New Type of Digital Elevation Model. *Cartographica* 23:29-61.

Durand, P., A. Robson, and C. Neal. 1992. Modelling the Hydrology of Submediterranean Montane Catchments (Mont-Lozere, France) using TOPMODEL: Initial Results. *Journal of Hydrology* 139:1-14.

Dury, G.H. 1973. Magnitude-Frequency Analysis and Channel Morphometry. IN: M. Morisawa(ed). *Fluvial Geomorphology, A Proceedings Volume of 4th Annual Geomorphology Symposia Series*, Binghamton, NY. State Univisity of New York. pp91-121.

Famiglietti, J.S., E.F.Wood, M.Sivapalan, and D.J.Thongs. 1992. A Catchment Scale Water Balance Model for FIFE, *J. Geophys.Res.* 97(D17):18997-19007.

Foley, J.D., A.V. Dam, S.K.Feiner, J.F. Hughes, and R.L.Philips. 1993. *Introduction to computer graphics.* Addison-Wesley Publishing Company, Inc. pp557.

Freeze, R.A., and R. L. Harlan 1969. Blueprint for a Physically-based Digitally-simulated Hydrologic Response Model. *Journal of Hydrology* 9:237-258.

Grayson, R.B., I.D.Moore, and T.A.McMahon. 1992a. Physically Based Hydrologic Modeling, 1, A Terrain-Based Model for Investigative Purpose. *Water Resource Research* 28(10):2639-2658.

Grayson, R.B., I.D.Moore, and T.A.McMahon. 1992b. Physically Based Hydrologic Modeling, 1, Is the Concept Realistic? *Water Resource Research* 28(10):2659-2666.

Greysukh, V.L. 1967. The Possibility of Studying Landforms by Means of Digital Computer. *Soviet Geography*, Review and Translated. pp.137-149.

Gupta, R.S. 1989. *Hydrology and Hydraulic Systems.* Prentice-Hall, Inc. pp627.

Haan, C.T., H.P.Johnson, and D.L.Brakensiek, 1982. *Hydrologic Modeling of Small Watersheds.* An ASAE Monograph, Number 5 in a series published by American Society of Agriculture Engineers. pp533.

Jensen, M.E. and R.H.Haise. 1963. Estimating evapotranspiration from solar radiation. *ASCE Proc.* 89(IR4):15-41.

Jenson, S.K. 1985. Automated Derivation of Hydrologic Basin Characteristics from Digital Elevation Models. *Proceedings of Auto-Carto 7, Digital Representation of Spatial Knowledge*, pp.301-310.

Jones, N.L., S.G.Wright & D.R.Maidment, 1990. Watershed Delineation with Triangle-Based Terrain Models. *Journal of Hydraulic Engineering*, 116(10): 1232-1251.

Lee, Richard, 1978. *Forest Microclimatology*. Columbia University Press.

Martz, L.W., and J. Garbrecht. 1993a. Automated Extraction of Drainage Network and Watershed Data from Digital Elevation Models. *Water Resources Bulletin* 29(6):901-908.

Martz, L.W., and J. Garbrecht. 1993b. Network and Subwatershed Parameters Extracted from Digital Elevation Models: The Bills Creek Experience. *Water Resources Bulletin* 29(6):909-916.

Monteith, J.L. 1973. *Principles of Environmental Physics*. In E.J.W. Barrington, and A.J.Willis (Eds): A Series of Student Texts in Comtemporary Biology. American Elsevier Publishing Co. Inc.

Moore, I.D., and R.B.Grayson. 1991. Terrain-Based Catchment Partitioning and Runoff Prediction Using Vector Elevation Data. *Water Resources Research* 27(6):1177-1191.

Moore, I.D., E.M.O'Loughlin, and G.J.Burch. 1988. A Contour-based Topographic Model for Hydrological and Ecological Applications. *Earth Surface Processes and Landforms*. 13:305-320.

Norton, S.A.; R.F.Wright, J.S.Kahl, and J.P.Scofield. 1992. The Magic simulation of surface water acidification at, and first year results from, the Bear Brook Watershed Manipulation, Maine, USA. *Environmental Pollution* 77: 279-286

O'Callaghan, J.F., and D.M.Mark. 1984. The Extraction of Drainage Networks from Digital Elevation Data, *Compu. Vision Graphics Image Process*. 28:323-344.

O'Loughlin, E.M. 1981. Saturation Regions in Catchments and Their Relations to Soil and Topographic Properties. *Journal of Hydrology* 53:229-246.

O'Loughlin, E.M. 1986. Prediction of Surface Saturation Zones in Natural Catchments by Topographic Analysis. *Water Resource Research* 22:794-804.

O'Loughlin, E.M. 1990. Modeling Soil Water Status in Complex Terrain. *Agr. For. Met.*, 50:23-38.

Puecker, T.K. and D.H.Douglas. 1975. Detection of Surface-Specific Points by Local Parallel Processing of Discrete Terrain Elevation Data. *Computer Graphics and Image Processing* 4:375-387.

Robson, A.J., P.G.Whitehead, and R.C.Johnson. 1993. An Application of a Physically Based Semi-Distributed Model to the Balquhidder Catchments. *Journal of Hydrology* 145:357-370.

Ross, J.K., and H.G.Tooming. 1968. In 'Aktinometriya i optika atmosfery' (Actinometry amd Atmospheric Optics), pp283-288. Valgus, Tallinn (Russian).

Running, S.W., and J.C.Coughlan. 1988. A General Model of Forest Ecosystem Processes for Regional Applications. I. Hydrologic Balance, Canopy Gase Exchange and Primary Prodution Processes.

Ecological Modeling 42:125-154.

Tachikawa, Y., M.Shiiba, and T.Tahasao. 1994. Development of a Basin Geomorphic Information System Using a TIN-DEM data structure. *Water Resource Bulletin* 30(1):9-17.

Todd, D.K. 1980. *Groundwater Hydrology*, 2nd ed., John Wiley & Sons, Inc., New York.

Toriwaki, J., and T. Fukumura. 1978. Extration of Structural Information from Grey Pictures. *Computer Graphics and Image Processing* 7:30-51.

Wigmosta, M.S., L.W.Vail, and D.P.Lettenmaier. 1994. A Distributed Hydrology-Vegetation Model for Complex Terrain. *Water Resources Research* 30(6):1665-1679.

Zhang, W., and D.R.Montgomery, 1994. Digital Elevation Model Grid Size, Landscape Representation, and Hydrologic Simulations. *Water Resources Research* 30(4):1019-1028.

APPENDICES

Appendix I: *List of Objects used in OWLS*

OWLS Modules	Object Name (Induced Object)	Descriptions	Features (Additional Features)	Functions (Additional Functions)
OWLSUnit	OWLSUnit	Modules for unit conversions	MeasureSystem MeasureLength	convert
	(OWLSLength)	length object for unit conversion	(value)	
	(OWLSArea)	area object for unit conversion	(value)	
	(OWLSVolume)	volume object for unit conversion	(value)	
	(OWLSTemperature)	temperature object for unit conversion	(value)	
OWLSModel	OWLSModel	simulation model main object	*parent (OWLSHydrology) startT entT currentT step saveInterval flow[] reflow[] conductivity[] streamFlowV[] streamSurFlow[] streamSoilFlow[] streamPipeFlow[] depthFlow[] depthRain[] depthET[] depthCanopy[] depthSurface[] depthSoil[] volumePipe[] soilMoist[] basinTemp[]	getInitial waterBalance routing run print save read
	OWLSPhysicalModel	Physical Simulation Model main object	(***canopy) (***surface) (***subsurface) (***macropore) (*soilDepth) (*moisture) (*alpha) (nSegments) (**segment) (*streamFlow) (*edge2Seg) (*velocity) (*discharge) (*channelWidth) (initFileName) (inputUnit) (outputUnit) (infiltration_k)	(draw) (saveInterruption) (saveGraphics) (readInterruption) (readGraphics) (sortCells) (sortSegments) (infiltration) (snowMelt) (potentialET) (surfaceFlow) (surfaceFlowRate) (subsurfaceFlow) (subsurfaceFlowRate) (macroporeFlow) (canopyWaterBalance) (surfaceWaterBalance) (subsurfaceWaterBalance)

Appendix I: *List of Objects used in OWLS (Continued)*

OWLS Modules	Object Name (Induced Object)	Descriptions	Features (Additional Features)	Functions (Additional Functions)
			(infiltration_a) (conductivityAdjust) (infiltration0Adjust) (surfaceMacroporeConst) (soilMacroporeConst) (snowMelt_Tb) (snowMelt_Df) (ET_a) (ET_b) (underCanopyETConstant) (canopyETConstant) (soilETConst) (roughness) (soilMoisture) (layerWeight1) (layerWeight2) (initialStreamDepthRatio) (porosity) (depthConstant) (depthPow) (widthTopConstant) (widthTopPow) (widthBotConstant) (widthBotPow) (minDiameter) (pipeRatio) (radiusA) (radiusB) (radiusV) (frictionCoeff) (countA) (countB) (countC) (countD) (countE) (times) (traceTime) (*fTime) (*sortedSegIdx) (*sortedCellIdx) (*virtualIdx) (*pipe) (useHorton) (isCrossModel) (isMacroModel) (useDepthOutput) (useDirectInputs) (interativeErr) (maxIterations) (upSurfaceWaterDepth) (dnSurfaceWaterDepth) (upSoilWaterTable) (dnSoilWaterTable) (upPipeWaterVolume) (dnPipeWaterVolume) (upSurfaceInFlow) (upSoilInFlow) (upPipeInFlow)	(macroporeWaterBalance) (streamWaterBalance) (getCrossArea) (getRoutingTime) (outletFlow) (getTime) (putTime) (initialMacropore) (initialStreamWater) (initialStreamFlow) (initialSegment) (setEdgeZeroWeight) getCellBoundaryData) (shift)

Appendix I: *List of Objects used in OWLS (Continued)*

OWLS Modules	Object Name (Induced Object)	Descriptions	Features (Additional Features)	Functions (Additional Functions)
OWLSObject	OWLSObject	Geometric objects for vitualization	*next (OWLSObject) name localTransMatrix	updateVC draw print matrix operator * translate rotateX rotateY rotateZ mirrorX mirrorY mirrorZ scale
	OWLSText	text object for visualization model	(*s) (string of the text) (headMC) (widthMC) (heightMC) (direction) (upVector) (textFont) (tailMC) (cornerMC) (headMC) (tailVC) (cornerVC) (widthVC) (heaghtVC)	
	OWLSLine	line object	(color) (length) (startMC) (endMC) (startVC) (endVC)	(getLength)
	OWLSLineBunch	object for line groups	(nNodes) (nLines) (*nodesMC) (*nodesWC) (*nodesVC) (*linesPtrs)	(operator =)
	(OWLSContour)	object for a contour line	(*linesPtrs) (value)	
	OWLSPolygon	polygon object	(nNodes) (color) (*nodesMC) (*nodesVC) (normalVC)	(include) (intersect) (reflect)
	OWLSPolyhedron	object of a polygon group	(nNodes) (nFacets) (*nodesMC) (*nodesWC) (*nodesVC) (*facetPtrs)	
	(OWLSIrrObject)	irregular 3-D object		
	(OWLSTriTopo)	triangular 3-D object		
	(OWLSRecTopo)	rectangular 3-D object		

Appendix I: *List of Objects used in OWLS (Continued)*

OWLS Modules	Object Name (Induced Object)	Descriptions	Features (Additional Features)	Functions (Additional Functions)
	(OWLSCone)	object for 3-D cone		
	(OWLSInnerBox)	object for a 3-D in-box		
	(OWLSCube)	object for a 3-D cube		
	(OWLSEgg)	object for a 3-D egg		
	(OWLSCylinder)	object for a 3-D cylinder		
	(OWLSPyramid)	object for a 3-D pyramid		
	OWLSWatershed	3-D watershed object	(nNodes) (nEdges) (nCells) (area) (projectedArea) (rootIdx) (**nodesMC) [OWLSNode] (**edgePtrs) [OWLSEdge] (**cellPtrs) [OWLSCell] (*boundary) [OWLSebTree] (*biTree) [OWLSBiTree] (*flowPath) [OWLSPath] (*stream) [OWLSPath] (**leaf) [OWLSPathNode]	(getInput) (read) (save) (getEqDepth) (getEdgeIndex) (getLineEdgeIndex) (isOntheEdge) (isAPeak) (slopeOf2Points) (getCrossingPoint) (growBoundaryTree) (getBoundary) (splitCell) (getFlowPaths) (getFlowPathTree) (addPathToTree) (getTreeLeaves) (markStream) (runHydrology) (runChemistry) (sortList) (removeDepress)
	(OWLSHydrology)	Hydrologic Model Simulation Object	(complexID) (rainCPLX) (airTempCPLX) (soilOTempCPLX) (soilBTempCPLX) (soilCTempCPLX) (soilCPLX) (vegCPLX) (*rain0) [OWLSRain] (*airTemp0) [OWLSTemp] (*soilOTemp0) [OWLSTemp] (*soilBTemp0) [OWLSTemp] (*soilCTemp0) [OWLSTemp] (runoff0) [OWLSFlow] (testCounter) (catchmentID) (latitude) (longitude) (turbidity) (trueNCells) (trueNEdges) (trueNNodes) (*cellIdxs) (snowSeasonBeginMMDD)	(*getGages) (*getSoils) (*getVegetation) (*getInstantTemp) (getStaticParam) (getDynamicParam) (application) (*getRain) (*getTemp) (*getFlow) (getCloud) (solarRadiation) (getCellSoilCharacters) (getCellVegCharacters)

Appendix I: *List of Objects used in OWLS (Continued)*

OWLS Modules	Object Name (Induced Object)	Descriptions	Features (Additional Features)	Functions (Additional Functions)
			(snowSeasonEndMMDD) (tempRange) (nRainGages) (*rainGage) [OWLSGauge] (nAirTempGages) (*airTempGage) [OWLSGauge] (nSoilOTempGages) (*soilOTempGages) [OWLSGauge] (nSoilBTempGages) (*soilBTempGages) [OWLSGauge] (nSoilCTempGages) (*soilCTempGages) [OWLSGauge] (nStreamGages) (*streamGage) [OWLSGauge] (nSoil) (*soil) [OWLSSoil] (nVeg) (*veg) [OWLSVegetation] (Cmanning) (**rain) [OWLSRain] (**airTemp) [OWLSTemp] (**soilOTemp) [OWLSTemp] (**soilBTemp) [OWLSTemp] (**soilCTemp) [OWLSTemp] (cloud) [OWLSCloud] (**cellSoil) [OWLSSoil] (**cellVeg) [OWLSVegetation] (inputUnit) (outputUnit) (monthlyAirTemp[])	
OWLSFacet	OWLSFacet	object for facet	*parent [OWLSPolyhedron] nNodes *nodeIdxs color [Color] area	draw fill print unitNormalMC unitNormalWC facetColor getFacetArea whichSide intersect reflect
	OWLSCell	cell object for watershed model	(*parent) [OWLSWatershed] (marked) (nEdges) (*edgesIdx) (edgesWeight) (value) (info) [sCellInfo]	(save) (read) (print) (getCellInfo) (getWeight) (nodeInCell) (nodeOnCell) (getCellArea) (averageAspect) (unitNormalMC)

Appendix I: *List of Objects used in OWLS (Continued)*

OWLS Modules	**Object Name (Induced Object)**	**Descriptions**	**Features (Additional Features)**	**Functions (Additional Functions)**
OWLSPoint	OWLSPoint	point object	x y z w	operator + operator - operator *
	OWLSGauge	gauge object for watershed	(name) (dataFile) (nRecords) (startTime) [OWLSTime] (endTime) [OWLSTime] (step)	
	OWLSNode	node object for watershed model	(d1) (d2)	(save) (read)
OWLSLines	OWLSLines	line object for watershed model	nNodes color *nodeIdxs length *parent (OWLSLineBunch]	getLength updateVC draw print
OWLSBiTree Node	OWLSBiTreeNode	node object for binary tree	*pathNode [OWLSPathNode] *parent [OWLSBiTreeNode] *left [OWLSBiTreeNode] *right [OWLSBiTreeNode]	remove
OWLSStream	OWLSStream	stream object for watershed	*parent [OWLSWatershed] *first [OWLSSegment] *current [OWLSSegment]	
OWLSMatric	OWLSMatric	matric object for matrix calculation	nColumes nRows **elms	operator + operator - operator * operator /
OWLSSoil	OWLSSoil	soil object for watershed	name ID porosity nConductivities *moisture0 *conductivity infiltration0 infiltrationC	getConductivity
TransMatrix	TransMatrix	transformation matrix for calculation	elms[][]	operator * translate rotateX rotateY rotateZ mirrorX mirrorY mirrorZ scale
DevPoint	DevPoint	device point object	x y	operator + operator - operator * operator /
OWLSVector	OWLSVector	vector object	x y z	operator + operator - operator * operator % length flatLength normalize

Appendix I: *List of Objects used in OWLS (Continued)*

OWLS Modules	Object Name (Induced Object)	Descriptions	Features (Additional Features)	Functions (Additional Functions)
OWLSPath	OWLSPath	flow path object	*parent [OWLSWatershed] *first [OWLSPathNode] *current [OWLSPathNode] marked size drainArea drainSlopyArea	insertPathNode growPathTree addPath insertPath addChildPath seekPathNodes getPathDensity markNode markPath print save addPathDensity searchNode
OWLSCloud	OWLSCloud	cloud object for watershed model	coverage t1 [OWLSTime] t2 [OWLSTime]	
OWLSMacropore	OWLSMacropore	macropore object for watershed model	radius count	getCrossArea getAllCrossArea getVolume getAllVolume getActiveCount
OWLSFlow	OWLSFlow	flow object for watershed	t1 [OWLSTime] t2 [OWLSTime] volume	discharge
OWLSVegetation	OWLSVegetation	vegetation object for watershed	name vegID coverage LAI0 itRate0 etRate0	seasonRatioLAI getAverageLAI
OWLSenNode	OWLSenNode	edge-node tree's node object	enIdx nChildren *parent [OWLSenNode] **child [OWLSenNode]	removeNode
OWLSenTree	OWLSenTree	edge-node tree object	*root [OWLSenNode] *current [OWLSenNode] *parent [OWLSWatershed]	print save read saveForDraw getSize growBoundaryTree characterized seekParent isAParent defineChildren seekBoundary remove
sCellInfo	sCellInfo	cell information object	slope aspect area eqLength eqWidth eqUpDepth eqDnDepth minY maxY center	

Appendix I: *List of Objects used in OWLS (Continued)*

OWLS Modules	Object Name (Induced Object)	Descriptions	Features (Additional Features)	Functions (Additional Functions)
OWLSFlux	OWLSFlux	flux object	nFluxs *flux	
DATA	DATA	data object for input routine	t (type) itemNumber itemBoolean itemChar itemString	
DevRange	DevRange	Device Range	Min Max	bDoesOverlap bTouching operator |= operator &= operator =
OWLSTime	OWLSTime	time object	julianDay hhmmss yymmdd	getXX getYY getZZ julian2Long toXXYYZZ toHour toJulianDay toLongDate toLongTime getYear getMonth getDate getHour getMinute getSecond operator + operator - operator * operator / operator > operator < operator != operator >= operator <=
OWLSRain	OWLSRain	rain object	RainOrSnow t1 (OWLSTime) t2 (OWLSTime) depth	intensity
OWLSBiTree	OWLSBiTree	binary tree object	*root [OWLSBiTreeNode] *current [OWLSBiTreeNode]	find add
OWLSTemp	OWLSTemp	temperature object	t1 [OWLSTime] t2 [OWLSTime] degreeC	
Color	Color	color object	MAX_INTENSITY R G B	
OWLSEdge	OWLSEdge	edge object	marked length ptIdx1 ptIdx2 cellIdx1 cellIdx2	save read getLength getMarked

Appendix I: *List of Objects used in OWLS (Continued)*

OWLS Modules	Object Name (Induced Object)	Descriptions	Features (Additional Features)	Functions (Additional Functions)
OWLSPathNode	OWLSPathNode	flow path node object	marked nChildren nodeIdx node edgeIdx c1Idx c2Idx length area slopyArea *parent [OWLSPathNode] *child [OWLSPathNode]	operator = save read removePathNode
OWLSSegment	OWLSSegment	segment object	*p [OWLSPathNode] **flow [OWLSFlux] maxTopWidth bottomWidth maxDepth *parent [OWLSSegment] **child [OWLSSegment]	getDepht getWidth getArea getHydraulicRadius
Pigi	Pigi	Platform independent graphics interface	PigiMode mode paintCount currentColor size	setColor getColor setPixel getPixel drawLine fillPolygon beginPaint endPaint selectFont removeFont textSize outText message messageOK messageStop errorExit setTextAttribute
	WinPigi	Pigi for MS Window platform	(*hwnd) (*hdc) (FontRec) (*font)	(initialFont) (getPixelColor)
	MemPigi	Pigi for Memory implimentation	(*data)	(getIndex)
	(FilePigi)	Pigi for File	(*fileName) (frameNumber) (flushFlags)	(getNumberedName) (flush)

Appendix II: *Object Linkages Associated with OWLS*

Main Object	**Internal Linked Objects**	**Inherited Linked Objects**	**External Linked Objects**
WinPigi	Color DevPoint tagLOGFONT	Pigi	Tfont HWND__ HDC__
MemPigi	Color DevPoint	Pigi	Color
FilePigi	Color DevPoint	MemPigi	Color
OWLSLength		OWLSUnit	
OWLSArea		OWLSUnit	
OWLSVolume		OWLSUnit	
OWLSTemperature		OWLSUnit	
OWLSModel	OWLSTime		OWLSHydrology
OWLSPhysicalModel	OWLSTime	OWLSModel	OWLSModel OWLSFlux OWLSSegment OWLSStream OWLSMacropore
OWLSObject	TransMatrix		OWLSObject
OWLSPolygon	OWLSVector Color	OWLSObject	OWLSPoint
OWLSPolyhedron		OWLSObject	OWLSPoint OWLSFacet
OWLSLine	Color	OWLSObject	OWLSPoint
OWLSLineBunch		OWLSObject	OWLSPoint OWLSLines
OWLSContour		OWLSLineBunch	
OWLSText	OWLSVector OWLSPoint TextAttribute	OWLSObject	
OWLSWatershed		OWLSObject	OWLSNode OWLSEdge OWLSCell OWLSenTree OWLSBiTree OWLSPath OWLSPathNode

Appendix II: *Object Linkages Associated with OWLS (Continued)*

Main Object	**Internal Linked Objects**	**Inherited Linked Objects**	**External Linked Objects**
OWLSHydrology	OWLSFlow OWLSCloud	OWLSWatershed	OWLSRain OWLSTemp OWLSGauge OWLSSoil OWLSVegetation
OWLSIrrObject		OWLSPolyhedron	
OWLSTriTopo		OWLSPolyhedron	
OWLSRecTopo		OWLSPolyhedron	
OWLSCone		OWLSPolyhedron	
OWLSInerBox		OWLSPolyhedron	
OWLSInnerBox		OWLSPolyhedron	
OWLSCube		OWLSPolyhedron	
OWLSEgg		OWLSPolyhedron	
OWLSCylinder		OWLSPolyhedron	
OWLSPyramid		OWLSPolyhedron	
OWLSFacet	Color		OWLSPolyhedron
OWLSCell	sCellInfo	OWLSFacet	OWLSWatershed
OWLSGauge	OWLSTime	OWLSPoint	
OWLSNode		OWLSPoint	
OWLSLines	Color		OWLSLineBunch
OWLSBiTreeNode			OWLSPathNode OWLSBiTreeNode
OWLSStream			OWLSWatershed OWLSSegment OWLSStream
OWLSPath			OWLSWatershed OWLSPathNode
OWLSCloud	OWLSTime		OWLSCloud
OWLSFlow	OWLSTime		
OWLSenNode	OWLSen		OWLSenNode
OWLSenTree			OWLSenNode OWLSWatershed
TextAttribute	Color		
sCellInfo	OWLSNode		
OWLSRain	OWLSTime		
OWLSBiTree			OWLSBiTreeNode
OWLSTemp	OWLSTime		
OWLSEdge			OWLSWatershed
OWLSPathNode	OWLSNode		OWLSPathNode
OWLSSegment			OWLSPathNode OWLSFlux OWLSSegment

Appendix III: User's Manual

USER'S MANUAL

Object Watershed Link Simulation (OWLS)

(Version 1.0)

by Huaisheng Chen

December 10, 1996

TABLE OF CONTENTS

USER'S MANUAL

Object Watershed Link Simulation (OWLS)

(Version 1.0)

1. About the OWLS model

1.1. *What?*

1.1.1. *What is an Object?*

An object is an entity having some specific properties and certain types of functions; Virtually everything in the real world is an object. In the field of computer science, an object is defined as a container of data type (or class in C++) which has some specific properties or features (or data members) and which also has certain types of functions (also called member functions).

1.1.2. *What is a Watershed?*

A watershed is an area that drains to a common point or outlet.

1.1.3. *What is a Link?*

A link (or Linkage) is a direct relation between two objects. There are three types of linkages:

(1) Internal linkage: One object is included within another. For example, the object OWLSFlow includes object OWLSTime. OWLSTime becomes one of the features of OWLSFlow, which is accordant with the natural flow. Features of OWLSTime are automatically passed to OWLSFlow (in-to-out).

(2) Inherited linkage: One object is inherited from another object. For example, the object OWLSPhysicalModel is the inherited object from OWLSModel; and object OWLSWatershed is the inherited object from OWLSPolyhedron. Parameters (features) of OWLSPolyhedron automatically passes to OWLSWatershed (parent-to-children).

(3) External linkage: One object contains a member acting as a gateway to another object. Such member is also called *pointer*, or External link in OWLS' term. In such cases, some functions of the object can easily use parameters from another object through this link without complex analytic procedures. External linkage not only makes the cell-to-cell connection possible and is relatively easy, but also assists in the connection of flow paths and stream networks.

1.1.4. *What is Simulation*

A Simulation is a process to recreate or approximate a natural process. There two types of simulations: Physical and Mathematical. The OWLS model utilizes mathematical simulation to represent the hydrologic process.

1.1.5. *What is OWLS?*

OWLS is the abbreviation from the Object Watershed Link Simulation. It is a mathematical model simulating the hydrologic processes of small forested watershed. It is organized as follows: Every component in a watershed and its hydrologic processes is considered as an object. Relations among the components are the Linkages. Flow in the watershed is transported through these Linkages. Thus, the OWLS model is basically a collection of computer programs calculating these linkages for each object.

In the OWLS program, the watershed is constructed as follows: Starting from the basic components (objects) of a watershed: points, lines and cells, establish the properties that these objects have (e.g. elevation, length, slope, soil, vegetation ...) and the types of functions they perform (e.g., infiltration, surface flow ...); Then find the relations between these objects (linkage) to form the bigger objects of the watershed (e.g., flow path, stream network, canopy, surface, soil, macropore pipes ...); Finally establish how these objects operate together (linkage) to reflect watershed behavior (e.g. stream flow, stream chemistry).

1.1.6. *What is the information that the OWLS model provides?*

The information that the OWLS model can provide includes: Stream flow at the watershed outlet, stream flow at each stream segment of the channel, soil moisture conditions in the soil of each cells and the whole watershed and much more. Check out section 3 in this manual for details.

1.1.7. *What program language is the OWLS model used?*

The language that the OWLS model used is C++, with Borland Windows Classes for PC Windows interface. While OWLS can be run under both Windows 3.x or Windows 95, it seems to have better performance in Windows 95. The majority of the OWLS model, including the hydrologic, visualization and data processing models are coded using standard C++ program, which can be easily transplanted into different operation systems (like UNIX) for faster run or larger data sets. The Windows User Interface portion is the only portion that is platform dependent and should be re-coded for a different operation system.

1.2. *Who?*

1.2.1. *Who contributed to the development of the OWLS model?*

The idea of the OWLS model was presented in the research proposal for the Ph.D. thesis by Mr. Huaisheng Chen in late 1994, a student in the Department of Forest Engineering in Oregon State University. After obtaining feedback from his major professor Dr. Robert L. Beschta and his committee (Dr. Marvin Pyles, Dr. Chaur-Fong Chen, Dr. Wayne C. Huber, Dr. Peter C. Klingeman and Dr. Parker J. Wigington). The concept of an Object Watershed Link type of simulation began to emerge.

As part of the thesis research, Mr. Chen carried out all the development tasks. The development of the OWLS model includes object subdivision, object design, modular design, watershed layout design, data entry, data processing, program coding, program debugging, program testing, literature research and so forth. In the course of the development of the OWLS model, there were several technical difficulties involved: the watershed object automatic delineation, flow calculation for irregular cells, flow routing model, macropore flow model, program error control and program debugging. In the course of choosing an experimental watershed which had sufficient data available for the OWLS model, Dr. Wigington of the US EPA Corvallis Lab and Dr. Steve Norton from the Department of Geology in University of Maine were instrumental in providing basic data for the Bear Brook watershed in Maine.

1.2.2. *Who may be interested in the OWLS model?*

Professional people who study watershed problems regarding water quantity, quality in soil, surface and stream channels may be interested in the OWLS model.

1.2.3. *Who is able to run the OWLS model?*

The current version of the OWLS model is not user-friendly. Thus, it requires professional knowledge about watershed hydrology, geography, meteorology, soil physics. Knowledge about geometry, spread-sheet usage, and basic knowledge about computer programming are also required. If the OWLS model is to be applied to a different watershed, a potential user needs to read through this manual and follow step-by-step instructions. Knowledge about the C++ programming will greatly help in applying this model, and even help in the development of new codes to enhance the OWLS or move the OWLS to different computer platform.

The OWLS model can be applied to small forested watersheds. The size of the watershed is ideally 5 to 1000 hectares when OWLS is used on a personal computer. But since the OWLS model has not been applied to any other watershed and the data handle capacity of PC has been increased dramatically, the range of watershed is only a guess.

2. Inputs for the OWLS model

2.1. *Required Data*

(1) DEM data

Digital Elevation Model data provide quantitative characterizations of watershed elevations and area. Typically for small watersheds, an array of data representing many sampled points (or surveyed points) of watershed elevation related to a local reference point are sufficient for the OWLS model. For experimental watersheds like BBWM, a detailed land survey produced elevation data for points on the watershed located 15 meters apart. In some areas, like the riparian zone, additional measurements have been undertaken to reflect detailed variation of watershed topography. For watersheds without special survey data, a DEM database provided by the USGS can be adopted.

Topographical data is required for the OWLS program. By organizing the data into a certain format, the data processing modular will convert the elevation data into a vectorized database assuming a triangular meshed watershed.

(2) Precipitation Data

Precipitation data are the depth of rainfall or snowfall within each time interval (e.g. 1 hour). Currently, almost all experimental watersheds have precipitation data available. For areas without precipitation data, using nearby precipitation records nay be a reasonable approximation. The OWLS model allows more than one precipitation station for a watershed. When there are at least three rain gauges available in (around) a watershed, the model has a precipitation module that uses a spatial linear interpolation technique to distribute rain gauge data into different cells of the watershed.

(3) Geographical Coordinators

The geographical location of the watershed's center needs be provided by a user. At the BBWM, the latitude is 44.87 (in degrees North) and the longitude is 68.1 (in degrees West). Geographical location is used by the solar radiation model in OWLS, for the purpose of calculating potential evapotranspiration.

(4) Air Temperature Data

In order to estimate the snowfall and snowmelting, the model uses air temperature as the criterion to determine if it is snow or rainfall, and also if snowmelt is occurring. Air temperature is also used to calculate the potential evapotranspiration from vegetation and soil. Therefore, air temperature is also a required variable for the watershed model.

For an experimental watershed like the BBWM, there may be more than two air temperature gauges. In this case, a vertical air temperature model was used to interpolate the elevation difference of air temperature distribution. For an area with fewer air temperature records (like only has daily records of max-min data), the OWLS model has a built-in daily air temperature interpolation model to approximate the daily air temperature fluctuation, so that daily fluctuations in hydrologic processes can be simulated for the watershed (especially day from night). In addition to the observed hourly air temperature or daily air temperature characteristic values, an OWLS user also need to provide mean monthly air temperature and the date for the beginning and ending of the snow season. In case of missing data or no records, the OWLS model will compute an approximation of air temperature using monthly and seasonal data with its built-in air temperature model.

(5) Soil Survey Data

Soil survey data include two major components: the depth of the soil and the type of the soil. The OWLS model is designed to have information for two-layered soil depth for the watershed, representing the vertical differences of soil properties. The model also accepts as many as soil types as are available on the watershed including representative soil properties. For areas that lack soil data, these two properties can be simplified into two numbers: the mean watershed soil depth and the average soil type. In this case, the watershed's soil is assumed homogeneous. In this application to the BBWM, since we have detailed soils data, soil depth was determined from the surveyed soil depth data. There are two layers of soil in the BBWM, but in the OWLS model, the soil column is considered as one-layered with uneven physical properties in a vertical dimension. The depth of the soil in the model is more hydrologically sensitive than the soil definition itself. The OWLS model provides a weighted parameter for a user to determine the depth of the two soil layers.

2.2. *Optional Data*

(1) Streamflow Data

Streamflow data is required only when the OWLS model is used for parameter calibration and validation. Measured streamflow is considered as an objective measure of watershed response. If modeled flow patterns generally match the observed streamflow, then the model is considered calibrated. The values of the model parameters are then considered calibrated parameters. To prove the model is a good simulator of hydrological processes, streamflow which was not used for calibration is then simulated; this process represents model validation. If the modeled results still reproduce measured flows, then the model can be considered to represent the hydrologic processes of that watershed.

Streamflow data is not need if the model is used to simply simulate the hydrologic processes of a watershed. In that case, the streamflow produced from the model will represent uncalibrated output.

(2) Soil Infiltration Data

For some watersheds, relations between soil infiltration rate and the soil moisture for each type of soil may be available field data. This information is very valuable and can dramatically reduce the number of parameters that need to be calibrated in the model. If this information is available, it can be simply stored into a set of defined files. The Horton model is then disabled so that the OWLS model will use the corresponding soil infiltration rate for various soil moisture condition. When detailed infiltration data are not available, the OWLS mode uses the modified Horton's infiltration model. In the BBWM watershed, we used modified Horton's model.

(3) Macropore Pipe Data

Little is known regarding the macropore system of forest soils. However, the OWLS model has a built-in macropore pipe model developed using some major assumptions from pipe flow theory. Information regarding the following relationship will hopefully be available in the future and could be used to replace the currently assumed parameters:

a. relational data between catchment area and macropore pipe radius;
b. relational data between catchment area and numbers of macropore pipes;

For the BBWM, macropore pipe data were not available, thus, assumed parameters were used to calculate the macropore pipe property in a cell.

(4) Channel Geometry Data

Data for channel geometry can be obtained directly from the field observation and measurement. This data can be used directly by the OWLS model to replace its built-in channel geometric model. Relational data for channel geometry include:

a. Drainage area and stream depth;
b. Drainage area and stream top width;
c. Drainage area and stream bottom width;

For the BBWM, no channel geometry data were collected; Thus, we used the built-in model.

2.3. *System Parameter Data*

There are two types of system parameters for the OWLS model: system control parameters and system model parameters. System control parameters are those used to determine the performance of the model, e.g., English vs. SI unit, calculation time step, etc. These parameter are chosen by the user and do not require calibration. System model parameters are those parameters required by the watershed model itself and directly involve the simulation of watershed processes, e.g. infiltration coefficient, hydraulic conductivity. While many of these parameters have a physical interpretation and a certain range of values, their performance within a watershed model still needs to be determined. Therefore, they usually need to be calibrated.

2.3.1. System Control Parameters

(1) Unit Usage:

In the United States, both English and SI units are used in scientific and professional disciplines. To overcome this difficulty, the OWLS model allows a user to choose either set of units for input or output.

(2) Time Domain:

A definition of the time range from start to finish should be provided before running the model, including: start date (yymmdd), start time (hhmmss) , end date, end time, time step for calculation, and the time interval for saving results. By providing the time domain, the OWLS model is capable of simulating the hydrological process for a defined period and also interpolate or accumulate input information like precipitation and air temperature at a desired interval. Also, by specifying the interval for saving results, only the desired results are saved, which is beneficial for optimizing calculations and computer resources.

(3) Output Unit Option:

The OWLS model provides an option for output units: in terms of "depth" or "volume". When the useDepthOutput is set to TRUE, the output of streamflow, evapotranspiration, etc., is illustrated in depth over the whole watershed area per unit time. This is especially useful when we doing a water balance. When the useDepthOutput is set to FALSE, the output unit is volume per unit time, such as flow in cubic meters per second.

(4) File Name Definitions:

The OWLS model needs to work with relatively large amounts of data which are stored amongst a variety of files. These files are defined as control parameters so that the OWLS program can access them. Although control parameters are fixed, their values are determined by a user. File format for each parameters are also fixed. Some of the control parameters may remain undefined when data is not available for them. The following is an example:

LAItoInterceptFileName = "lai2intc.dat"

Where LAItoInterceptFileName is one of the control parameters and “lai2intc.dat” is the value of this parameter which is a file name. The file name for LAItoInterceptFileName stores the relational data between the Leave Area Index (LAI) and the Intercepting ratio (R_{LAI}). The intercepting ratio is the ratio between the interception storage capacity of the forest having a specified LAI at a certain time of the year in relation to the maximum interception storage capacity. As LAI is a seasonal factor, interception storage capacity of a forest will also change accordingly by adjusting the interception ratio. The file format of the lai2intc.dat is as follows:

```
// FILENAME: LAI2INTC.DAT
// Data file for relations between
// available LAI (LAI * coverage) and
// interception capacity ratio (R)
// INTC = INTC0 * R(LAI(t))
// LAI -- INTC
0  0
1  0.1
2  0.2
3  0.3
.. ...
8  0.8
9  0.9
10 1.0
```

The lines heading with double slash "//" are comment lines (the standard comment marking in C program). Comment lines can be any where in the data file. The lines heading with numbers are the data lines. The first value in each line represent the LAI value and the second value corresponds to the interception capacity ratio. All relational data follow the above format so that values can be selected and used for interpolation. For example, in the above data, if the LAI is 8.5, then the OWLS model will pick the values from *LAI = 8* (which is 0.8) and *LAI = 9* (which is 0.9) and then solve for a value between is 0.85. While default values within the OWLS model can be used, if field information provides a relationship between these two parameters, the user can directly modify this file so that we have more a accurate model for the watershed.

File name definitions for the OWLS hydrologic model are as follows:

[1]. LAItoInterceptFileName: This file that stores relational data between LAI and Interception ratio.

[2]. LAItoETFileName: This file that stores relational data between LAI and ET ratio from canopy.

[3]. drainArea2PipeRadius: This file that stores relational data between soil depth and macropore pipe radius.

[4]. area2PipeCount: This file that stores relational data between upper drainage area and macropore pipe count.

[5]. area2Depth : This file that stores relational data between drainage area and stream depth.

[6]. area2TopWidth: This file that stores relational data between drainage area and top width of a stream cross-section.

[7]. area2BotWidth: This file that stores relational data between drainage area and bottom width of a stream cross-section.

[8]. paramFileName: Interruption Protection file for model parameters. This file is in binary format. The OWLS model is designed to protect results from interruption. In case of power loss or manual interruption, the results will not be lost. The simulation can be continued from the last saving point before the loss of power instead of to starting again from the very beginning.

[9]. fluxFileName: Interruption protection file of flux map from last calculation step. This file is in binary format.

[10]. channelFileName: This data file stores simulated channel information, including the depth of water, width of channel, discharge in the channel and all details of a channel segment. This file is primarily for graphical visualization. The visualization model can read this file and provide visual output for a user. This file is in binary format.

[11]. cellFileName: This data file stores simulated information for watershed cells, including the soil moisture, the flux occurring within the cell and water depths (surface, snow cumulation, soil, macropore etc.)

[12]. sumFluxFileName: This data file stores series of flow data at the watershed outlet. Data are saved in a pre-defined saving step. The visualization model within the OWLS model can read the data file and output to the user.

[13]. counterFileName: This counter file maintains the records that have been simulated.

[14]. textOutputFileName: This is comprehensive data file in text format showing rainfall, observed flow, simulated flow, canopy intercepted water, surface water and flow, soil water, soil flow, soil moisture, macropore water, macropore flow, and so on for the stream outlet. It is intended to contain summary information about the simulation. It can be imported into a spreadsheet and used for simple graphical analysis.

[15]. riverOutputFileName: This is a text file for storing simulated streamflow, including flows for each stream segment for each time saving step.

(5) Switch Parameters

There are built-in functions within the OWLS model that allow a user to switch parameters depending upon their availability as model inputs. These parameters are usually presented as values like TRUE or FALSE. Some of them use digital integers for choosing among more than two options. Switch parameters do not need calibration. The following are the switch parameters used in the OWLS model:

[1]. useHorton: when TRUE, the modified Horton's equation will be used for surface infiltration calculation. When FALSE, relational data will be obtained from the files for infiltration rate under different soil moisture conditions.

[2]. useDirectInputs: This switch parameter is used to determine whether direct precipitation onto the stream surface will be considered as part of the watershed water inputs. The default value is FALSE since the adding of direct inputs may cause a slightly imbalance of water circle in the watershed. Although the OWLS model simulates the dynamic stream water

surface during a precipitation event, it is not easy to deduct this portion of stream water surface from the nearby land unit, e.g. if the watershed area is 100,000 m^2 and the stream water surface area is 1,000 m^2. This 1,000 m^2 could not be deducted from the watershed area. Thus, if we add direct rainfall onto the stream, there will be actually 100,000+1,000 m^2 surface area receiving water from the sky, which is not correct. So the default for this model is to neglect the direct rainfall instead of amplify its effects.

[3] iterativeErr: This is the control factor for Newton's iterations in the kinematic wave simulations for both surface water routing and stream flow routing procedures. It is an error criterion. The smaller it is, the more precise the result will be. On the other hand, a small error term increases the time/steps needed for calculations. While it can greatly affect run times, it will not significantly affect simulation results. In the BBWM modeling, the iterativeErr is taken the value 0.01.

[4] maxIterations: This is another control factor for Newton's iterations to avoid unexpected looping. This parameter sets an upper limit for the number of iterations. When its value is relatively large (like 100), it will not affect the simulation result. But if set as low as 1 or 2, the result of simulation will be significantly altered.

(6) Visualization Control

Parameters to control the screen output include color, view specification, time range etc.

2.3.2. *System Model Parameters*

Parameters that directly drive the hydrological process of the watershed are system model parameters. Many of these parameters are not physically known, especially at a watershed scale. Thus, they need calibrations so that the model can fit to local watershed. There are 37 model parameters used in OWLS:

[1]. infiltration_k: the constant in 1/hr for Horton's equation;

[2]. infiltration_a: the index for soil-moisture and Horton time function

[3]. conductivityAdjust: an adjustment factor for soil conductivity 1.0 means no change, 1.1 means a 10% increase. This parameter is set for calibration purposes: since soil conductivity is not only a function of soil type but also soil moisture content. The OWLS model

obtains relational data between soil hydraulic conductivity and the soil moisture contained in files or from user supplied data. In actuality, the hydraulic conductivity of a watershed may have a systematic offset from these values. Therefore, a conductivityAdjust parameter is used to create this offset as a bridge between laboratory data and field conditions.

[4]. infiltration0Adjust: similar to the conductivityAdjust, different types of soil have different maximum infiltration rates (which most likely come from lab experiments). This parameter allows an offset to adjust laboratory results to fit field situations.

[5]. infiltrationCAdjust: an adjustment factor for Horton Minimum infiltration rate.

[6]. snowMelt_Df: a snowmelt degree factor, in inches of water equivalent per hour per degree F.

[7]. snowMelt_Tb: the base air temperature above which snowmelt can occur.

[8]. ET_a: the "a" constant for potential evapotranspiration (PET).

[9]. ET_b: the "b" constant for PET.

[10]. underCanopyETConstant: the ratio between water evaporation on the soil surface water and PET.

[11]. soilETConstant: the ratio between soil ET and PET.

[12]. roughness: the surface roughness (Manning's coefficients) for watershed surfaces.

[13]. layerWeight1 and layerWeight2: the OWLS model utilizes input data for two soil layers. But hydrologically there is a depth of soil that may not fit exactly into either the first or second layer. It may occur somewhere in between. Thus, the equation for the depth of soil is weighted by the depth of both layer, which means the layerWeight1 (weight for the 1st layer) and layerWeight2 (weight for the 2nd layer of soil) should be sum to 1.0. In the BBWM watershed, based on hydrograph responses, we chose the bottom layer as the boundary of the soil depth, or layerWeight1 as 0.0 and layerWeight2 as 1.0.

[14]. minDiameter: the parameter used by the Macropore flow model. It is the minimum diameter that qualifies a soil tunnel pipe for being a macropore pipe and, which allows the movement of gravitational water.

[15]. pipeRatio: this parameter is used to adjust the average macropore pipe radius. The parameter is based on the assumption that in a soil column, the macropore pipes are not evenly distributed vertically. Larger pipes may be found more on the top than at the bottom of the soil column. Therefore, when more macropores are saturated by the water, there the average effective macropore radius will be larger. The pipeRatio parameter is used to adjust the average effective macropore diameter. There is a reason to introduce this parameter: On the hydrograph, the flow recession curve is usually more gradual than ordinary soil and pipe flow

can simulate. This parameter will be able to speed up the flow peak and slow down the flow recessions.

[16]. radiusA, radiusB and radiusC: a parameter for the macropore pipe radius equation;

[17]. countA, countB, countC, countD and countE: parameters for the macropore pipe count equation;

[18]. surfaceMacroporeConst: a ratio between water flow into macropores from the surface and that to the soil (infiltration). Since the exact equation for the process by which surface water flows into the soil macropore system is unknown, a simplified assumption is that the amount of water flux from surface to the macropore pipe system is proportional to the soil infiltration flux. Under this assumption, surfaceMacroporeConst is used to calculate the amount of water flow into the macropore system from the surface.

[19]. soilMacroporeConst: the constant for soil water flux to the macropore pipe system. Similarly, the water flux from the soil to a macropore pipe system are unknown. This constant is based on the assumption that the higher soil water tables, the greater the flux from the soil to the macropore pipes. Currently, the macropore model only allows inputs from surface water and soil water, but does not allow the reverse process (i.e., water moving from macropore pipe to the soil or non-stream surface).

[20]. frictionCoeff: the friction coefficient for macropore pipes. For an ordinary system of non-macropore pipes, this parameter is ranged from 0.04 to 0.06 for Re > 4000 turbulent flow. But for hydrograph simulations of the BBWM watershed, the value was 350, otherwise the simulated hydrograph was always to steep up and down (i.e., water moves too fast through the macropore system). This result indicates that natural macropore system may have more friction than associated with simple pipes.

[21] widthTopConstant and widthTopPow: the parameters used to calculate the channel top width in relation to catchment area.

[22] widthBotConstant and widthBotPow: the parameters used to calculate the channel bottom width in relation to catchment area.

[23] depthConstant and depthPow: the parameters used to calculate the channel depth in relation to catchment area;

[24] soilMoisture: the initial soil moisture condition. This value will only affect the initial stage of simulated flows and will become less influential as calculations proceed.

[25] initialStreamDepthRatio: the initial stream water depth condition, as a ratio to its maximum physical depth. This parameter is relatively unimportant. Like soilMoisture, it can

only affect the initial stages of simulated flows and its effect disappears with continued calculations.

[26] soilWaterSupplyC1: the maximum rate of water supply that a soil can offer to the vegetation canopy.

[27] soilWaterSupplyC2: the coefficient for the equation to calculate soil water supply for the canopy. This parameter will affect the curve of soil-moisture to water supply. When it is zero, than the curve become a straight line which means they are linearly related. From a conceptual basis, the value is in the range between 0 and 0.1618 of the soilWaterSupplyC1. It represents the maximum offset from the linear straight line.

3. Outputs from the OWLS model

The OWLS model generates two different kind of outputs: 2-D data and 3-D data.

3.1. *Two-Dimensional Data*

Two-Dimensional Data include stream flow at the watershed outlet, total water storage for different components of the watershed, vertical flux of the watershed, average soil moisture content and watershed air temperature. All of these data correspond to a certain time step of simulation. They are stored in two formats: binary format and text format.

The binary-formatted file is defined by the parameter sumFluxFileName. It has repeating data blocks with the structure shown in Table M - 1.

Table M - 1. Data structure of sumFluxFileName file

Size of Offset	Numbers of Offset	Value Represented
int	1	Series Number
int	1	The number of steps for the current simulation
float	1	Simulated discharge (m^3/hr)
float	1	Observed discharge (m^3/hr)
double	1	Rainfall depth (m)
double	1	ET depth (m)
double	1	Stream water volume (m^3)
double	1	Stream Surface Flow Component (m^3/timestep)
double	1	Stream Subsurface Flow Component (m^3/timestep)
double	1	Stream Macropore Pipe Flow Component (m^3/timestep)
double	1	Intercepted Water Depth (m)
double	1	Surface Water Depth (m)
double	1	Soil Water Depth (m)
double	1	Macropore Pipe Water Volume (m^3)
double	1	Relative Soil Moisture Content
double	1	Soil Conductivity (m/hr)
double	1	Basin Averaged Air Temperature (degree C)

The text-formatted file is defined by the parameter textOutputFileName. The text file contains two portions: parameter portion and data portion. The parameters portion describe the major parameters that used for the simulation. The data portion are simulated results which has the format as shown in Table M - 2.

Table M - 2. Data format for textOutputFileName file

```
Row------2------3----------4---------5----------6---------7---------8
No.   Date   Time  julianDay  SimuFlow   MeasFlow  SimuFlow  Rainfall
--  yymmdd hhmmss       days      m3/s       m3/s        m3        m3
  0 901103      0 33180.0000     0.000      0.002       0.0       0.0
  1 901103  10000 33180.0417     0.001      0.002       2.4       0.0
  2 901103  20000 33180.0833     0.001      0.002       4.0       0.0
  3 901103  30000 33180.1250     0.001      0.002       4.4       0.0
 ..     ..     ..         ..        ..         ..        ..        ..

--9---------10--------11--------12--------13--------14--------15--------16
 ET RiverWater  RSurFlow RSoilFlow RPipeFlow     Canopy   Surface SoilWater
 m3         m3        m3        m3        m3        m3        m3        m3
0.0          1         0         6         0         0         0      6288
0.0          1         0         6         0         0         0      6276
0.0          2         0         6         1         0         0      6264
0.0          2         0         6         1         0         0      6252
 ..         ..        ..        ..        ..        ..        ..        ..

-------17--------18--------19--------20
PipeWater soilMoist soilCondu basinTemp
       m3   percent     mm/hr     deg.C
        5     9.872   208.330     5.205
       10     9.922   208.330     5.391
       15     9.938   208.330     5.749
       19     9.939   208.330     6.256
       ..        ..        ..        ..
```

3.1.1. *Stream Flow*

Streamflow for the watershed outlet has three columns in the output file: columns 6, 7 and 11. Columns 6 and 7 are simulated and observed discharge at the watershed outlet respectively. Column 11 is the simulated flow in volume or depth associated with a particular time step. It can be used to compare with precipitation (also in volume or depth) in column 8.

3.1.2. *Water Storage*

The water storage of a watershed includes: canopy water storage (column 14), surface water storage (column 15), soil water storage (column 16), macropore water storage (column 17), and channel water storage (column 10).

3.1.3. *Vertical Flux*

The vertical flux, as output, includes the simulated total evapotranspiration (column 9) and the calculated watershed average precipitation (column 8).

3.1.4. *Soil Moisture Content*

As calculated from individual cells, the soil moisture content (column 18) is the watershed averaged relative soil moisture.

3.1.5. *Temperature*

The air temperature (column 20) in the watershed is simulated using the Air Temperature Extension Model. The results show whether it is rain or snow.

3.1.6. *Conductivity*

The averaged value of soil hydraulic conductivity (column 19) is calculated as a function of soil moisture content.

3.2. *Three-Dimensional Data*

All three-dimensional output data are stored in files with binary format to reduce size and insure fast access. 3-D output data from the OWLS model includes:

3.2.1. *Topographical Output*

3.2.1.1. Stream Segment

The watershed stream segments are stored in a file defined by parameter segmentFileName. This file is in binary format and stored as follows:

Table M - 3. Data format for segmentFileName file.

Size of Offset		**Numbers of Offset**		**Value Represented**	
integer		1		Total number of stream segments (nSegments)	
integer		1		Numbers of Steps of the simulated data	
OWLSTime	double long long	1	1 1 1	Time Information:	Julian Day; hhmmss; yymmdd;
float		nSegments		velocity of the segment flow (m/s)	
float		nSegment		discharge of the segment flow (m^3/hr)	
float		nSegments		segment water width (m)	
....		....		....	
float		nSegments		velocity of the segment flow (m/s)	
float		nSegment		discharge of the segment flow (m^3/hr)	
float		nSegments		segment water width (m)	

3.2.1.2. Stream Network

The watershed stream network is saved in the file defined by parameter basinStreamTreeFileName. The file is in binary format and stored in a edge-node tree structure. The size of the stream network tree may vary. It needs to be retrieved by a recursive function (Table M - 4).

This recursive function creates a stream tree while reading data from the file defined by the basinStreamTreeFileName. Two components in the modular: OWLSenNode and nChildren which is an integer for the number of

Table M - 4. Recursive function

```
void OWLSenTree::read(FILE *file)
{
  if (root == NULL)
  {
    root = new OWLSenNode();
    current = root;
  }
  OWLSenNode * old;
  old = current;
  int i;
  fread(&old->enIdx,
    sizeof(OWLSen), 1, file);
  fread(&old->nChildren,
    sizeof(int), 1, file);
  if (old->nChildren)
  {
    old->child = new
      OWLSenNode*[old->nChildren];
    for (i = 0;
      i < old->nChildren; i++)
    {
      old->child[i] = new
        OWLSenNode();
      current = old->child[i];
      read(file);
    }
  };
  current = old;
};
```

children that the current tree node has. If the number of children is not zero, the function continues to move on to each child until an additional cannot be found.

3.2.1.3. Watershed Cells, Boundary and Flowpath

All topographic data of a watershed are stored in a file defined by the parameter newBasinFileName. The file is in binary format (Table M - 5).

Table M -5. Format of newBasinFileName file

Size of Offset		**Numbers of Offset**		**Value Represented**	
integer		1		Total number of nodes (nNodes)	
integer		1		Total number of edges (nEdges)	
integer		1		Total number of cells (nCells)	
OWLSNode	float	nNodes	1	Node Data:	value on x;
	float		1		value on y;
	float		1		value on x;
	float		1		value w,=1;
	float		1		d1, soildepth1;
	float		1		d2, soildepth2;
OWLSEdge	integer	nEdges	1	Edge Data:	Marker in/out;
	float		1		Length;
	integer		1		1st node index;
	integer		1		2nd node index;
	integer		1		1st neighbor cell;
	integer		1		2nd neighbor cell;
OWLSCell	integer	nCells	1	Cell Data:	Marker in/out;
	integer		1		nEdges;
	float		1		Cell value;
	sCellInfo		1		Cell Information;
	integer		nEdges		Edge Index;
	float		nEdges		Edge Weight;
	integer		nEdges		Node Indices;
	Color		1		Cell Color;
	float		1		Cell Area;
integer		1		Watershed outlet node index	
float		1		Watershed Slopy Area	
float		1		Watershed Area	
OWLSEdge	integer	nEdges	1	Edge Data:	Marker in/out;
	float		1		Length;
	integer		1		1st node index;
	integer		1		2nd node index;
	integer		1		1st neighbor cell;
	integer		1		2nd neighbor cell;

Table M -5. Format of newBasinFileName file (continued)

Size of Offset		**Numbers of Offset**		**Value Represented**	
OWLSEdge	integer	nEdges	1	Edge Data:	Marker in/out;
	float		1		Length;
	integer		1		1st node index;
	integer		1		2nd node index;
	integer		1		1st neighbor cell;
	integer		1		2nd neighbor cell;
OWLSenTree	OWLSenNode integer	(size will be determined by the recursive modular)		Watershed Boundary Data, stored in a tree structure. Size of this block may varied. Need to be retrieved by the recursive function as addressed aboved.	
OWLSenTree	OWLSenNode integer	(size will be determined by the recursive modular)		Watershed Flowpath Tree stored in a tree structure. Size of this block may varied.Need to be retrieved by an recursive function as addressed aboved.	

3.2.2. Hydrologic Output

3.2.2.1. Flows for Stream Segments

There are two files for 3-D flows: a text version and a binary version. The text version is stored in the file defined by parameter riverOutputFileName which has the format as shown in Table M -6.

Table M -6. Format of riverOutputFileName file

```
Stream Flow Rate in cub.m/s for each segment
Total Number of Segments = 29
      segment ID :      0       1       2       3       4       5       6       ..      28
segment RiverMile:    0.0    72.6   112.3   145.0   147.6   209.2   226.4      ..   259.9
 Number JulianDay
      0 32629.0000  0.000   0.000   0.000   0.000   0.000   0.000   0.000      ..   0.000
      1 32629.0417  0.001   0.001   0.000   0.000   0.000   0.000   0.000      ..   0.000
      2 32629.0833  0.001   0.001   0.001   0.000   0.000   0.000   0.000      ..   0.000
      3 32629.1250  0.001   0.001   0.001   0.000   0.000   0.000   0.000      ..   0.000
     ..  ..........  .....   .....   .....   .....   .....   .....   .....     ..   .....
    245 32639.2083  0.011   0.011   0.010   0.007   0.007   0.006   0.006      ..   0.002
    246 32639.2500  0.017   0.017   0.015   0.011   0.010   0.008   0.008      ..   0.002
```

The first 2 rows are information about the units of flow and the total number of stream segment. The 3rd row is the stream segment ID. The 4th row is the segment cross-section river-mile, which is the distance to the stream outlet. Its units, when flow is in SI unit, is in meters. Starting from row 6, each row represents a simulated flow results for each stream segment. The 3rd column is flow from the stream outlet and columns after that are simulated flows for each segment.

The binary version of the segment flows is stored in the file defined by the parameter channelFileName, which has the repeating blocks with the structure shown in Table M - 7.

Table M - 7. Format of channelFileName file

Size of Offset		**Numbers of Offset**		**Value Represented**
int		1		The number of stream segments in the watershed stream tree: nSegments
int		1		The number of steps for the current simulation
OWLSTime	double	1	1	The time for the current simulation: Julian Day, HHMMSS, YYMMDD.
	long		1	
	long		1	
float		nSegments		The velocity for each stream segment
float		nSegments		The discharge for each stream segment
float		nSegments		The width of each stream segment (channel width)

3.2.2.2. Water Storage for Watershed Cells

The file for water storage for watershed cells is defined by parameter cellFileName. It is in binary format and has repeating blocks with the structure as shown in Table M - 8.

Table M - 8. Format of cellFileName file

Size of Offset	**Numbers of Offset**	**Value Represented**
int	1	The number of cells in the watershed: nCells
int	1	The number of steps for the current simulation
OWLSTime	1	The current time
float	nCells	The relative soil moisture content for each watershed cell.
float	nCells	The intercepted water depth in the canopy for each cells
float	nCells	The surface water depth for each cells
float	nCells	The macropore water volume for each cells
float	nCells	The total water depth for each cells

4. Steps to Run the OWLS model

The OWLS model is a data-demanding model, perhaps it is the common charaterization for physically-based watershed models. There are many steps involved in running the OWLS model assumimg a watershed all the necessary data are available. Steps for running the OWLS model include: raw data processing, parameter assignment, watershed checking, model testing, etc. Following are the step-by-step instructions for running the OWLS model:

4.1. *Run the Data Processing Program*

4.1.1. Check List

Raw data that used by the OWLS model are defined by the parameters in Table M - 9.

Table M - 9. Parameters for data source files

No.	Parameter	In File	Description	Required?
1	gridFileName	owls.ini	Gridded elevation data for the watershed	Yes, used by the data processing model.
2	surveyElevFileName	owls.ini	Watershed topographic survey data	Yes, if not available, use the same one as gridFileName.
3	triangleFileName	owls.ini	Watershed triangular mesh data, based on the survey data points.	No, only when you need a detailed meshing
4	contourFileName	owls.ini	An Arc/Infor asc file for contour data.	No, for graphical visualization only
5	boundaryFileName	owls.ini	An Arc/Infor asc file for boundary data of the study area.	No, for graphical visualization only
6	streamFileName	owls.ini	An Arc/Infor asc file for digitalized stream data of the watershed.	No, for graphical visualization and comparison.
7	soilFileName	owls.ini	An Acr/Info asc file for digitalized soil type data of the watershed.	No, for graphical visualization only.
8	soilDepthFileName	owls.ini	The soil depth data of two layered soil for each sample location	Yes, used by the hydrologic model
9	soilCellFileName	owls.ini	The soil type data for each soil cell	No, currently for graphical visualization only.
10	arrowPointsFileName	owls.ini	The point data for 3-D Directional Arrows	No, used by the visualization model only
11	arrowFacetsFileName	owls.ini	The cell data for 3-D Directional Arrows	No, used by the visualization model only
12	testPointFileName	owls.ini	The point data for a 3-D testing model	No, used by the visulation model only

Table M - 9. Parameters for data source files (continued)

No.	Parameter	In File	Description	Required?
13	testIrrFacetFileName	owls.ini	The cell data for a 3-D irregular polygon testing model	No, used by the visulation model only
14	testRecFacetFileName	owls.ini	The cell data for a 3-D rectangular polygon testing model	No, used by the visulation model only
15	origRainFileName	bbwmdata.ini	A text file contains observed rainfall data for all gages	Yes, used by the data processing model
16	origTempFileName1	bbwmdata.ini	A text file contains observed air temperature ans soil temperature for the 1st temperature gage	Yes, used by the data processing model
17	origTempFileName2	bbwmdata.ini	A text file contains observed air temperature ans soil temperature for the 2nd temperature gage	Yes, used by the data processing model
18	origRunoffFileName	bbwmdata.ini	A text file contains observed stream flow data for all gages	Yes, used by the data processing model
19	rainGageFileName	owlshydr.ini	A text file contains information about all the rain gages	Yes, used by the data processing and hydrologic model
20	cloudFileName	owlshydr.ini	A text file contains cloud coverage data for the watershed	Yes, used by the hydrologic model
21	LAIRatioFileNamePrefix	owlshydr.ini	The prefix of a series of file names, which contain Leaves Area Index ratio for different vegetations in different time of a year.	Yew, used by the hydrologic model
22	airTempGageFileName	owlshydr.ini	A text file contains information about all the air temperature gages	Yes, used by the data processing and hydrologic model
23	soilOTempGageFileName soilBTempGageFileName soilCTempGageFileName	owlshydr.ini	A text file contains information about all the soil O-, B- and C-horizon temperature gages	Yes, used by the data processing, but have not been used by the hydrologic model. Their values will not have any effect on the simulation results
24	soilTypeFileName	owlshydr.ini	A text file contains data for each soil type in the watershed	Yes, used by the hydrologic model
25	vegTypeFileName	bbwmdata.ini	A text file contains data for each vegetation type in the watershed	Yes, used by the hydrologic model
26	streamGageFileName	bbwmdata.ini	A text file contains information about all the stream gages in the watershed	Yes, used by the hydrologic model
27	monthlyAirTempFile	bbwmdata.ini	A text file contains average monthly air temperature data for the watershed	Yes, used by the hydrologic model whenever observed data is missing.
28	physicalModelInitFile	bbwmdata.ini	A text file contains parameters for the OWLS hydrologic model	Yes, used by the hydrologic model
29	LAItoInterceptFileName	(physicalModelInitFile)	A text file contains relational data between LAI and interception capacity ratio.	Yes, used by the hydrologic model

Table M - 9. Parameters for data source files (continued)

No.	Parameter	In File	Description	Required?
30	m2fFilePrefix	(physicalModelInitFile)	A prefix for a series of text files which contain relational data between relative soil moisture content and infiltration ability for all types of soil in the watershed	No, optional for used when available. To substitute the modified Horton's Equation.
31	LAItoETFileName	(physicalModelInitFile)	A text file contains relational data between LAI and ET capacity ratio.	Yes, used by the hydrologic model
32	drainArea2PipeRadius	(physicalModelInitFile)	A text file contains relational data between drainage area and macropore pipe radius	No, optional when available to subsititute the macropore pipe model.
33	area2PipeCount	(physicalModelInitFile)	A text file contains relational data between area of a cell and the numbers of macropore pipes.	No, optional when available to subsititute the macropore pipe model.
34	volume2PipeCount	(physicalModelInitFile)	A text file contains relational data between soil volume and numbers of macropore pipes.	No, optional when available to subsititute the macropore pipe model.
35	area2Depth	(physicalModelInitFile)	A text file contains relational data between the drainage area and the stream cross-section depth	No, optional when available from survey. To substitute the channel geometric model.
36	area2TopWidth	(physicalModelInitFile)	A text file contains relational data between the drainage area and the stream cross-section top width	No, optional when available from survey. To substitute the channel geometric model.
37	area2BotWidth	(physicalModelInitFile)	A text file contains relational data between the drainage area and the stream cross-section bottom width	No, optional when available from survey. To substitute the channel geometric model.

4.1.2. *Raw Data Format*

4.1.2.1 gridFileName

The file defined by this parameter has the format shown in Table M - 10.

Table M - 10. Data format for the gridFileName file

Line#	Content	Description
1	DSAA	Magic Number to identify the file format is OWLS grid data format. It also compatible with data output from Surfer (TM) graphical software

Table M - 10. Data format for the gridFileName file (continued)

Line#	**Content**	**Description**
2	36 31	Numbers of data point in x direction (East) and y direction (North)
3	0 1050	Data range for x, in meters
4	0 900	Data range for y, in meters
5	235.18 452.6	Data range for value (elevation, or z direction), meters
6	1.70141e+038 1.70141e+038 1.70141e+038 1.70141e+038 .... 1.70141e+038 248.967 249.275 1.70141e+038 .... 1.70141e+038 1.70141e+038 1.70141e+038 1.70141e+038	Data block 1, for the first row (y = dy) of data along x direction, in this example, it has 36 data
7	1.70141e+038 1.70141e+038 1.70141e+038 1.70141e+038 .... 1.70141e+038 249.491 1.70141e+038 1.70141e+038 .... 1.70141e+038 1.70141e+038 1.70141e+038	Data block 2, has 36 data, for y = 2dy.
....:	....	Total 31 data blocks in this example

4.1.2.2 surveyElevFileName

The file defined by this parameter has the format shown in Table M - 11.

Table M - 11. Data format for surveyElevFileName file

Line#	**Content**	**Description**
1	762	Total number of land survey points
2	0 30 540 391.21	The 1st point, starting as series number 0, and the x, y, location and the elevation value. All in meters.
3	1 45.7 542 391.55	Data for 2nd survey point.
...	....	.... total 762 points in this example

4.1.2.3 triangleFileName

The file defined by this parameter has the format shown in Table M - 12.

Table M - 12. Data format for triangleFileName file

Line#	Content	Description
1	1074	Total number of triangular cells in the watershed
2	0 3 0 3 6	Point index for the 1st cell. Respectively: cell index, number of node for the cell, 1st node index, second node index and third node index. The node index is corresponding to the survey point index and is organized counter-clockwise.
3	1 3 0 7 3	Point index for the 2nd cell.
...	....	.... total 1074 cells in this example

4.1.2.4 contourFileName

The file defined by this parameter has the format shown in Table M - 13.

Table M - 13. Data format for contourFileName file

Line#	Content	Description
1	385	The 1st line value, elevation in meters
2-7	481.219330 807.898010 476.138397 807.658203 470.083588 808.373718 464.661804 808.531799 458.784790 809.106995 456.121857 811.939453	The 1st line x and y value, in meters.
8	END	The terminator for the 1st line
9	386	The 2nd line value in meters
10-12	488.248047 805.389160 483.474915 807.160583 483.474915 807.898010	The 2nd line x and y value in meters
13	END	the terminator for the line
...	...	...
...	END END	the terminator for the last line the terminator for the data file

4.1.2.5 boundaryFileName : (Same format as Table M - 13).

4.1.2.6 streamFileName: (Same format as Table M - 13).

4.1.2.7 <u>soilFileName</u>

The file defined by this parameter has the format shown in Table M - 14.

Table M - 14. Data format for soilFileName file

Line#	**Content**	**Description**
1	110 1002.452820 297.392731	The 1st soil cell ID and the x, y value of the center point
2-12	989.437500 296.000000	The x and y values for 1st cell polygon.
	980.825562 299.785461	
	980.825562 299.785461	
	1024.152466 299.675171	
	1024.152466 299.675171	
	1017.687500 297.000000	
	1011.312500 296.000000	
	1011.312500 296.000000	
	1004.500000 295.000000	
	1004.500000 295.000000	
	996.937500 295.000000	
	989.437500 296.000000	
13	END	The terminator for the 1st cell
...	END	the terminator for the last cell

4.1.2.8 <u>soilDepthFileName</u>

The file defined by this parameter has the format shown in Table M - 15.

Table M - 15. Data format for soilDepthFileName file

Line#	**Content**	**Description**
1	47	Total number of soil survey point
2	810 300 1.2 2.9	The 1st survey point, starting as the x, y values of the sample location, and the depth of the 1st and 2nd layer soil. All in meters.
3	630 270 0 3.4	Data for 2nd survey point.
...	....	.... total 47 points in this example

4.1.2.9 soilCellFileName

The file defined by this parameter has the format shown in Table M - 16.

Table M - 16. Data format for soilCellFileName file

Line#	Content	Description
1	157	Total number of soil cells
2	0 2 8	The 1st soil cell, starting as the series number, the soil cell ID number (from file defined in 4.1.2.7) and the soil type ID.
3	1 5 8	Data for 2nd soil cell.
...	....	.... total 157 soil cells in this example

4.1.2.10 arrowPointsFileName

Same format as Table M - 11 except the values of x, y, and z are for the points of the directional arrows.

4.1.2.11 arrowFacetsFileName

Same format as Table M - 12 except the indices represent the facets of the directional arrow.

4.1.2.12 testPointFileName

Same format as Table M - 11 except the values of x, y, and z are for the points of the testing model.

4.1.2.13 testIrrFacetFileName

Same format as Table M - 12 except the indices represent the facets of the testing model.

4.1.2.14 testRecFacetFileName

Same format as Table M - 12 except the indices represent the facets of the testing model.

4.1.2.15 origRainFileName

The format for the original rain data, shown for the BBWM rainfall data file, is as shown in Table M - 17. The front portion is simply a description of the data and their format. The OWLS program start to retrieve data after the head-line "DATE TIME ..." is read.

Table M - 17. Data format for origRainFileName file

```
Watershed Manipulation Project - Bear Brook Watershed in Maine.
....
DATE      In yymmdd format.
TIME      In 24 hour clock - HH:MM.
HOUR_S
HOUR_C
HOUR_E
          Hourly precipitation.  Hourly precipitation represents
          the amount of precipitation which fell in the previous
          60 minutes.  Units are inches of precipitation.  _S stands
          for Summit station, _C stands for Camp station, and _E
          stands for East Bear station.
DAY_S
DAY_C
DAY_E
          Daily precipitation.  Daily precipitation represents
          the total which fell in the previous 24 hours.
          Units are inches of precipitation.  _S stands for Summit
          station, _C stands for Camp station, and _E stands for
          East Bear station.
(  .  )   is for missing data.

DATE    TIME   HOUR_S HOUR_C HOUR_E DAY_S  DAY_C  DAY_E


871230  0:00   0      0      .      0      0      .
871230  1:00   0      0      .      .      .      .
871230  2:00   0      0      .      .      .      .
......  ....   ....   ....   ....   ....   ....   ....
890513  6:00   0.03   0      0      .      .      .
890513  7:00   0.03   0.03   0.07   .      .      .
890513  8:00   0.03   0.02   0.03   .      .      .
......  ....   ....   ....   ....   ....   ....   ....
```

4.1.2.16 origTempFileName1

Temperature Data File for temperature gauge 1. The format of the data file is that from the BBWM watershed (Table M - 18).

Table M - 18. Data format for origTempFileName1 file

```
Watershed Manipulation Project - Bear Brook Watershed in Maine (BBWM)
Soil and air temperatures (Centigrade) from the Camp station.  Data covers collection date from December 30, 1987
to December 31, 1992.


            ---------  AVERAGE  ------------   ----------  MAXIMUM  ------------   ----------  MINIMUM  ------------
                O        B        C                 O        B        C                 O        B        C
DATE   TIME HORIZ    HORIZ    HORIZ     AIR     HORIZ    HORIZ    HORIZ     AIR     HORIZ    HORIZ    HORIZ     AIR

871230   0  1.018    2.204    3.629    -18.04   1.056    2.23     3.653    -16.72   0.934    2.109    3.436    -20.51
871231   0  0.837    2.179    3.625    -18.51   0.996    2.214    3.653    -15.05   0.658    1.919    3.488    -22.76
880101   0  0.625    2.132    3.617    -9.36    0.72     2.172    3.645    -3.701   0.555    2.088    3.591    -15.68
880102   0  0.583    2.078    3.602    -1.016   0.643    2.114    3.63     5.051    0.469    1.977    3.502    -4.17
...... ... .....    .....    .....    ......   .....    .....    .....    .....    .....    .....    .....    .....
```

This file contains all temperature information from the temperature gauge including three soil horizons, and air temperature. All data are charaterizational data, or in other word, average, maximum, and minimum. As with 4.1.2.15, the OWLS program starts reading data after identifying the first string is a number.

4.1.2.17 origTempFileName2

Same as Table M - 18 except for temperature gauge 2.

4.1.2.18 origRunoffFileName

The format of runoff data, shown for from the BBWM watershed is as shown in Table M - 19.

Table M - 19. Data format for origRunoffFileName file.

```
This file contains provisional hourly discharges in cubic feet per second from
December 1, 1988 to November 30, 1992.
Data collection stations included are:
                         EAST BRANCH BEAR BROOK  USGS Station No. 01022340
                         WEST BRANCH BEAR BROOK  USGS Station No. 01022350

DATE     Data collection date (yymmdd).
TIME     Data collection time (hh:mm).
CFS_E    Hourly discharge data collection site is East Bear Brook, Station No. 01022340
FLAGE    Data flags for CFS_E.  (e) is for estimated data.
                                (.) is for missing data.
CFS_W    Hourly discharge data collection site is West Bear Brook, Station No. 01022350
FLAGW    Data flags for CFS_W.  (e) is for estimated data.
                                (.) is for missing data.


DATE    TIME CFS_E  FLAGE   CFS_W  FLAGW
--------------------------------------
881201  1:00 0.182    .     0.166    .
881201  2:00 0.182    .     0.166    .
881201  3:00 0.182    .     0.166    .
881201  4:00 0.182    .     0.166    .
881201  5:00 0.179    .     0.166    .
881201  6:00 0.166    .     0.166    .
......  .... .....    .     .....    .
```

The file contains observed data from two different stream gauges. As with 4.1.2.15, the OWLS program starts reading data after identifying the first string is a number.

4.1.2.19 rainGageFileName

The data file contains information about the rain gauges. It has the format as shown in Table M - 20.

Table M - 20. Data format for rainGageFileName file.

Line#	Content	Description
1	`nGages = 3;`	Total number of rain gauges

data block 1	`// data for gage 1`
	`name1 = "Summit Precipitation Station";`
	`dataFile1 = "rainsubh.bit";`
	`x1 = 210;`
	`y1 = 750;`
		The information about the 1st rain gauge.
		// is comment line;
		name1 is the name of the gauge
		dataFile1 is the binary file stores the processed rainfall data for the 1st gauge;
		x1, y1 is the x, y values of the 1st gauge;

Table M - 20. Data format for rainGageFileName file (continued).

Line#	Content	Description

data block 1	elevation1 = 450;
	startDate1 = 871022;
	startTime1 = 000000;
	endDate1 = 921231;
	endTime1 = 230000;
	step1 = 1;
		elevation1 is the elevation of the 1st gauge;
		startDate1, startTime1 are the date and time of the first record for gauge 1;
		endDate1, endTime1 are the date and time of the last records for gauge 1;
		step1 is the interval of the record at gauge 1.

data block 2	// data for gage 2
	name2 = "Camp Precipitation Station";
	dataFile2 = "raincamh.bit";
	x2 = 510;
	y2 = 690;
	elevation2 = 385;
	startDate2 = 871022;
	startTime2 = 000000;
	endDate2 = 921231;
	endTime2 = 230000;
	step2 = 1;
		Same as above for gauge 2.

...	// data for gage 3
	name3 = "East Bear Brook Precipitation Station";
	....
		same as above for gauge 3.
		....

4.1.2.20 cloudFileName

The format of this file is as shown in Table M - 21.

Table M - 21. Data format for cloudFileName file.

```
// Data structure for the cloud data file:
// 1. every lines with '//' heading is memo line
// 2. first digital data is the ID code of the file contain:
//    0 - no data available, follow up a constant cloud coverage;
//    1 - statistical monthly data available, follow up with
//        mm1 cloud1,..., each pair in one line
//    2 - statistical daily data available, follow up with
//        mmdd1 cloud1, ..., each pair in one line
//    3 - observed data available, follow up with
//        yymmdd1 mmddhh1 cloud1, ..., each pair in one line
0
0.8
```

For the example in Table M - 21, observed cloud coverage data are not available and a value of 0.8 cloud cover has been assigned for all days.

4.1.2.21 LAIRatioFileNamePrefix

A prefix for the filename means there are a series of files available. For example, if the prefix is "ABC", then the series of file names are: ABC0.DAT, ABC1.DAT, ABC2.DAT, etc. Total numbers of files is 1 plus the types of vegetation that the study watershed has. In the BBWM watershed, there are 12 files with 11 different types of vegetation. The one with "0" in the file name is the data for the whole watershed average. It is used in the simple model and it is very useful since many watershed including the BBWM watershed do not have LAI information for each type of vegetation. This file contains the LAI ratio data for different months of the year using the format as described and shown in Table M - 22.

Table M - 22. Data format for LAIRatioFileNamePrefix file.

```
// FILENAME: SLAI0.dat
// Seasonal Adjustable LAI Ratio for vegetation type 0 - "Average"
// ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~=~~~~=====~==
// Theory: for many forest speces, leave area index (LAI) is changing with
//         different time season of a year. An equation here is used to
//         adjust this difference:
//   LAI(t) = R(t) * LAI0;
//   LAI(t) is the LAI of the time t
//   R(t)   is the seasonal ratio presented in this file. Conifer and Hardwood
//          will expect have different R(t);
//   LAI0   is the maximum LAI of this vegetation in a year
// The 1st data is the id data (data magic number)
// id -- interger id for the file structure
//    1. when id = 0:
//       only on constant, which is not time dependent
//    2. when id = 1:
//       only monthly average data available, which arranged like this:
//         mon1 value1
//         mon2 value2
//         .... ......
//    3. when id = 2:
//       only daily data available:
//         mmdd1 value1
//         mmdd2 value2
```

Table M - 22. Data format for LAIRatioFileNamePrefix file (continued).

```
//          ..... ......
//     4. when id = 3:
//        observed hourly data available:
//          yymmdd1 hhmmss1 value1
//          yymmdd2 hhmmss2 value2
//          ....... ....... ......
// Here, data 1 means monthly data
1
1 0.01
2 0.03
3 0.1
4 0.4
5 0.7
6 0.9
7 1.0
8 1.0
9 1.0
10 0.7
11 0.1
12 0.05
```

4.1.2.22 airTempGageFileName (Table M - 23).

Table M - 23. Data format for airTempGageFileName file.

Line#	Content	Description
1	nTempGages = 2;	Total number of temperature gauges

data block 1	// data for gage 1
	name1 = "Camp Air Temperature Station";
	dataFile1 = "tempcama.dat";
	// camp station, air temperature
	x1 = 510;
	y1 = 690;
	elevation1 = 385;
	startDate1 = 871230;
	startTime1 = 000000;
	endDate1 = 921231;
	endTime1 = 230000;
	step1 = 24;
		The information about the 1st T. gauge.
		// is comment line;
		name1 is the name of the gauge
		dataFile1 is the binary file stores the processed rainfall data for the 1st gauge;
		x1, y1 is the x, y values of the 1st gauge;
		elevation1 is the elevation of the 1st gauge;
		startDate1, startTime1 are the date and time of the first record for gauge 1;
		endDate1, endTime1 are the date and time of the last records for gauge 1;
		step1 is the interval of the record at gauge 1.

data block 2	// data for gage 2
	name2 = "East Bear Brook Air Temperature Station";
	dataFile2 = "tempebba.dat";
	x2 = 630;
	....
		same as above for gauge 1.
		....

4.1.2.23 soilOTempGageFileName, soilBTempGageFileName, soilCTempGageFileName

All these files have the same structure as shown in Table M - 20.

4.1.2.24 soilTypeFileName

Different soil has different properties. The soilTypeFileName introduces a file having the hydraulic properties for different type of soils. The file has the format as shown in Table M - 24.

Table M - 24. Data format for soilTypeFileName file.

Line#	Content	Description
1	nSoil = 11;	Total number of soil types

data block 1	// soil 1, very stony
	name1 = "Berkshire_Fine_Sandy_Loam";
	ID1 = 1;
	porosity1 = 0.3; // volumetric
	nConductivities1 = 7;
	moisture11 = 1.0 // saturated
	conductivity11 = 0.0020833; // m/hr
	moisture12 = 0.9
	conductivity12 = 0.0002375; // m/hr
	moisture13 = 0.8
	conductivity13 = 0.0000371; // m/hr
	moisture14 = 0.7
	conductivity14 = 0.0000054; // m/hr
	moisture15 = 0.6
	conductivity15 = 0.0000004; // m/hr
	moisture16 = 0.5
	conductivity16 = 0.0000001; // m/hr
	moisture17 = 0.3
	conductivity17 = 0; // m/hr
	infiltration01 = 0.02; // m/hr
	infiltrationC1 = 0.002; // m/hr
		Data for the 1st soil type:
		// is comment line;
		name1 is the name of the soil;
		ID1 is the 1st type soil ID;
		porosity1 is the porosity for the 1st type soil;
		moisture11, conductivity11;
		moisture12, conductivity12;
		.........., ...;
		moisture17, conductivity17;
		are the paired relational data for the unsaturated hydraulic conductivity for the soil type 1;
		infiltration01 is the Horton maximum soil infiltration rate;
		infiltrationiC1 is the Horton minimum soil infiltration rate;

data block 2	// soil 2, very stony
	name2 = "Dixfield_Fine_Sandy_Loam";
	ID2 = 2;
	porosity2 = 0.3; // volumetric
	......
		same as above for gauge 1.
		....

4.1.2.25 vegTypeFileName

Simular to the soil types, this file stores information about different types of vegetation (Table M - 25).

Table M - 25. Data format for vegTypeFileName file.

Line#	Content	Description
1	`nVeg = 11;`	Total number of vegetation types

data block 1	`// vegetation 1`
	`name1 = "American_Beech"`
	`vegID1  =      1;`
	`coverage1 =   1;`
	`LAI1 =       10;`
	`itRate1 =  0.01;`
	`etRate1 =   0.9;`
		Data for the 1st soil type:
		// is comment line;
		name1 is the name of the soil;
		vegID1 is the 1st type soil ID;
		coverage1 is the maximum canopy coverage;
		LAI1 is the maximum leave area index in a year for the 1st type vegetation;
		itRate1 is the interception ability ratio;
		etRatio1 is the ET ability ratio;

data block 2	`// vegetation 2`
	`name2 = "Grey_Birch";`
	`vegID2  =      2;`
	`coverage2 =   1;`
	`LAI2 =       10;`
	`......`
		same as above for gauge 1.
		....

4.1.2.26 streamGageFileName

Same format as Table M - 20.

4.1.2.27 monthlyAirTempFile

This is the monthly averaged air temperature file. It contains 12 variables as shown in Table M - 26.

Table M - 26. Data format for monthlyAirTempFile file

```
// Monthly average air temperature defined by the user
// unit is in degree C
JANAvgTemp = -15.0;
FEBAvgTemp = -10.0;
MARAvgTemp = -5.0;
APRAvgTemp = 0.0;
MAYAvgTemp = 8.0;
JUNAvgTemp = 15.0;
JULAvgTemp = 18.0;
AUGAvgTemp = 17.0;
SEPAvgTemp = 10.0;
OCTAvgTemp = 5.0;
NOVAvgTemp = 3.0;
DECAvgTemp = -7.0;
```

4.1.2.28 physicalModelInitFile

This file contains all the parameters used by the OWLS hydrologic model as well as the parameters used by the simulation visualization model. All the parameters are assigned as variables. Their occurences are not ordered, which provides maximum freedom in parameter assignment. Following is the example used by the OWLS model for the BBWM watershed (Table M - 27):

Table M - 27. Data format for phydicalModelInitFile file.

```
// Parameters used for Physical Hydrological Model
//
// Unit Selections
// Unit     Length Depth  Area      Volumn       Temperature  Angle   Discharge
// -------  ------ ------ --------  -----------  -----------  -----   ---------
// INPUTS/OUTPUTS:
// Metric   meter  mm     sq.meter  cubic meter  degree C     degree  m3/s
// English  foot   inch   sq.foot   cubic foot   degree F     degree  cfs
// RUNNINGS:
//          meter  meter  sq.meter  cubic meter  degree C     radius  m3/hr
// Other unit as follows:
//          Input     Output     Model
//          ------    ------     ------
// Energy:  W/m2      W/m2       W/m2      // 1Watt/m2 = 1000 Langly/s
//          Watt      Watt       Watt      // 1Watt = 1000 Joule
//          Joule     Joule      Joule     // 1Cal = 4.186 Joule
//
```

Table M - 27. Data format for phydicalModelInitFile file (continued).

```
inputUnit  = English;           // using English unit as inputs
outputUnit = Metric;            // using Metric unit as outputs

// === Output control ===
startDate  = 890501;            // start date of the output
startTime  = 000000;            // start time of the output
endDate    = 900501;            // ending date of the output
endTime    = 000000;            // ending time of the output
step         = 1;               // step of time in running
saveInterval = 1;               // save results every # steps
useDepthOutput = FALSE;         // using depth value for outputs instead of volumes

// === interception capacity ===
LAItoInterceptFileName = "lai2intc.dat";   // file name for LAI to Interception
relations
canopyMinETRate = 0.01;         // in inch/hr

// === Horton infiltration equation parameters ===
// f = fc + (f0 - fc) * exp(-k * t)
useHorton = TRUE;               // option to use model instead
m2fFilePrefix = "sm2f";         // prefix of moist to infiltration relation datafile
infiltration_k  = 0.8;          // empirical constant in 1/hr
infiltration_a  = 0.5;          // empirical index for soil-moisture and Horton
                                // time function
conductivityAdjust = 500;       // adjust factor for soil conductivity 1.0 means no
                                // change, 1.1 mean increase 10%
infiltration0Adjust = 0.10;     // adjust factor for Horton Maximun infiltration
                                // rate, =1 means no change
infiltrationCAdjust = 0.10;     // adjust factor for Horton Minimun infiltration
                                // rate, =1 means no change

// === Snow melting Degree-Time Melt Equation ===
// modified from Degree-Day Melt Equation:
// M = Df * (Ta - Tb)
// M - daily melt => timely melt (L/T)
// Ta - average daily air temperature => average time temp (Te)
// Tb - base melt temperature (Te)
// Df - degree-day factor (L/T/Te) = 0.05 - 0.15
//      => timely factor (L/T/Te) = (0.05 - 0.15) / timeDay
snowMelt_Df = 0.004;      // in in/hr/dF of water equavalance
snowMelt_Tb = 33;         // when T > 0.5 degreeC, snow melts

// === ET model ===
// A physical ET function modified from Jensen&Haise (1963):
// PET = (a * T + b) * Rs * dt / 24.0
// where:
// PET -- potential evaportransipiration, (cm/day);
// T   -- mean air temperature, (dC);
// Rs  -- solar radiation, (cm/day equavalent water)
// a   -- constant, = 0.025 for daily estimation;
```

Table M - 27. Data format for phydicalModelInitFile file (continued).

```
// b   -- constant, = 0.078 for daily estimation;
// dt  -- time interval in hour(s)
ET_a = 0.025;
ET_b = 0.078;
underCanopyETConstant = 0.04;
soilETConstant = 0.01;
soilWaterSupplyC1 = 0.1;        // means maximum 0.1"/hr
soilWaterSupplyC2 = 0.01;       // means offset is 0.01"/hr
LAItoETFileName = "lai2et.dat"; // ET ratio from canopy is seasonal varied

// === surface roughness ===
roughness = 0.8;

// === soil ===
// into distributed parameters
layerWeight1 = 0.0;             // the weight that defines the soil depth
layerWeight2 = 1.0;             // by the 1st or 2nd layers

// === macropore ===
// options provided by the program:
// option 1: user provide relational data in the following files (recommended)
// option 2: if no file is found, use model build-in conceptual function
// option 1:
drainArea2PipeRadius = "s2piper.dat";
                          // relation between soil depth and macropore pipe radius
area2PipeCount       = "a2pipec.dat" ;
                          // relation between upper drainage area and macropore
                          // pipe count
volume2PipeCount     = "v2pipec.dat";
                          // relation between soil volume and the pipe counts
// option 2:
minDiameter = 0.0003937;  // in 0.03937 inch, or 1 mm
pipeRatio   = 2.5;        // power to the pipe water saturation to adjust
                          // the diameter
radiusA = 0.02;           //0.002, = 0.02"=0.5mm
                          // pipeRadius = radiusA + radiusB * drainArea^radiusC
radiusB = 0.005;          // = 0.0005", 0.0015
radiusC = 0.5;            // 0.5 this parameter should be consider in metric area
countA = 1;               // = 5, pipeCount = countA + countB * area^countC
                          //                  + countD * volume^countE
countB = 0.5;
countC = 0.5;
countD = 0;
countE = 0;
surfaceMacroporeConst = 1.6;    //  in decimal toMacroporeFlow =
                                //  C * infiltrationDepth * (1 - satuation)
```

Table M - 27. Data format for phydicalModelInitFile file (continued).

```
soilMacroporeConst    = 0.000001;    // = 0.005 in 1/hr toMacroporeFlow = C *
                                     //   S_waterDepth * dt * (1 - satuation)
frictionCoeff = 20;                  // 0.04-0.06 for Re > 4000, turbulant flow

// === flow Routing model ===
iterativeErr = 0.01;            // error criterion for Newton's iterations
maxIterations = 100;            // ...
useDirectInputs = FALSE;

// === channel geometry ===
// NOTE: using user provided data relation file is first priority
//       parameters will be used only this file is not available
// like macropore, has two obtions
// option 1: relation data files:
area2Depth  = "a2rdepth.dat";// relation between upper drainage area and river depth
area2TopWidth  = "a2rtopw.dat"; // relation between upper drainage area and river
                                // top depth
area2BotWidth  = "a2rbotw.dat"; // relation between upper drainage area and river
                                // bottom depth
// option 2: empirical constant for channel cross-section equation
// REMEMBER: drainage area will converted to sq.km
widthTopConstant = 5.0;    // 3.0 topWidth = widthTopConstant * catchArea^widthTopPow
widthBotConstant = 2.6;    // 1.6
depthConstant    = 4.0;    // 0.3
widthTopPow      = 0.3;    // 0.3
widthBotPow      = 0.3;    // 0.3
depthPow         = 0.3;    // 0.2

// === initial condition ===
soilMoisture            = 0.52; // initial soil moisture for all cells
initialStreamDepthRatio = 0;    // constant used to set initial stream water
condition
                                // in related to the maxDepth
// === Filenames for outputs ===
// Remember to change the filename for different watershed
paramFileName         = "ebbpar.dar";    // for interruption
fluxFileName          = "ebbflux.dar";   // for interruption
basinInfoFileName     = "ebbinfo.dar";   // for graphics
segmentFileName       = "ebbseg.dar";    // for graphics
channelFileName       = "ebbchan.dar";   // for graphics
cellCanopyFileName    = "ebbcan.dar";    // for graphics
cellSurfaceFileName   = "ebbsur.dar";    // for graphics
cellSoilFileName      = "ebbsub.dar";    // for graphics
cellMacroporeFileName = "ebbmac.dar";    // for graphics
cellAllFileName       = "ebball.dar";    // for graphics
sumFluxFileName       = "ebbfroot.dar";  // for graphics
counterFileName       = "ebbcount.dar";  // for graphics, size of records
textOutputFileName    = "ebbok.txt";     // text output file
riverOutputFileName   = "ebbriver.txt";  // text output file for stream flow
```

Table M - 27. Data format for phydicalModelInitFile file (continued).

```
// ==== Graphic Visualization Options ====
// User defines time range
isUserTimeRange    = TRUE;   // Automatic when FALSE
displayStartDate   = 890510; // date of start   901108, 890510
displayEndDate     = 890517; // date of end     901120, 890517
displayInterval    = 2;      // every interval
// User defines data range
isRelationalGrade  = TRUE;   // FALSE use following data
minCellV           = 100;    // for cell water
maxCellV           = 900;    // moist(100);
                             // depths: 100mm - 900mm (all)
                             // flow:   15 - 30 (0.001m3/s)
minSegV            = 0;      // for stream water
maxSegV            = 0.20;    // V(1.0);Q(0.05-0.2);W(2);H(0.5)
widthConst         = 10.0;   // Amplification ratio for width, 10
heightConst        = 1000.0; // Amplification ratio for height, 1000

// Option of Components to be colored
waterComponentID  = 51;
     //= 0 None
     //= Canopy Information 1x                 Unit in File     Unit for Output
          // = 11 Intercepted Water Depth                (m)               (mm)
          // = 12 Intercepted Snow Depth                 (m)               (mm)
          // = 13 Rainfall to the Canopy                 (m)               (mm)
          // = 14 Canopy ET                              (m)               (mm)
          // = 15 Canopy Net Precipitation               (m)               (mm)
          // = 16 Intercepted Water&Snow Depth =11+12    (m)               (mm)
     //= Surface Information 2x
          // = 21 Surface Water Depth                    (m)               (mm)
          // = 22 Surface Snow Depth                     (m)               (mm)
          // = 23 Surface ET                             (m)               (mm)
          // = 24 Surface Infiltraion                    (m)               (mm)
          // = 25 Surface Flow                          (m3)         (0.001m3/s)
          // = 26 Surface Water&Snow Depth  = 21+22      (m)               (mm)
     //= Subsurface Information 3x
          // = 31 Soil Moisture                          (m)               (mm)
          // = 32 Soil Water Depth                       (m)               (mm)
          // = 33 Soil ET                                (m)               (mm)
          // = 34 Soil Flow                             (m3)         (0.001m3/s)
      //= Macropore Information 4x
           // = 41 Macropore Water Depth                 (m)               (mm)
           // = 42 Water from Suface                     (m)               (mm)
           // = 43 Water from Soil                       (m)               (mm)
           // = 44 Flow                                 (m3)         (0.001m3/s)
           // = 42 Water from Suface&Soil  = 42+43       (m)               (mm)
      //= Summation 5
           // = 51  Total Cell Water Depth               (m)               (mm)
           // = 52  Total Cell Flow (=30)               (m3)         (0.001m3/s)
riverComponentID  = 2;  // = 0 None
```

Table M - 27. Data format for phydicalModelInitFile file (continued).

```
                        // = 1 Velocity              (m/s)
                        // = 2 Discharge             (m3/s)
                        // = 3 Channel Width         (m)
                        // = 4 Stream Water Depth    (m)
// Coloring
basinColor.R       = 255;    // Basin outside area color
basinColor.G       = 200;
basinColor.B       = 200;
// Legend Color
nGrades            = 20;     // number of grades
baseMinCellColor.R = 255;    // the minimum color for the cell
baseMinCellColor.G = 50;     // representing maximum values (darker)
baseMinCellColor.B = 50;
baseMaxCellColor.R = 255;    // the maximum color for the cell
baseMaxCellColor.G = 255;    // representing minimum values (lighter)
baseMaxCellColor.B = 50;
baseMinSegColor.R  = 50;     // the darker color for the stream
baseMinSegColor.G  = 50;     // representing higher values
baseMinSegColor.B  = 100;
baseMaxSegColor.R  = 200;    // the ligher color for the stream
baseMaxSegColor.G  = 200;    // representing lower values
baseMaxSegColor.B  = 255;
timeMajorInt       = 24;     // major tig interval, in hours
timeMinorInt       = 1;      // minor tig interval, in hours
// 2-D
twoDBackground.R   = 255;
twoDBackground.G   = 255;
twoDBackground.B   = 255;
isZeroOrigin       = FALSE;   // y start from 0 except air temp.
AirTempColor.R     = 255;
AirTempColor.G     = 0;
AirTempColor.B     = 0;
PrecipColor.R      = 0;
PrecipColor.G      = 0;
PrecipColor.B      = 0;
ETColor.R          = 255;
ETColor.G          = 0;
ETColor.B          = 0;
ObservedColor.R    = 0;
ObservedColor.G    = 0;
ObservedColor.B    = 0;
SimulatedColor.R   = 255;
SimulatedColor.G   = 0;
SimulatedColor.B   = 0;
SurfaceFColor.R    = 100;
SurfaceFColor.G    = 255;
SurfaceFColor.B    = 100;
MPipeFColor.R      = 50;
MPipeFColor.G      = 50;
MPipeFColor.B      = 255;
```

Table M - 27. Data format for phydicalModelInitFile file (continued).

```
SoilFColor.R      = 0;
SoilFColor.G      = 0;
SoilFColor.B      = 100;
CSnowDColor.R     = 255;
CSnowDColor.G     = 255;
CSnowDColor.B     = 255;
CWaterDColor.R    = 250;
CWaterDColor.G    = 250;
CWaterDColor.B    = 250;
SSnowDColor.R     = 200;
SSnowDColor.G     = 200;
SSnowDColor.B     = 200;
SWaterDColor.R    = 255;
SWaterDColor.G    = 255;
SWaterDColor.B    = 200;
MPipeDColor.R     = 255;
MPipeDColor.G     = 200;
MPipeDColor.B     = 255;
SoilDColor.R      = 50;
SoilDColor.G      = 100;
SoilDColor.B      = 100;
```

4.1.2.29 LAItoInterceptFileName

This is a relational data file with the format as shown in Table M - 28:

Table M - 28. Data format for LAItoInterceptFileName file.

```
// FILENAME: LAI2INTC.DAT
// Data file for relations between
// available LAI (LAI * coverage) and
// interception capacity ratio (R)
// INTC = INTC0 * R(LAI(t))
// LAI -- INTC
0  0
1  0.1
2  0.2
3  0.3
4  0.4
5  0.5
6  0.6
7  0.7
8  0.8
9  0.9
10 1.0
```

4.1.2.30 m2fFilePrefix

This is prefix for a series of text files which contain relational data between relative soil moisture content and infiltration ability for all types of soil in the watershed. The files are in the format of relational data structure. For example, when m2fFilePrefix = "infil". The file for soil type 1 in the BBWM is infl1.dat with the format in Table M - 29.

Table M - 29. Data format for m2fFilePrefix file.

```
// FILENAME: infil1.dat
// infiltration relation for soil type 1 - "Berkshire_Fine_Sandy_Loam"
// infiltrationRate = f(soilMoisture)
// soilMoisture = volumetric relative soil moisture contain range : 0 - 1.0
// infiltration = rate of water infiltrate to the soil under that soil moisture
//               condition, in m/hr
0   0.10
0.1 0.09
0.2 0.08
0.3 0.07
0.4 0.06
0.5 0.05
0.6 0.04
0.7 0.03
0.8 0.02
0.9 0.01
1   0.005
```

4.1.2.31 LAItoETFileName

Same format as Table M - 28. The 1st column represents the leaf-area-index (LAI) and the 2nd column the ET factor.

4.1.2.32 drainArea2PipeRadius

When available, this file is also organized as a relational data format shown in Table M - 28. The 1st column represents the drainage area and the 2nd column the averaged macropore pipe radius.

4.1.2.33 area2PipeCount

Same format as Table M - 28. The 1st column represents the area of the cell and the 2nd column the averaged macropore pipe counts.

4.1.2.34 volume2PipeCount

Same format as Table M - 28. The 1st column represents the volume of soil in a cell and the 2nd column the averaged macropore pipe counts.

4.1.2.35 area2Depth

Same format as Table M - 28. The 1st column represents the drainage area upstream and the 2nd column the averaged depth of the channel segment.

4.1.2.36 area2TopWidth

Same format as Table M - 28. The 1st column represents the drainage area upstream and the 2nd column the averaged top width of the channel segment

4.1.2.37 area2BotWidth

Same format as Table M - 28. The 1st column represents the drainage area upstream and the 2nd column the averaged bottom width of the channel segment

4.1.3. *Parameter Assignment*

Besides the file names and their corresponding data, there are a number of parameters required for the data processing model (Table M - 30).

Table M - 30. Parameters for the data processing model.

No	Parameter	In File	Description	Required?
1	searchRange	owls.ini	Range of data searching for watershed automatic mesh creation. in meters	Yes
2	contourPointFileName contourLineFileName	owls.ini	Processed contour file in binary format	No, used only in graphical visualization model
3	boundaryPointFileName boundaryLineFileName	owls.ini	Processed study boundary file in binary format	No, used only in graphical visualization model
4	streamPointFileName streamLineFileName	owls.ini	Processed stream file in binary format	No, used only in graphical visualization model
5	soilPointFileName soilLineFileName	owls.ini	Processed soil file in binary format	No, used only in graphical visualization model
6	griddingMethod	owls.ini	Options for method of creating triangular mesh from a cluster of 4 points.	=0, the OWLS model will use hypotenuse center close to reference point; =1, the OWLS model will use the hypotenuse determined by difference of normal vector; =2, the OWLS model will use the hypotenus direction from SW to NE; =3, the OWLS model will use the hypotenus direction from NW to SE;
7	referCenterFileName	owls.ini	Calculated reference point data in binary format	Yes when griddingMEthod = 0.
8	basinPointFileName	owls.ini	The processed point file in binary format	Yes, used for output and by the hydrologic model
9	basinCellNodeFileName	owls.ini	The processed cell node file in binary format	Yes, used for output and by the hydrologic model
10	basinCellEdgeFileName	owls.ini	The processed cell edge file in binary format	Yes, used for output and by the hydrologic model
11	basinEdgeFileName	owls.ini	The processed edge file in binary format	Yes, used for output and by the hydrologic model
12	basinBoundaryTreeFileName	owls.ini	The processed binary file for basin boundary tree.	Yes, used for output and by the hydrologic model
13	basinBoundFileName	owls.ini	The processed binary file for basin boundary.	Yes, used for output and by the hydrologic model
14	basinFlowpathsFileName	owls.ini	The processed binary file for all the flowpaths from each watershed cell	Yes, used for output and by the delineation model
15	basinFlowpathTreeFileName	owls.ini	The processed binary file for basin flowpath tree.	Yes, used for output and by the hydrologic model
16	basinStreamTreeFileName	owls.ini	The processed binary file for basin stream tree.	Yes, used for output and by the hydrologic model
17	newBasinFileName	owls.ini	The processed binary file for processed basin.	Yes, used for output and by the hydrologic model

Table M - 30. Parameters for the data processing model (continued).

No	Parameter	In File	Description	Required?
18	basinOutletNodeIdx	owls.ini	The node index for the basin outlet.	Yes, it should be determined before run the data processing model and adjusted to fit the stream gage station by examing the graphical output.
19	basinFileFormat	owls.ini	The option for output file: either BINARY or ASCII (Text). Recommend binary for faster access and smaller files.	Yes.
20	boundaryDivideCell	owls.ini	Allow the new generated watershed boundary to subdivide the cells that it pass through.	Yes. Recommended to be TRUE.
21	markCellOnPath	owls.ini	A switch for marking the flowpath. When TRUE, a cell will be marked when a flowpath go through it. When FALSE, a cell will not be marked unless the flowpath from this cell go through the stream outlet node.	Yes. No much different to the results of the watershed boundary between choosing TRUE and FALSE. Pick TRUE as perfered.
22	isPartialPaths	owls.ini	An switch for running time saving purpose. It takes a while to create the watershed flowpaths from top to down. When isPartialPaths is TRUE, all flowpath creation will be stopped whenever the path node has lower elevation than the stream outlet. It saves time but the path will be partial.	Yes. Pick TRUE for faster completion. FALSE for the whole look.
23	messageLevel	owls.ini	For program debugging purpose. =0, no debug. =1, debug water balance only, err.log file will be created for output; =2, debug stream segment only, files err.log and segment.log will be created. =3, debug visualiztion only, reserved; =4, debug both 1 and 2; =5, debug all.	Yes.
24	basinTreeLogFileName	owls.ini	A text log file have the stream tree data printout.	Yes. It will be updated when messageLevel = 5.
25	pathTreeLogFileName	owls.ini	A text log file for basin flowpath tree.	Yes. It will be updated when messageLevel = 5.
26-1	outRainFileName1	bbwmdata.ini	A binary file for storing processed rainfall data from origRainFileName data row 1	Yes, it is an output file
26-...	....	....	....	....
26-#	outRainFileName#	bbwmdata.ini	A binary file for storing processed rainfall data from origRainFileName data row #	Yes, it is an output file
27	ourDaulyFileName	bbwmdata.ini	A text file stores the daily precipitation output.	Yes, it is an output file

Table M - 30. Parameters for the data processing model (continued).

No	Parameter	In File	Description	Required?
28-1	outTempFileName11	bbwmdata.ini	A binary file stores the processed data for the 1st row of records from the 1st temperature gauge file defined by (<u>origTempFileName1</u>)	Yes, it is an output file
28-...	....	....	....	....
29-X	outTempFileNameX	bbwmdata.ini	A binary file stores the processed data for the Y's row of records from the 1st temperature gauge file defined by (<u>origTempFileNameX</u>)	Yes, it is an output file
30-1	outRunoffFileName1	bbwmdata.ini	A binary file stores the processed observed data from hydrologic gauge 1	Yes, it is an output file
30-...	....	....	....	Yes, it is an output file
30-#	outRunoffFileName#	bbwmdata.ini	A binary file stores the processed observed data from hydrologic gauge #	Yes, it is an output file
31	basinIndexFileName	owlshydr.ini	A text file stores the cell index table for the cells within the simulation watershed	

4.1.4. *Run Data Processing Program*

After all the above data files have been prepared and all the parameters have been assigned. The Data Processing Program can be test run by following the steps below:

Step 1: Run the Gauge Data Conversion Program by calling the windows as shown in Figure M - 1 under the OWLS Windows Program. There are three portions:

(1). Rainfall Data Processing: to process the precipitation data;

(2). Stream Flow Data Processing: to process the stream flow data;

(3). Temperature Data Processing: to process the air temperature data.

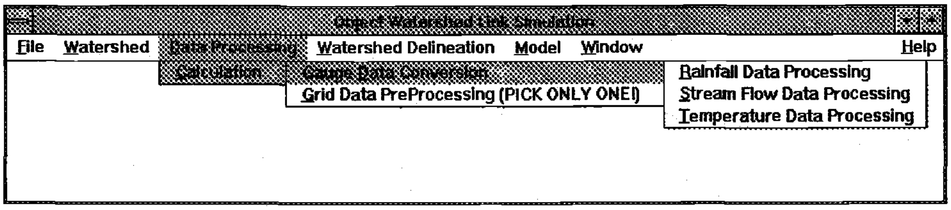


Figure M - 1. Windows to run data Gauge Data Conversion Programs.

Step 2: Run one of the Grid Data Preprocessing Program by calling the windows as shown in Figure M - 2 under the OWLS Windows Program. There are two portions:

(1). Automatic Gridding: to convert the digital elevation data into triangular mesh;

(2). Irregular Triangle Transforming: to transform the files defined by parameters surveyElevFileName and <u>triangleFileName</u> into <u>OWLS binary</u> data format for irregular triangular mesh.

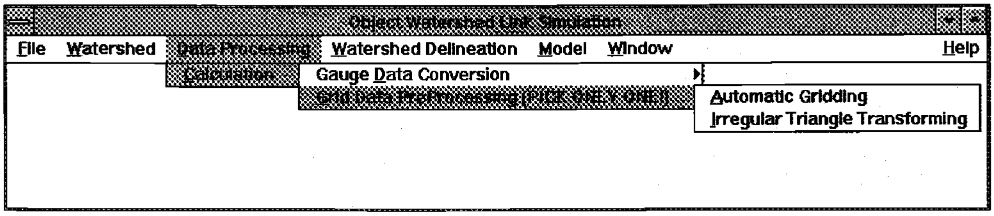


Figure M - 2. Windows to run Grid Data Processing Programs

Step 3: Run the Build Basin Program to perform the automatic delineation of watershed flowpaths, boundary and stream by calling the windows as shown in Figure M - 3 under the OWLS Windows Program. There are two portions:

(1). One step run: which will run build flow paths, then build flow path tree, then build stream, and at last divide boundary cells;

(2). Step-by-step run: to run individual portions to avoid redundent works like build flow paths for just attempting to find a watershed.

This program may run 20 or more minutes depending upon the size of the watershed and

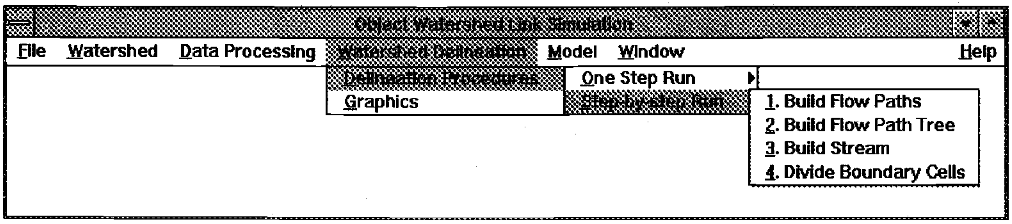


Figure M - 3. Windows to run Delineation Procedures.

the speed of the computer.

Step 4: After the aboved steps have been completed, run the OWLS visualization program to see if the processed watershed mesh is acceptable by calling the windows as shown in Figure M - 4 under the OWLS Windows Program.

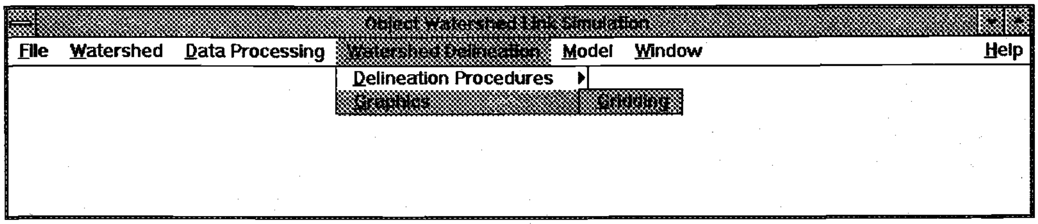


Figure M - 4. Window to run the visualization program of a delineated watershed

4.2. *Run the Visualization Model*

4.2.1. *Check List*

Table M - 31 lists the files that may be used by the visualization model.

Table M - 31. List of files for the visualization model.

No.	Parameter(s)	In File	Description	Required?
1 2	contourPointFileName contourLineFileName	owls.ini	Visulation of the watershed contour lines. These files are generated by the data processing model from contourFileName (4.1.2.4).	Yes if the watershed contour need to be displayed.
3 4	boundaryPointFileName boundaryLineFileName	owls.ini	Visulation of the watershed study boundary. These files are generated from the boundaryFileName (4.1.2.5) from the data processing model.	Yes if the study area need to be displayed.
5 6	streamPointFileName streamLineFileName	owls.ini	Visulation of the watershed digitalized stream channel(s). These files are generated from the streamFileName (4.1.2.6) from the data processing model.	Yes if the comparison between sminulated stream network and the digitalized stream network is required.
7 8	soilPointFileName soilLineFileName	owls.ini	Visualization of the watershed soil distribution. These files are generated from the soilFileName (4.1.2.7) and soilCellFileName (4.1.2.9) by the data processing model.	Yes if the watershed soil map need to be display.
9 10	arrowPointFileName arrowFacetFileName	owls.ini	Visualization of the Directions. These files are in ASCII format and preset. They display the direction of Up, North and East.	Yes.

Table M - 31. List of files for the visualization model (continued).

No.	Parameter(s)	In File	Description	Required?
11	testPointFileName	owls.ini	A simple 3-D model to test the visualization model. These file are in text format and preseted.	No. But it help to understand the visualization model by changing the data in these files.
12	testIrrFacetFileName			
13	testRecFacetFileName			
14	basinPointFileName	owls.ini	A set of files for the watershed, all in binary format, generated from the data processing model by using data provided by gridFileName (4.1.1.1), surveyElevFileName (4.1.1.2) and triangleFileName (4.1.1.3) plus the parameters as described in aboved section.	Yes.
15	basinCellNodeFileName			
16	basinCellEdgeFileName			
17	basinEdgeFileName			
18	newBasinFileName			
19	basinBoundaryTreeFileName			
20	basinBoundaryFileName			
21	basinFlowpathsFileName			
22	basinFlowpathTreeFileName			
23	basinStreamTreeFileName			
24	segmentFileName	(physicalModelInitFile)	A binary file stores the stream segment coordinate values and indices	Yes, used for simulation results visualization
25	channelFileName	(physicalModelInitFile)	A binary file stores the channel flow data from simulation	Yes, used for simulation results visualization
26	basinInfoFileName	(physicalModelInitFile)	A binary file stores the statistical information and values of parameters	Yes, used for the 2-D visualization
27	cellCanopyFileName	(physicalModelInitFile)	The binary files store the water information of the basin cells. Automatic generated from the hydrologic model.	Yes, used for simulation results visualization
28	cellSurfaceFileName			
29	cellSoilFileName			
30	cellMacroporeFileName			
31	cellAllFileName			
32	sumFluxFileName	(physicalModelInitFile)	A binary file stores the watershed average and outlet flow data	Yes, used for simulation results visualization
33	counterFileName	(physicalModelInitFile)	A text file output from hydrologic model contains the data about the simulation steps, times.	Yes, used for simulation results visualization
34	basinIndexFileName	owlshydr.ini	A text file output from data processing model for the indices of cell within the simulation watershed.	Yes, used for simulation results visualization
35	basinTreeLogFileName	owls.ini	Log file created from the program debugging. Storing the output for the basin stream tree.	Yes.
36	pathTreeLogFileName	owls.ini	Log file created from the program debugging. Storing the output of the basin flowpath tree.	Yes.

4.2.2. *Parameter Assignment*

There are many parameters used by the visualization model (Table M - 32):

Table M - 32. Parameters used by the visualization model.

No.	Parameter(s)	In File	Description	Required?
1	drawCoordinate	owls.ini	TRUE or FALSE, a switch to turn on/off the drawing of coordinate for the watershed.	Yes
2	drawPaintedInnerBox	owls.ini	TRUE or FALSE, a switch to turn on/off the drawing of coordinate wall for the watershed to enhance the 3-dimension effect.	Yes
3	drawBoundary	owls.ini	TRUE or FALSE, a switch to turn on/off the drawing of study boundary of the watershed.	Yes

Table M - 32. Parameters used by the visualization model (continued).

No.	Parameter(s)	In File	Description	Required?
4	drawBoundaryLines	owls.ini	TRUE or FALSE, a switch to turn on/off the drawing of simulated boundary line for the catchment associated with the user-defined outlet node.	Yes
5	drawBoundaryTree	owls.ini	TRUE or FALSE, a switch to turn on/off the drawing of the simulated boundary tree for the watershed. Note: it is not a complet boundary.	Yes
6	drawFlowpathTree	owls.ini	TRUE or FALSE, a switch to turn on/off the drawing of simulated flowpath tree for the watershed.	Yes
7	drawStreamTree	owls.ini	TRUE or FALSE, a switch to turn on/off the drawing of simulated stream network tree for the watershed.	Yes
8	drawFlowpaths	owls.ini	TRUE or FALSE, a switch to turn on/off the drawing of all the flowpaths (complete/partial) for the watershed.	Yes
9	drawOnlyMarked	owls.ini	A switch parameter for choosing if drawing only the flowpath that marked following some criteria. =0, draw all flowpath using color seperation, criteria by upper drainage area of the flowpath segment; =1, draw all flowpath using color seperation, criteria by the length from the flowpath segment to the catchment outlet.	Yes
10	drawShadeId	owls.ini	Parameter defines the coloring method of the catchment: =-1, no background color; =0x, even background coloring; =1x, solar shading background; x=0, no additional shade inside the catchment; x=1, even additional shade inside catchment; x=2, additional shade according to value defined by <u>drawCellValue</u> x=3, white shade inside the catchment.	Yes
11	drawCellValue	owls.ini	A switch to turn on the cell values drawing/writing at the position of the cell (writing is control by <u>drawCellText</u>): =0, do not draw it =1, shade/write out the cell index; =2, shade/draw/write out the cell elevation; =3, shade/write out the cell value (upper drainage area); =4, shade/write out the cell 1st layer soil depth; =5, shade/write out the cell 2nd layer soil depth;	Yes
12	shadeMethod	owls.ini	Shading method option: =0, linear; =1, log	Yes
13	drawCellText	owls.ini	TRUE/FALSE, a switch to turn on/off writing values onto the cell	Yes
14	thresholdLength	owls.ini	A value in meters to cut out the portion of flowpath having flow-length to the catchment outlet larger than this number (upper-stream portion).	Yes

Table M - 32. Parameters used by the visualization model (continued).

No.	Parameter(s)	In File	Description	Required?
15	thresholdArea	owls.ini	A value in sq. meters to cut out the portion of flowpath having upper drainage area smaller than this number (upper-stream portion).	Yes
16	fillSurface	owls.ini	TRUE/FALSE, an option to turn on/off the surface painting (filling) of the watershed.	Yes
17	gridDrawFill	owls.ini	TRUE/FALSE, an option to turn on/off the mesh-line drawing while painting painting (filling) the surface of the watershed.	Yes
18	hiddenSurface	owls.ini	TRUE/FALSE, an option to turn on/off the back surface removal.	Yes
19	backgroundDisplay	owls.ini	TRUE/FALSE, an option to turn on/off the background painting.	Yes
20	viewSpecified	owls.ini	TRUE/FALSE, an option to turn on/off the viewing coordinates specification.	Yes
21	viewTrueScale	owls.ini	TRUE/FALSE, an option to turn on/off the coordinate scale (exact/relative).	Yes
22	delta	owls.ini	The mesh size interval that used to draw the watershed. When pick 1, all mesh of the watershed will be drawen. When pick 2, every other mesh line will be drawn and so forth.	Yes
23	viewAngleStep	owls.ini	The turn-around step angle in degree used for watershed bird-viewing.	Yes
24	viewHorizonAngle	owls.ini	The angle (degree) to the North from the viewing-line that the viewer looks at a watershed in the sky.	Yes
25	viewZenithAngle	owls.ini	The angle (degree) to the zenith from the viewing-line that the viewer looks at a watershed in the sky.	Yes
26	viewReference	owls.ini	The location that the viewer focuses on. Values include: CENTER, NORTH, EAST, SOUTH, WEST, NE, SE, SW, NW.	Yes
27 28 29	viewUp.x viewUp.y viewUp.z	owls.ini	The direction vector of the wiewing window's up-direction. Expressed as values in x (East), y (Up) and z (North).	Yes
30 31	marginX marginY	owls.ini	The screen display margin in x and y direction. Values are the ratiio to the screen size.	Yes
32	normalCoord	owls.ini	TRUE/FALSE, a switch value to turn on/off relative coordinates.	Yes
33	normals	owls.ini	The maximum value of the relative coordinates in both direction. Note: it is a relative coordinate maximum value used for display. It would not affect the display results.	Yes, but no change necessary.
34	verticalScale	owls.ini	A scale ratio to increase/decrease the unit on the verticle coordinate. Since the range in vertical topography is much less than horizontal. To clearly see the topographical changes, scaling the vertical unit is necessary. The smaller scale value will enlarge the vertical difference.	Yes

Table M - 32. Parameters used by the visualization model (continued).

No.	Parameter(s)	In File	Description	Required?
35	pointLightOn	owls.ini	TRUE/FALSE, a switch parameter to allow the point-light (sun light) taking into effect on the 3-D view. It will dramatically enhance the 3-D effect.	Yes
36 37 38	ambientLight.R ambientLight.G ambientLight.B	owls.ini	A set of parameters represents the Red, Green and Blue spectrum for the ambient light. Range of the value is 0 to 255. The larger the value, the lighter (more intensive) of the light. For black, {R,G,B} will be assigned {0, 0, 0}. For Bright Red {255, 0, 0}.	Yes
39 40 41	farpointLight.R farpointLight.G farpointLight.B	owls.ini	Similar as above. Its the light intensity for the far-point light source (sun). Strong White Light will be {255, 255, 255}.	Yes
42 43 44	lightVector.x lightVector.y lightVector.z	owls.ini	The direction vector of the light, ponting from the watershed to the light source and expressed as values in x (East), y (Up), and z (North).	Yes
45 46 47	reflectRatio.R reflectRatio.G reflectRatio.B	owls.ini	The raflection ratio of the watershed surface. Expressed in the ratio for Red, Green, and Blue spectrum. For example, if the watershed is pure green, the {R,G,B} ratio may be {0, 1.0, 0}.	Yes
48	diffuseCoeff	owls.ini	The diffusion coefficient of the watershed surface.	Yes
49	diffuseOrder	owls.ini	The diffusion parameter describing the orders of diffusion.	Yes
50	startLocalTime	owls.ini	The parameter is used for day time animation of the watershed. The value is the hour of a day when we start to see the watershed.	Yes
51	timeStep	owls.ini	Same use as above. Its value is the step increment that we see the watershed as time passing.	Yes
52	solarAngle	owls.ini	The solar angle for the time animation.	Yes
53	triangleGridView	owls.ini	TRUE/FALSE, a switch value to switch between the triangular and rectangular mesh.	Yes
54	deltaContour	owls.ini	The contour interval that you want to see from the watershed 3-D contour window.	Yes
55 56 57	backgroundColor.R backgroundColor.G backgroundColor.B	owls.ini	A set of paramters for the background of the screen, in R,G,B. Default is {255,255,255}.	Yes
58 59 60	foregroundColor.R foregroundColor.G foregroundColor.B	owls.ini	A set of parameter for the screen foreground color in R, G, B. Default is {0,0,0}.	Yes
61	perspective	owls.ini	TRUE/FALSE, a parameter to switch between perspective and orthographic projection for the 3-D objects and the watershed. Perspective projection offers a length changes between near and far and giving a better 3-D effect.	Yes
62	viewDistance	owls.ini	The distance from the viewer to the watershed. It only takes into effect for perspective projection.	Yes
63 64	DocumentWidth DocunebtHeight	owls.ini	The size of graphical file to save the screen image.	Yes

Table M - 32. Parameters used by the visualization model (continued).

No.	Parameter(s)	In File	Description	Required?
65	isUserTimeRange	(physicalModelInitFile)	TRUE/FALSE, Simulation Visualization Time Range, if TRUE, starts from displayStartDate and ends after displayEndDate. Otherwise goes from the begining of the simulation to the end.	Yes
66 67 68	displayStartDate displayEndDate displayInterval	(physicalModelInitFile)	Simulation Visualization Time range and interval of steps. Respectively in format of yymmdd, yymmdd and interger (1, 2, ...)	Yes
69	waterComponentID	(physicalModelInitFile)	Simulation Visualization. The option for water components to be colorred and displayed: Canopy Information 1# = 11 Intercepted Water Depth (mm) = 12 Intercepted Snow Depth (mm) = 13 Rainfall to the Canopy (mm) = 14 Canopy ET (mm) = 15 Canopy Net Precipitation (mm) = 16 Intercepted Water&Snow Depth =11+12 (mm) Surface Information 2# = 21 Surface Water Depth (mm) = 22 Surface Snow Depth (mm) = 23 Surface ET (mm) = 24 Surface Infiltraion (mm) = 25 Surface Flow (0.001m3/s) = 26 Surface Water&Snow Depth = 21+22 (mm) Subsurface Information 3x = 31 Soil Moisture (mm) = 32 Soil Water Depth (mm) = 33 Soil ET (mm) = 34 Soil Flow (0.001m3/s) Macropore Information 4x = 41 Macropore Water Depth (mm) = 42 Water from Suface (mm) = 43 Water from Soil (mm) = 44 Flow (0.001m3/s) = 42 Water from Suface&Soil = 42+43 (mm) = Summation 5 = 51 Total Cell Water Depth (mm) = 52 Total Cell Flow (=30) (0.001m3/s)	Yes
70	riverComponentID	(physicalModelInitFile)	Simulation Visualization. The option for flow components to be colorred and displayed in stream channel: = 0, none; = 1, flow velocity, in m/s; = 2, discharge, in m^3/s = 3, width of water surface in the stream segment, in m; = 4, height of water in the stream segment, in m;	Yes
71	isRelationalGrade	(physicalModelInitFile)	TRUE/FALSE, Simulation Visualization, an option for using the color grades created from range of the simulated data (TRUE) or from the user defined (FALSE, use minCellV, maxCellV, minSegV and maxSegV instead);	Yes
72 73	minCellV maxCellV	(physicalModelInitFile)	Simulation Visualization .The data range for cell water infomation coloring;	Yes
74 75	minSegV maxSegV	(physicalModelInitFile)	Simulation Visualization .The data range for stream segment flow information coloring;	Yes

Table M - 32. Parameters used by the visualization model (continued).

No.	Parameter(s)	In File	Description	Required?
76 77	widthConst heightConst	(physicalModelInitFile)	Simulation Visualization .The amplifying constant to the segment width and water depth.	Yes
78 79 80	basinColor.R basinColor.G basinColor.B	(physicalModelInitFile)	Simulation Visualization . The color used for cells outside simulation watershed.	Yes
81	nGrades	(physicalModelInitFile)	Simulation Visualization .The number of color grades.	Yes
82 83 84	baseMinCellColor.R baseMinCellColor.G baseMinCellColor.B	(physicalModelInitFile)	Simulation Visualization. A darker color (smaller RGB values) used to represent highest value in the cell;	Yes
85 86 87	baseMaxCellColor.R baseMaxCellColor.G baseMaxCellColor.B	(physicalModelInitFile)	Simulation Visualization. The lighter color (larger RGB values) used to represent lowest value in the cell.	Yes
88 89 90	baseMinSegColor.R baseMinSegColor.G baseMinSegColor.B	(physicalModelInitFile)	Simulation Visualization. A darker color (smaller RGB values) used to represent highest value in the segment;	Yes
91 92 93	baseMaxSegColor.R baseMaxSegColor.G baseMaxSegColor.B	(physicalModelInitFile)	Simulation Visualization. The lighter color (larger RGB values) used to represent lowest value in the segment.	Yes
94 95	timeMajorInt timeMinorInt	(physicalModelInitFile)	The time apart for the major and minor tig of the hydrograph for 3-D dynamic visualization	Yes
96 97 98	twoDBackground.R twoDBackground.G twoDBackground.B	(physicalModelInitFile)	The color used as the background of the 2-D hydrograph outputs	Yes
99	isZeroOrigin	(physicalModelInitFile)	TRUE/FALSE, when set to true, all values except air temperature will start from zero in the 2-D hydrograph.	Yes
100 101 102	AirTempColor.R AirTempColor.G AirTempColor.B	(physicalModelInitFile)	Line RGB color for the air temperature, 0 to 255.	Yes
103 104 105	PrecipColor.R PrecipColor.G PrecipColor.B	(physicalModelInitFile)	Line RGB color for the precipitation, 0 to 255.	Yes
106 107 108	ETColor.R ETColor.G ETColor.B	(physicalModelInitFile)	Line RGB color for the evapotranspiration, 0 to 255.	Yes
109 110 111	ObservedColor.R ObservedColor.G ObservedColor.B	(physicalModelInitFile)	Line RGB color for the observed stream flow, 0 to 255.	Yes
112 113 114	SimulatedColor.R SimulatedColor.G SimulatedColor.B	(physicalModelInitFile)	Line RGB color for the simulated stream flow, 0 to 255.	Yes
115 116 117	SurfaceFColor.R SurfaceFColor.G SurfaceFColor.B	(physicalModelInitFile)	Filling RGB color for the surface flow, 0 to 255.	Yes
118 119 120	MPipeFColor.R MPipeFColor.G MPipeFColor.B	(physicalModelInitFile)	Filling RGB color for the macropore pipe flow, 0 to 255.	Yes

Table M - 32. Parameters used by the visualization model (continued).

No.	Parameter(s)	In File	Description	Required?
121 122 123	SoilFColor.R SoilFColor.G SoilFColor.B	(physicalModelInitFile)	Filling RGB color for the soil flow, ranged from 0 to 255.	Yes
124 125 126	CSnowDColor.R CSnowDColor.G CSnowDColor.B	(physicalModelInitFile)	Filling RGB color for the canopy snow depth, ranged from 0 to 255.	Yes
127 128 129	CWaterDColor.R CWaterDColor.G CWaterDColor.B	(physicalModelInitFile)	Filling RGB color for the canopy water depth, ranged from 0 to 255.	Yes
130 131 132	SSnowDColor.R SSnowDColor.G SSnowDColor.B	(physicalModelInitFile)	Filling RGB color for the surface snow depth, ranged from 0 to 255.	Yes
133 134 135	SWaterDColor.R SWaterDColor.G SWaterDColor.B	(physicalModelInitFile)	Filling RGB color for the surface water depth, ranged from 0 to 255.	Yes
136 137 138	MPipeDColor.R MPipeDColor.G MPipeDColor.B	(physicalModelInitFile)	Filling RGB color for the macropore pipe water depth, ranged from 0 to 255	Yes
139 140 141	SoilDColor.R SoilDColor.G SoilDColor.B	(physicalModelInitFile)	Filling RGB color for the soil water depth, ranged from 0 to 255	Yes

4.2.3. *Run the Visualization Program*

There are several programs in current OWLS Windows version perform the visualization:

4.2.3.1. General Display of the Watershed

4.2.3.1.1. Steady View

There are four different windows for steady view of a watersheds (Figure M - 5):

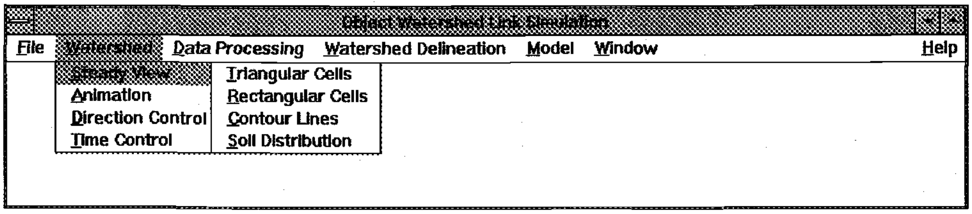


Figure M - 5. Window to run steady view of a watershed

(1) Triangular Mesh Watershed;

(2) Rectangular Mesh Watershed;

(3) Watershed Contours;

(4) Watershed Soil;

4.2.3.1.2. Watershed Animation

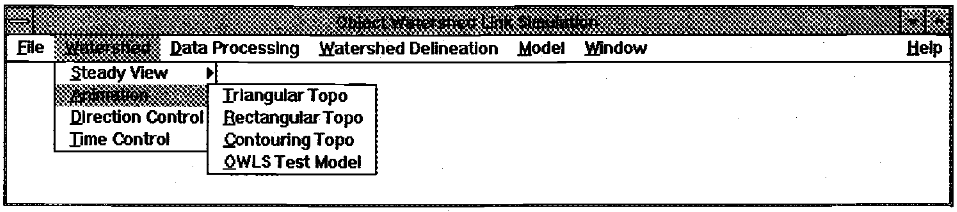


Figure M - 6. Windows to run the dynamic view of a watershed.

There are also four different windows for animation of the sky viewing of a watershed (Figure M - 6):

(1) Triangular Mesh Watershed;

(2) Rectangular Mesh Watershed;

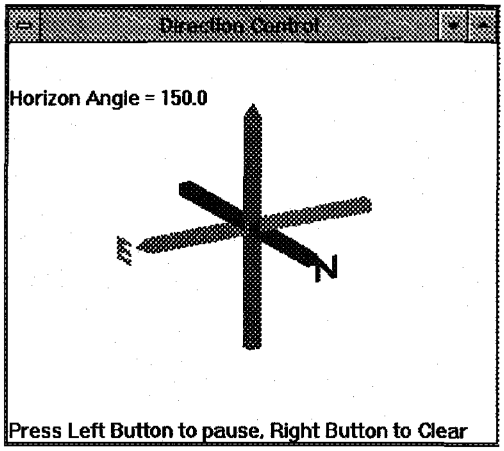
(3) Watershed Contours;

(4) OWLS Test Model, a testing ideal simple watershed;

4.2.3.1.3. Direction Control

Direction control is a rotating directional coordinate used to control the facing direction of a watershed. To activate the direction control, following the steps below:

(1). Open the windows of a three dimensional watershed



Horizon Angle = 150.0

Press Left Button to pause, Right Button to Clear

Figure M - 7. Direction Control

view, including the dynamic watershed hydrology window;

(2). Open the Direct Control window (see Figure M - 7);

(3). On the angle that you want to view the watershed, click the left buttom of your mouse, then go to the watershed window and click the right buttom to refresh the watershed view.

4.2.3.1.4. Time Control

Similar to the Direction Control, the time control displays the view of a watershed with different shading for different hour in a day. Upon the hour you desired, press the left buttom to pause the time and refresh the window for viewing a three dimentional watershed. Note: this control will not effect the dynamic watershed hydrology window.

4.2.3.2. Check the Watershed Layout

This visualization enables us to see the simulated watershed flowpaths, flowpath tree, stream tree, boundary, cells, etc., and it also enables us to compare the simulated stream channel with the digitalizated results. Just by adjusting the parameters (4 to 15 in page 45) in Table M - 30, then calling the window as shown in Figure M - 8 under the OWLS Windows Program.

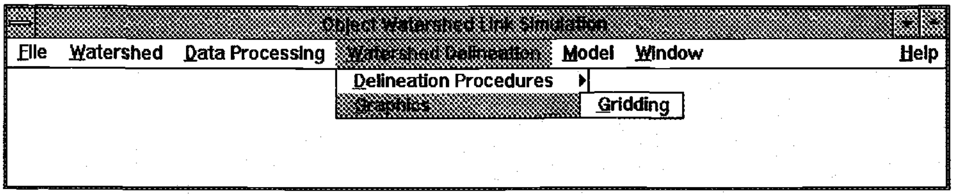


Figure M - 8. Window to visualize watershed delineation results

4.2.3.3. Simulation Results 2-D Presentation

The results from the OWLS hydrologic model can be displayed from the 2-D presention, which includes hydrograph for both simulated and observed flow, air temperature, precipitation and simulated ET, flow composition, water depth for in different vertical layers as well as the print out of watershed information and parameters used for the simulation. To run the model, call the window shown in Figure M - 9 from the OWLS window.

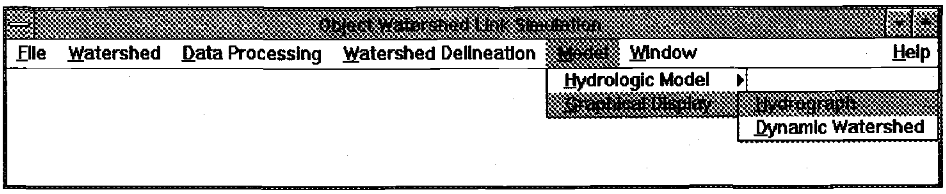


Figure M - 9. Window to run the 2-D visualization of the results from watershed hydrologic model

4.2.3.4. Simulation Result 3-D Animation

After running the OWLS hydrologic model, the simulated results can be visualized dynamically. The visualization model can spatially display the relative soil moisture content, water depths for different layers, flow generated from different layers and so on. On the stream channel, components which can be color-displayed include flow velocity, discharge, segment width, and water depth; in addition, the segment width and water depth are also displayed geometrically with a user-supplied amplification-ratio. The visualization model will display the watershed information for each time step in a continous manner so that it provides a dynamic view of a watershed's hydrology. To run the model, call the window shown in Figure M - 10 from the OWLS window.

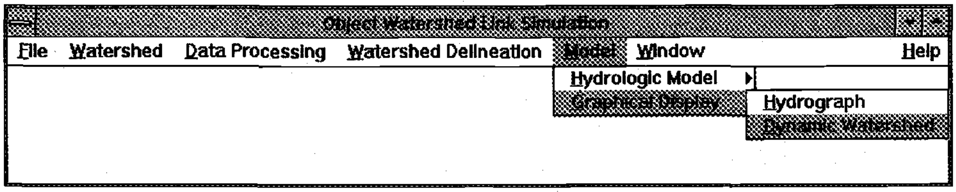


Figure M - 10. Window to run 3-D dynamic visualization of watershed hydrology

4.3. *Run the Hydrologic Model*

4.3.1. <u>Check List</u>

Files that may be used in the Hydrologic Model are listed in the Table M - 33.

Table M - 33. List of files used for the hydrologic model.

No.	Parameter	In File	Description	Required?
1	rainGageFileName	owlshydr.ini	see (4.1.2.19)	Yes
2	cloudFileName	owlshydr.ini	see (4.1.2.20)	Yes
3	LAIRatioFileNamePrefix	owlshydr.ini	see (4.1.2.21)	Yes
4	airTempGageFileName	owlshydr.ini	see (4.1.2.22)	Yes
5	soilOTempGageFileName	owlshydr.ini	see (4.1.2.23)	Not yet
6	soilBTempGageFileName	owlshydr.ini	see (4.1.2.23)	Not yet
7	soilCTempGageFileName	owlshydr.ini	see (4.1.2.23)	Not yet
8	soilTypeFileName	owlshydr.ini	see (4.1.2.24)	Yes
9	vegTypeFileName	owlshydr.ini	see (4.1.2.25)	Yes
10	streamGageFileName	owlshydr.ini	see (4.1.2.26)	Yes
11	traceTimeFileName	owlshydr.ini	A file stores the time for the currently-saved calculated records. It is a protection and a marker for continuous run after the program being brought down or interrupted.	Yes
12	monthlyAirTempFile	owlshydr.ini	see (4.1.2.27)	Yes
13	physicalModelInitFile	owlshydr.ini	see (4.1.2.28)	Yes
14	soilDepthFileName	owls.ini	see (4.1.2.8)	Yes
15	soilCellFileName	owls.ini	see (4.1.2.9)	Yes
16	newBasinFileName	owls.ini	see (4.1.3 Table No.17)	Yes
17	segmentLogFileName	owls.ini	A text log file created when messageLevel=2 or >3. It contain the coordinates of each stream segment and their water depth, discharge and velocity. The file is used for debugging purpose.	Yes
18	LAItoInterceptFileName	(physicalModelInitFile)	see (4.1.2.29)	Yes
19	m2fFilePrefix	(physicalModelInitFile)	see (4.1.2.30)	No, Optional when set <u>useHorton</u> = FALSE.
20	LAItoETFileName	(physicalModelInitFile)	see (4.1.2.31)	Yes
21	drainArea2PipeRadius	(physicalModelInitFile)	see (4.1.2.32) to (4.1.2.37)	No, Optional when available.
22	area2PipeCount			
23	volume2PipeCount			
24	area2Depth			
25	area2TopWidth			
26	area2BotWidth			

Table M - 33. List of files used for the hydrologic model (continued).

No.	Parameter	In File	Description	Required?
27	paramFileName	(physicalModelInitFile)	A temporal file to save the parameters used in the last saved calculation. It will be used after restart the program to prevent the data loss.	Yes
28	fluxFileName	(physicalModelInitFile)	A temporal file to save the fluxs of the whole watershed in the last saved calculation. It is used for faulty protection.	Yes
29	channelFileName	(physicalModelInitFile)	A binary file stores the simulated flow of the channel segment.	Yes
30 31 32 33 34 35	cellCanopyFileName cellSurfaceFileName cellSoilFileName cellMacroporeFileName cellAllFileName cellFileName	(physicalModelInitFile)	The binary files stores the cell water information of the watershed.	Yes
36	sumFluxFileName	(physicalModelInitFile)	A binary file stores the stream flow of the watershed outlet and averaged flux and water storage of the watershed.	Yes
37	counterFileName	(physicalModelInitFile)	The file stores the number of records saved.	Yes
38	textOutputFileName	(physicalModelInitFile)	The text file stores the stream flow, averaged flux and water storage of the watershed.	Yes
39	riverOutputFileName	(physicalModelInitFile)	The text file stores the flows of each segments in the watershed.	Yes
40	segmentFileName	(physicalModelInitFile)	see (4.2.1 table@24)	Yes
41	channelFileName	(physicalModelInitFile)	see (4.2.1 table@25)	Yes
42	cellFileName	(physicalModelInitFile)	see (4.2.1 table@26)	Yes
43	sumFluxFileName	(physicalModelInitFile)	see (4.2.1 table@27)	Yes
44	counterFileName	(physicalModelInitFile)	see (4.2.1 table@28)	Yes
45	basinIndexFileName	owlshydr.ini	see (4.2.1 table@29)	Yes
46	basinInfoFileName	(physicalModelInitFile)	A binary file to store watershed information and parameters	Yes

4.3.2. *Parameter Assignment*

Parameters used by the hydrologic model is shown in Table M - 34.

Table M - 34. Parameters used by the hydrologic model.

No.	Parameter	In File	Description	Required?
1	catchmentID	owlshydr.ini	The ID for the catchment when there more than one catchment available.	Yes. In BBWM, 1 is the West Bear Brook and 2 is the East Bear Brook. Note: Other parameters need to be changed too when change catchmentID!!!
2	inputUnit	owlshydr.ini & physicalModelInitFile	Input data unit. Either English or SI	Yes.
3	outoutUnit	owlshydr.ini & physicalModelInitFile	Output data unit. Either English or SI	Yes.
4	latitude	physicalModelInitFile	The latitude of the watershed center, in degree.	Yes.
5	longitude	physicalModelInitFile	The longitude of the watershed center, in dgree.	Yes.
6	turbidity (*)	physicalModelInitFile	The acerage clear day air turbidity, in %.	Yes.
7	Manning (*)	physicalModelInitFile	Manning's coefficient of the stream channel	Yes.
8	maxTempTime	physicalModelInitFile	The time of a day having maximum temperature, in hour of 24 hrs/day.	Yes.
9	avgTempDifference	physicalModelInitFile	The average difference of air temperature in a day. It will be used only when data are missing.	Yes.
10	snowSeasonBeginMMDD	physicalModelInitFile	The date of the snow season begins. It will be used when no air temperature is available.	Yes.
11	snowSeasonEndMMDD	physicalModelInitFile	The date of the snow season ends. Used only when no air temperature data available.	Yes.
12 13	startDate startTime	physicalModelInitFile	The start time for the simulation. For startDate, use YYMMDD, i,e, 901130 means Nov. 20, 1990. For startTime, use HHMMSS, i.e. 123015 means 12:35:15.	Yes.
14 15	endDate endTime	physicalModelInitFile	The time when the simulation completes. same usage as above.	Yes.
16	step	physicalModelInitFile	The calculation time interval in hour.	Yes.
17	saveInterval	physicalModelInitFile	The interval in steps that the OWLS model to save the raults.	Yes.

Table M - 34. Parameters used by the hydrologic model (continued).

No.	Parameter	In File	Description	Required?
18	useDepthOutput	physicalModelInitFile	see (2.3.1 (3))	Yes.
19	canopyMinETRate (*)	physicalModelInitFile	The canopy ET during night time.	Yes.
20	useHorton	physicalModelInitFile	see (2.3.1 (5)[1])	Yes.
21	infiltration_k	physicalModelInitFile	see (2.3.2 [1] & [2])	Yes.
22	infiltration_a			
23	infiltration0Adjust (*)	physicalModelInitFile	see (2.3.2 [4] & [5])	Yes.
24	infiltrationCAdjust (*)			
25	conductivityAdjust (*)	physicalModelInitFile	see (2.3.2 [3])	Yes.
26	snowMelt_Df (*)	physicalModelInitFile	see (2.3.2 [6] & [7])	Yes.
27	snowMelt_Tb (*)			
28	ET_a (*)	physicalModelInitFile	see (2.3.2 [8] & [9])	Yes.
29	ET_b (*)			
30	underCanopyERConstant (*)	physicalModelInitFile	see (2.3.2 [10])	Yes.
31	soilETConstant (*)	physicalModelInitFile	see (2.3.2 [11])	Yes.
32	soilWaterSupplyC1 (*)	physicalModelInitFile	see (2.3.2 [26] & [27])	Yes.
33	soilWaterSupplyC2 (*)			
34	roughness (*)	physicalModelInitFile	see (2.3.2 [12])	Yes.
35	layerWeight1 (*)	physicalModelInitFile	see (2.3.2 [13])	Yes.
36	layerWeight2 (*)			
37	minDiameter (*)	physicalModelInitFile	see (2.3.2 [14])	Yes.
38	pipeRatio (*)	physicalModelInitFile	see (2.3.2 [15])	Yes.
39	radiusA (*)	physicalModelInitFile	see (2.3.2 [16])	Yes.
40	radiusB (*)			
41	radiusC (*)			
42	countA (*)	physicalModelInitFile	see (2.3.2 [17])	Yes.
43	countB (*)			
44	countC (*)			
45	countD (*)			
46	countE (*)			
47	surfaceMcaroporeConst (*)	physicalModelInitFile	see (2.3.2 [18])	Yes.
48	soilMacroporeConst (*)	physicalModelInitFile	see (2.3.2 [19])	Yes.
49	frictionCoeff (*)	physicalModelInitFile	see (2.3.2 [20])	Yes
50	interiveErr (*)	physicalModelInitFile	see (2.3.1 (5) [3])	Yes
51	maxInterations	physicalModelInitFile	see (2.3.1 (5) [4])	Yes
52	useDirectInputs	physicalModelInitFile	see (2.3.1 (5) [2])	Yes
53	widthTopConstant (*)	physicalModelInitFile	see (2.3.2 [21] - [23])	Yes
54	widthBotConstant (*)			
55	depthConstant (*)			
56	widthTopPow (*)			
57	widthBotPow (*)			
58	depthPow (*)			
59	soilMoisture (*)	physicalModelInitFile	see (2.3.2 [24])	Yes
60	initialStreamDepthRatio (*)	physicalModelInitFile	see (2.3.2 [25])	Yes

(*) Parameters that need to be optimized or caliberated.

4.3.3. <u>*Run the Hydrologic Simulation Program*</u>

To run the Hydrologic Model from the <u>startTime</u>, choose the one of three options from the OWLS Windows program (Figure M - 11):

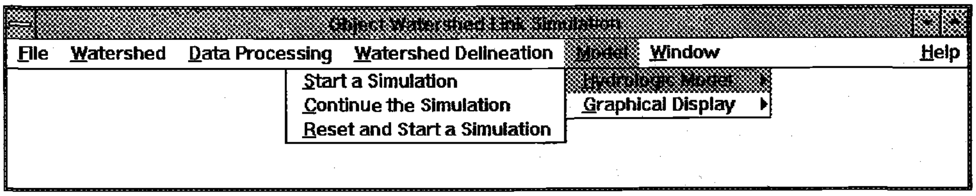


Figure M - 11. Window to run the hydrologic model.

(1). For starting the simulation the very first time, run **<u>S</u>tart a Simulation**;

(2). For continuing from the last interruption, run **<u>C</u>ontinue the Simulation**;

(3). For restart the simulation all over again, run **<u>R</u>eset and Start a Simulation**;

4.3.4. <u>*Stop the Hydrologic Simulation Program*</u>

To stop the Hydrologic Model while it is running, the user has to use Ctrl_Alt_Delete soft-interruption-keyboard-command or simply reset the computer.

4.4. *Display the Simulation Results*

4.4.1. <u>*Check List*</u>

After finishing the OWLS hydrologic program run, files for 2-D and 3-D visualization are available (Table M - 35):

Table M - 35. Files generated by the OWLS hydrologic model

No.	File Name	Description
1	channelFileName	Binary file for channel segment water storage and flow, see (3.2.2.1) for details.
2 3 4 5 6	cellCanopyFileName cellSurfaceFileName cellSoilFileName cellMacroporeFileName cellAllFileName	Binary files for cell water storage information, used by the 3-D visualization model, see (3.2.2.2) for details.
7	basinInfoFileName	Binary files for basin information and parameters, used by the 2-D presentation.
8	sumFluxFileName	Binary file for simulated flow output, see (3.1) for detailed discussion
9	counterFileName	Text file for number of records stored in the series of result files
10	textOutputFileName	Text file for simulated flow output, see (3.1) for detailed discussion
11	riverOutputFileName	Text file for simulated flow in stream segments, see (3.2.2.1) for details.

4.4.2. 2-D Display

(1) 2-D Visualization Model

The 2-D visualization model handles the graphical outputs from the OWLS hydrologic model. It displays the simulated hydrologic components as a function of time as well as the basin information and the parameters that were used for the model. It is a special design for hydrologic model, especially used for parameter calibration and result's presentation. Figures of hydrograph shown in the previous chapter are created from this model. As we can see, it includes the following information:

(1). Text information: basin name, size, simulation and system parameters. The importance of the parameters are decreasing in the order of from top to down;

(2) Air temperature curve, simulated from the daily characterization data (minimum, average and maximum);

(3) Precipitation as observed and evaportranspiration as simulated;

(4) Hydrograph for both simulated and observed flow;

(5) Simulated flow components, including surface flow, macropore pipe flow and soil flow;

(6) Water in different vertical components of the basin, including canopy intercepted water, canopy intercepted snow, surface water, surface snow, macropore pipe water and soil water.

Detailed of running the 2-D visualization model, see 4.2.3.3.

(2) Connect to spreadsheet

The task of drawing a 2-D hydrograph can be complished by utilizing a spreadsheet computer software. The text-file from the OWLS program can be imported into a spreadsheet (i.e. Microsoft Excel, Quattro Pro, or Lotus 123) and then use the spreadsheet's software to draw the hydrograph.

4.4.3. *3-D Display*

As described in section 4.2.3.4, the OWLS visualization model can provide a 3-D display of the simulated results include:

(1) Watershed Cell Hydrologic Components
  (1.1) Canopy Layer Information
    a. Intercepted Water Depth (mm);
    b. Intercepted Snow Depth (mm);
    c. Rainfall to the Canopy (mm);
    d. Canopy ET (mm);
    e. Canopy Net Precipitation (mm);
    f. Intercepted Water&Snow Depth (mm);
  (1.2) Surface Information
    a. Surface Water Depth (mm);
    b. Surface Snow Depth (mm);
    c. Surface ET (mm);
    d. Surface Infiltraion (mm);
    e. Surface Flow (0.001m3/s);
    f. Surface Water&Snow Depth (mm);

- (1.3) Subsurface Information
  - a. Soil Moisture (mm);
  - b. Soil Water Depth (mm);
  - c. Soil ET (mm);
  - d. Soil Flow (0.001m3/s);
- (1.4) Macropore Information
  - a. Macropore Water Depth (mm);
  - b. Water from Suface (mm);
  - c. Water from Soil (mm);
  - d. Flow (0.001m3/s);
  - e. Water from Suface&Soil (mm);
- (1.5) Layer Summation
  - a. Total Cell Water Depth (mm);
  - b. Total Cell Flow (0.001m3/s);

(2) Stream Segment Hydrologic Components

- (2.1) Flow velocity (m/s);
- (2.2) Discharge (m^3/s);
- (2.3) Channel width (m);
- (2.4) Water depth (m);

5. Trouble Shooting

5.1. Problem: Program abort without notice right after start.

Reasons: (1) Parameters in the initial files are either not completed or incorrectly typed.

(2) Data file does not have correct format.

5.2. Problem: Program abort without notice after running for couple of minutes.

Reasons: (1) Size of watershed mesh too big, limitation is not certain, depending upon the memory that the computer has. Reduce the numbers of cells in the watershed as well as reduce the time range for the simulation, which may have slight effect;

(2) Incorrect input data;

(3) Problems in the Program;

5.3. Problem: Program abort with General Protection Error.

Reasons: (1) Size of watershed mesh too big;

(2) Computer do not have enough resource (e.g., small hard disk space, small memory size).

(3) File(s) that are used by the OWLS model are openned by another application. For example, you may use MS Word to edit the initial file and did not close it before you launch the OWLS application.

(4) Problems in the model;

5.4. Problem: Program abort with Floating Point Overflow.

Reasons: (1) Incorrect data inputs;

(2) Abnormal topograph (flat surfuce, pond);