ELSEVIER

Environmental Modelling & Software 15 (2000) 151–160

Environmental Modelling & Software

www.elsevier.com/locate/envsoft

A unit hydrograph rainfall-runoff model using *Mathematica*

T.V. Hromadka II *

Department of Mathematics, California State University, Fullerton, CA 92634, USA

Received 29 December 1997; received in revised form 1 January 1999; accepted 12 August 1999

Abstract

Many of the most commonly used rainfall-runoff computer programs involve runoff generation and flood hydrograph routing algorithms that reduce into Toeplitz matrix systems (see Hromadka II and Whitley, 2000, ASCE Journal of Hydrolic Engineering, submitted for publication). A *link-node* model representation of the watershed is constructed by subdividing the watershed into numerous subareas, and connecting the subareas by a network of hydrologic routing links (using Muskingum, convex, convolution, or pure translation methods). The U.S. Army Corps of Engineers Hydrologic Engineering Center's computer program HEC-1, or related programs, is used to implement the details of the link-node modeling.

In contrast, the *single-area* unit hydrograph model approach represents a watershed as a single subarea, and utilizes a single unit hydrograph to represent all the effects being modeled by a link-node model representation.

The system of Toeplitz matrices developed in Hromadka II and Whitley (2000) precisely describes the *single-area* and *link-node* model structures as they are actually applied; namely, in discretized timestep unit period additions and multiplications. It was shown that the *single-area* unit hydrograph model structure similar to the *link-node* unit hydrograph model structure; namely, both structures are Toeplitz matrix systems of the same dimension. It was also shown that the calibrated *single-area* unit hydrograph model Toeplitz matrices which are developed by direct calibration to gauged data achieves the minimum variance between the model structure and the available rainfall-runoff data.

In this paper, a computer program is presented that implements the procedures presented in Hromadka II and Whitley (2000) for optimizing the unit hydrograph method.

Keywords: Unit hydrographs; Rainfall-runoff models; Runoff hydrographs; Optimization

Software availability

Name of software: MATHEMATICA for Rainfall-Runoff Model
Developer: T. V. Hromadka II
Contact address: Department of Mathematics, California State University, Fullerton, CA 92634, USA
First available: December 1997
Program language: MATHEMATICA
Hardware requirement: IBM PC or compatible
Cost: none

1. Introduction

The purpose of this paper is to document the development of a computer code which (1) calibrates the *single-area* unit hydrograph model Toeplitz matrix system (as developed in Hromadka II and Whitley, 2000) for a specific watershed based on historical rainfall-runoff data for *ns* storms, where *ns* is the number of historical rainfall-runoff storm events used to calibrate the unit hydrograph model Toeplitz matrix system; and (2) develops a Toeplitz correction matrix for each storm event contained in the data set. The calibrated *single-area* unit hydrograph model and the *ns* correction matrices can then be used as the "best estimator model" for an approximation of the stochastic distribution of runoff hydrograph realizations for an assumed rainfall event (i.e., a forecast) for the subject watershed. In Hromadka II and Whitley (2000), the mathematical underpinnings of the classical rainfall-runoff unit hydrograph method are presented, including a generalized procedure to resolve the catchment response function, or unit hydrograph. In the current paper, a *Mathematica* code is presented to implement the procedures developed in Hromadka II and Whitley (2000).

* Tel.: +1-714-755-5380; fax: +1-714-979-9975.

1364-8152/00/$ - see front matter
PII: S1364-8152(99)00030-4

A format for the rainfall-runoff data for the *ns* storms is described. The computer program reads the input data, organizes and stores the data, and calculates the necessary model matrices and vectors as output.

Mathematica Version 3 is well suited for all these tasks, and was selected for this computer code application. All specific built-in *Mathematica* code objects used are described in Wolfram (1996).

The specific mathematical equations and operations which are used are described in Section 2. The developed computer code and instructions for its use are provided throughout the text. The input and output for a test case with three storms is also provided in the text of this paper. The upfront code needed to manage the program is given in Appendix A.

2. Discussion

The basic equation, for each storm *i*, used to calculate the "best estimator model" matrices is:

$$\underset{\sim}{Q}^i=[T^i][U_0]\underset{\sim}{P}^i_g-[F_M]\underset{\sim}{\delta}^i_{k_i}+\underset{\sim}{B} \quad (1)$$

where $\underset{\sim}{Q}^i$ is the vector representation (a $nt\times1$ column vector) of the measured storm runoff for storm *i*; $[T^i]$ is a $nt\times nt$ Toeplitz correction matrix for storm *i* runoff estimate, $\underset{\sim}{Q}^i_M$; $[U_0]$ is a single-area calibrated unit hydrograph model $nt\times nt$ Toeplitz matrix structure (the same for each storm); $\underset{\sim}{P}^i_g$ is a vector representation (a $nt\times1$ column vector) of the gauged precipitation for storm *i*; $[F_M]$ is a $nt\times nt$ Toeplitz matrix representation of the watershed losses (the same for each storm). In this program, the watershed loss function is assumed to be a distributed phi-index loss (ϕ); [see Hromadka II and Whitley (1989) for details]; $\underset{\sim}{B}$ is a vector representation (a $nt\times1$ column vector) of the base flow, which is the same for each storm, $\underset{\sim}{B}=b_o\underset{\sim}{\delta}_{nt}$; $\underset{\sim}{\delta}^i_{k_i}$ is a unit pulse vector (a $nt\times1$ column vector) for storm *i*, where the values of rows 1 to k_i are 1, and the values of rows k_i+1 to *nt* are 0, and k_i is the non-zero length of the rainfall precipitation vector of storm *i*.

In this paper, the base flow (b_o and $\underset{\sim}{B}$) are assumed to be zero, where b_o is a selected constant baseflow (assumed to be zero in this report). The $\underset{\sim}{Q}^i_M$, $\underset{\sim}{P}^i_g$, and $\underset{\sim}{\delta}^i_{k_i}$ column vectors, each having a standardized length of *nt*, are formed from the input data as discussed in Section 3. In the above, *nt* is the length of the longest precipitation/runoff input record in all of the storm events used to calibrate the unit hydrograph model Toeplitz matrix system. A key assumption is made that the elements of the $\underset{\sim}{P}^i_g$ vectors are greater or equal to the ϕ index loss function so that the effective rainfall (rainfall less losses) has positive and nonzero values.

The $[T^i]$ matrix is initially set to be the identity matrix for the calibration of the "best estimator model". Calibrated $[U_0]$, $[F_M]$ matrices (which are held identical for each storm), are computed during this step. The least squares minimization is performed simultaneously with respect to all storm events and the *nt* components of both $[U_0]$ and $[F_M]$, respectively, using a singular-value decomposition method. The details of this process are discussed in the Section 4.

The $[T^i]$ correction matrix is then computed individually for each storm by solving Eq. (1), using a forward substitution method. The details of this step are addressed in Section 5.

The set of calculated $[U_0]$, $[F_M]$, and $[T^i]$ matrices collectively define the "best estimator model" for the subject watershed. The use of this model is discussed in Section 6. Each of these model matrices has the form of $nt\times nt$ lower triangular Toeplitz matrix:

$$[A]=\begin{pmatrix} a_1 & 0 & 0 & \cdots & 0 \\ a_2 & a_1 & 0 & \cdots & 0 \\ a_3 & a_2 & a_1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & 0 \\ a_{nt} & a_{nt-1} & a_{nt-2} & \cdots & a_1 \end{pmatrix} \quad (2)$$

where the a_i are real constants; $i=1, 2, 3, \ldots, nt$.

Lower triangular Toeplitz matrices have the following properties:

(a) If *A* and *B* are $nt\times nt$ Toeplitz matrices and *a* and *b* are scalars, then $aA+bB$ is also a Toeplitz matrix.
(b) The product of two $nt\times nt$ Toeplitz matrices is a Toeplitz matrix, and the product is commutative.
(c) If a Toeplitz matrix is nonsingular, then its inverse is a Toeplitz matrix.
(d) If *A* is a Toeplitz matrix of order $nt\times nt$ and $\underset{\sim}{B}$ is a column vector of the following form:

$$\underset{\sim}{B}=\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \cdots \\ \cdots \\ b_{nt} \end{pmatrix}, \text{ then } A\cdot B=a_1\begin{pmatrix} b_1 \\ b_1 \\ b_2 \\ \cdots \\ \cdots \\ b_{nt} \end{pmatrix}+a_2\begin{pmatrix} 0 \\ b_1 \\ b_2 \\ \cdots \\ \cdots \\ b_{nt-1} \end{pmatrix} \quad (3)$$

$$+a_3\begin{pmatrix} 0 \\ 0 \\ b_1 \\ \cdots \\ \cdots \\ b_{nt-2} \end{pmatrix}+\ldots+a_{nt}\begin{pmatrix} 0 \\ 0 \\ 0 \\ \cdots \\ \cdots \\ b_1 \end{pmatrix}$$

(e) If *A*, *B*, and *U* are Toeplitz matrices of the same order and $A=U\cdot B$, then $\underset{\sim}{A}=U\cdot\underset{\sim}{B}$ where $\underset{\sim}{A}$ and $\underset{\sim}{B}$ are the first column vectors of *A* and *B*, respectively.

3. Computer program: reading and processing of input

Rainfall-runoff data for a number of storms, *ns*, are used to calibrate the model for a specific watershed. A sample input for a three-storm calibration (see Table 1), along with the corresponding *Mathematica* code, is as follows:

3.1. Test case (with three storms)

```
Preliminary setupTable 1
SetDirectory ["C:\My Documents"]
C: \My Documents
!! stormtest1.txt
Storms 3
Begin storm 1 Intervals 8
Interval      Precip      Runoff
1             2.          0
2             5           2
3             10          3
4             4           6
5             1           4
6             0           2
7             0           1
8             0           1
End Storm 1
Begin storm 2 Intervals 6
Interval      Precip      Runoff
1             1           0
2             3           0
3             4           1
4             2           3
5             0           2
6             0           2
End Storm 2
Begin Storm 3 Intervals 7
Interval      Precip      Runoff
1             1           0
2             6           0
3             6           1
4             3           4
5             0           5
6             0           2
7             0           1
End Storm 3
End of File
```

3.2. Computer code and output

```
nt=8;
Stormdata=OpenRead["stormtest1.txt"]
Skip[stormdata, Word]; ns=Read[stormdata, Number];

pg=Table[0, {i, ns}, {j, nt}];
qm=Table[0, {i, ns}, {j, nt}];
kri=Table[0, {ns}];
fmv=Table[0, {i, ns}, {j, nt}];
For[h=1, h≤ns, h++,
Skip[stormdata, {Word, Word}]; st=Read[stormdata,
Number];
Skip[stormdata, Word]; ni=Read[stormdata, Number];
Skip[stormdata, {Word, Word, Word}];
temp=ReadList[stormdata, {Number, Number, Number},
ni];
Skip[stormdata, {Word, Word, Number}];
trantemp=Transpose [temp];
pg[[h]]=Flatten[trantemp[[2]]];
For[i=ni+1, i≤nt, i++, pg[[h]]=Append[pg[[h]], 0]];
pg[[h]]=Flatten[pg[[h]]];
lis=pg[[h]]; mar=h; kr=1;
  Label[beginning]; If[lis[kr]]≠0, kr=kr+1, Goto[next-
  line]];
If [kr≤nt, Goto[beginning], Goto[nextline]];
Label[nextline]; kri[[mar]]=kr−1;
qm[[h]]=Flatten[trantemp[[3]]];
For[i=ni+1, i≤nt, i++, qm[[h]]=Append[qm[[h]],0]];
```

Table 1
Three-storm calibration test data[a]

Begin storm 1			Begin storm 2			Begin storm 3		
Interval	Precip	Runoff	Interval	Precip	Runoff	Interval	Precip	Runoff
1	2.	0.	1	1.	0.	1	1.	0.
2	5.	2.	2	3.	0.	2	6.	0.
3	10.	3.	3	4.	1.	3	6.	1.
4	4.	6.	4	2.	3.	4	3.	4.
5	1.	4.	5	0.	2.	5	0.	5.
6	0.	2.	6	0.	2.	6	0.	2.
7	0.	1.	End storm 2			7	0.	1
8	0.	1.				End storm 3		
End storm 1								

[a] The actual input format uses a single column with each storm listed sequentially as shown in Section 3.1.

```
qm[[h]]=Flatten[qm[[h]]]];
Close[stormdata];
toeplitzexp[a_]:=
  Module[{bt, n}, n=Length[a]; bt=IdentityMatrix[n];
  bt[[1]]=Flatten[a];
For[k=2, k≤n, k++,
     bt[[k]]=Delete[bt[[k-1]], -1];
       bt[[k]]=Prepend[bt[[k]], 0];
       bt[[k]]=Flatten[bt[[k]]]];
Transpose[bt]];

cronk[t_]:=Module[{ }, ckv=Table[1, {nt}];
For[i=t+1, i≤nt, i++, ckv=ReplacePart[ckv, 0, i]]];

pushdown[a_,c_]:=
  Module[{pdt, n, k}, n=Length[a]; pdt=Table[0,{nt}];
  pbt=Flatten[a];
For[k=1, k≤c, k++,
  pbt=Delete[pbt, -1]; pbt=Prepend[pbt, 0]]; pbt=Flat-
  ten[pbt]];

qmgsv=Flatten[Table[qm[[i]], {i,ns}]];
mgs=ns nt; ngs=2nt;
agsvtran=Table[0,{i,ngs}, {j,mgs}];

For[i=1, i≤nt, i++, agsvtran[[i]]=
  Flatten[Table[pushdown[pg[[j]], i-1], {j, ns}]]];

For[i=1, i≤ns, i++, c=kri[[i]]; fmv[[i]]=Table[-1, {nt}];
For[kd=1, kd≤nt-c, kd++, fmv[[i]]=Delete[fmv[[i]],
-1]]; For[ka=1, ka≤nt-c,
  ka++, fmv[[i]]=Append[fmv[[i]], 0]]; fmv[[i]]=Flat-
  ten[fmv[[i]]]];

For[i=nt+1, i≤2nt, i++, agsvtran[[i]]=
  Flatten[Table[pushdown[fmv[[j]], i-nt-1], {j, ns}]]];

agsv=Transpose[agsvtran];
a=agsv; at=Transpose[a]; b=qmgsv;
{u, md, v}=SingularValues[a]; ut=Transpose[u];
apsu=PseudoInverse[a];
xcalc=apsu.b;
fov=Take[xcalc, -nt];
fo=toeplitzexp[fov];

tivcalc=Table[0, {i, ns}, {j, nt}];
For[st=1, st≤ns, st++, k=kri[[st]]; cronk[k];
xv=qm[[st]]+(fo.ckv); nt=Length[xv];
yv=uo.pg[[st]];
yr=Delete[yv,1]; yr=Flatten[Append[yr, 0]];
uvcalc=Table[0, {i nt}, {j, 1}];

uvcalc[[1]]=xv[[1]]/yv[[1]]; utoep=toeplitzexp[uvcalc];

For[s=2, s≤nt, s++,
  Uvcalc[[s]]=N[(xv[[s]]-(utoep[[s-1]].yr))/yv[[1]]];
  utoep=toeplitzexp[uvcalc]];

tivcalc[[st]]=uvcalc];
tivcalctran=Transpose[tivcalc];

Print["Output"];
Print["\n\n U-MATRIX ELEMENTS"]; Print[Matrix-
Form[uov]];
Print["\n\n F-MATRIX ELEMENTS"]; Print[Matrix-
Form[fov]];
Print["\n\n"];
For[i=1, i≤ns, i++, Print["T-MATRIX ELEMENTS:
STORM", i];
  Print[MatrixForm[tivcalc[[i]]]]];
```

The unit time period for each storm record must be identical (such as 3 min). The dimensional units of the rainfall and runoff should be consistent (e.g., inches). Additionally, the initial precipitation value of each storm record must be nonzero. The remaining input parameter which must be input into the program is the value of *nt*, the greatest number of intervals in the rainfall-runoff data sets. For the sample input value above, the value of *nt* is 8.

The program reads the values of *ns* and *nt*, and the precipitation and runoff data for each storm. Zero elements are added, as needed, to each precipitation and runoff vector to produce the $\underset{\sim}{P}_g^i$ and $\underset{\sim}{Q}^i$ vectors, respectively, which have a standardized length equal to *nt*. The values of k_i (the non-zero length of the rainfall precipitation vector) and $\underset{\sim}{\delta}_{k_i}^i$ (the $nt{\times}1$ unit pulse vector for storm *i*) are also computed for each storm and stored.

For the sample input above, the value of k_i is 5, 4, and 4 for Storms 1, 2, and 3, respectively. Thus, recalling nt=8,

$$\underset{\sim}{\delta}_{k_i}^i=\begin{pmatrix}1\\1\\1\\1\\1\\0\\0\\0\end{pmatrix}\text{ for storm 1, and }\underset{\sim}{\delta}_{k_i}^i=\begin{pmatrix}1\\1\\1\\1\\0\\0\\0\\0\end{pmatrix}\text{ for storms 2 and 3}$$

4. Least squares calibration

To calibrate the "best estimator", the [*T*] matrix is initially set to be an identity matrix. For this report, the base flow (b_o and $\underset{\sim}{B}$) are assumed to be zero. Therefore, for calibration Eq. (1) for each storm *i* can be simplified as follows:

$$\underset{\sim}{Q}^i_{\mathrm{M}}=[U_0]\underset{\sim}{P}^i_{\mathrm{g}}-[F_{\mathrm{M}}]\underset{\sim}{\delta}^i_{k_i} \tag{4}$$

The $[U_0]$ and $[F_{\mathrm{M}}]$ matrices each have the form of a $nt\times nt$ lower triangular Toeplitz matrix. Specifically,

$$[U_0]=\begin{pmatrix} u_1 & 0 & 0 & \cdots\cdots\cdots & 0 \\ u_2 & u_1 & 0 & \cdots\cdots\cdots & 0 \\ u_3 & u_2 & u_1 & \cdots\cdots\cdots & 0 \\ \cdots & \cdots & \cdots & \cdots\cdots\cdots & 0 \\ \cdots & \cdots & \cdots & \cdots\cdots\cdots & \cdots \\ u_{nt} & u_{nt-1} & u_{nt-2} & \cdots\cdots\cdots & u_1 \end{pmatrix}, \text{ and} \tag{5}$$

$$[F_{\mathrm{M}}]=\begin{pmatrix} f_1 & 0 & 0 & \cdots\cdots\cdots & 0 \\ f_2 & f_1 & 0 & \cdots\cdots\cdots & 0 \\ f_3 & f_2 & f_1 & \cdots\cdots\cdots & 0 \\ \cdots & \cdots & \cdots & \cdots\cdots\cdots & 0 \\ \cdots & \cdots & \cdots & \cdots\cdots\cdots & \cdots \\ f_{nt} & f_{nt-1} & f_{nt-2} & \cdots\cdots\cdots & f_1 \end{pmatrix}$$

Thus, using property (d) for Toeplitz matrices, Eq. (4) for each storm i can be rewritten as follows:

$$\underset{\sim}{Q}^i_{\mathrm{M}}=u_1\begin{pmatrix} p_{i,1} \\ p_{i,2} \\ p_{i,3} \\ \cdots \\ \cdots \\ p_{i,nt} \end{pmatrix}+u_2\begin{pmatrix} 0 \\ p_{i,1} \\ p_{i,2} \\ \cdots \\ \cdots \\ p_{i,nt-1} \end{pmatrix}+u_3\begin{pmatrix} 0 \\ 0 \\ p_{i,1} \\ \cdots \\ \cdots \\ p_{i,nt-2} \end{pmatrix}+\ldots \tag{6}$$

$$+u_{nt}\begin{pmatrix} 0 \\ 0 \\ 0 \\ \cdots \\ \cdots \\ p_{i,1} \end{pmatrix}+f_1\begin{pmatrix} -\underset{\sim}{\delta}^i_{k,1} \\ -\underset{\sim}{\delta}^i_{k,2} \\ -\underset{\sim}{\delta}^i_{k,3} \\ \cdots \\ \cdots \\ -\underset{\sim}{\delta}^i_{k,nt} \end{pmatrix}+f_2\begin{pmatrix} 0 \\ -\underset{\sim}{\delta}^i_{k,1} \\ -\underset{\sim}{\delta}^i_{k,2} \\ \cdots \\ \cdots \\ -\underset{\sim}{\delta}^i_{k,nt-1} \end{pmatrix}$$

$$+f_3\begin{pmatrix} 0 \\ 0 \\ -\underset{\sim}{\delta}^i_{k,1} \\ \cdots \\ \cdots \\ -\underset{\sim}{\delta}^i_{k,nt-2} \end{pmatrix}+\ldots+f_{nt}\begin{pmatrix} 0 \\ 0 \\ 0 \\ \cdots \\ \cdots \\ -\underset{\sim}{\delta}^i_{k,1} \end{pmatrix}$$

The $\underset{\sim}{Q}^i_{\mathrm{M}}$ and $\underset{\sim}{P}^i_{\mathrm{g}}$ vectors in Eqs. (4) and (6) are the standardized length runoff and rainfall vectors for the ith storm. The minus signs in the column vectors on the right side of Eq. (1) account for the minus sign of the second term in Eq. (4). The $-\underset{\sim}{\delta}^i_{k,1}$ notation, for example, refers to row 1 of the vector $\underset{\sim}{\delta}^i_{k_i}$.

Let <**N**> denote the "best estimator" model parameter set: $u_1, u_2, \ldots, u_{nt}$ and $f_1, f_2, \ldots, f_{nt}$. This parameter set is the first column vector components of the $[U_0]$ and $[F_{\mathrm{M}}]$ Toeplitz matrices, and as such they fully define the $[U_0]$ and $[F_{\mathrm{M}}]$ matrices. The least squares minimization which solves for <**N**> is performed simultaneously with respect to all ns storm events.

To set up the least squares solution two matrices are constructed, namely a and b. The a matrix has $2\times nt$ columns, and its first nt rows consist of the storm 1 column vectors on the right side of Eq. (6); the next nt rows consist of the storm 2 column vectors on the right side of Eq. (6), etc. for all ns storms.

The overall a matrix therefore has $ns\times nt$ rows and $2\times nt$ columns (for test case 2, this amounts to 540 rows and 36 columns). The b matrix has one column, and its first nt rows consist of the storm 1 $\underset{\sim}{Q}^i_{\mathrm{M}}$ column vector on the left side of Eq. (4); the next nt rows consist of the storm 2 $\underset{\sim}{Q}^i_{\mathrm{M}}$ column vector on the left side of Eq. (4), etc. for all ns storms. The overall b matrix therefore has $ns\times nt$ rows and one column (for test case 2, this amounts to 540 rows and one column). The a and b matrices for our test case illustrate the results of this construction technique and are shown below.

Matrix a

MatrixForm[a]

2	0	0	0	0	0	0	0	−1	0	0	0	0	0	0	0
5	2	0	0	0	0	0	0	−1	−1	0	0	0	0	0	0
10	5	2	0	0	0	0	0	−1	−1	−1	0	0	0	0	0
4	10	5	2	0	0	0	0	−1	−1	−1	−1	0	0	0	0
1	4	10	5	2	0	0	0	−1	−1	−1	−1	−1	0	0	0
0	1	4	10	5	2	0	0	0	−1	−1	−1	−1	−1	0	0
0	0	1	4	10	5	2	0	0	0	−1	−1	−1	−1	−1	0
0	0	0	1	4	10	5	2	0	0	0	−1	−1	−1	−1	−1
1	0	0	0	0	0	0	0	−1	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	−1	−1	0	0	0	0	0	0
4	3	1	0	0	0	0	0	−1	−1	−1	0	0	0	0	0
2	4	3	1	0	0	0	0	−1	−1	−1	−1	0	0	0	0
0	2	4	3	1	0	0	0	0	−1	−1	−1	−1	0	0	0
0	0	2	4	3	1	0	0	0	0	−1	−1	−1	−1	0	0
0	0	0	2	4	3	1	0	0	0	0	−1	−1	−1	−1	0
0	0	0	0	2	4	3	1	0	0	0	0	−1	−1	−1	−1
1	0	0	0	0	0	0	0	−1	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	−1	−1	0	0	0	0	0	0
6	6	1	0	0	0	0	0	−1	−1	−1	0	0	0	0	0
3	6	6	1	0	0	0	0	−1	−1	−1	−1	0	0	0	0
0	3	6	6	1	0	0	0	0	−1	−1	−1	−1	0	0	0
0	0	3	6	6	1	0	0	0	0	−1	−1	−1	−1	0	0
0	0	0	3	6	6	1	0	0	0	0	−1	−1	−1	−1	0
0	0	0	0	3	6	6	1	0	0	0	0	−1	−1	−1	−1

Matrix b
MatrixForm[b]

$$\begin{pmatrix} 0 \\ 2 \\ 3 \\ 6 \\ 4 \\ 2 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 3 \\ 2 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 4 \\ 5 \\ 2 \\ 1 \\ 0 \end{pmatrix}$$

P_g vectors and k_i values
MatrixForm[pg]
MatrixForm[kri]

$$\begin{pmatrix} 2 & 5 & 10 & 4 & 1 & 0 & 0 & 0 \\ 1 & 3 & 4 & 2 & 0 & 0 & 0 & 0 \\ 1 & 6 & 6 & 3 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 5 \\ 4 \\ 4 \end{pmatrix}$$

T-MATRIX ELEMENTS: STORM 1

$$\begin{pmatrix} 0.477971 \\ 3.84049 \\ -9.07767 \\ 6.9524 \\ 15.9601 \\ -53.1656 \\ 36.5777 \\ 146.075 \end{pmatrix}$$

T-MATRIX ELEMENTS: STORM 2

$$\begin{pmatrix} 0.955942 \\ 0.798544 \\ -0.956656 \\ -1.75624 \\ 5.57011 \\ -5.17108 \\ -5.15017 \\ 27.6918 \end{pmatrix}$$

T-MATRIX ELEMENTS: STORM 3

$$\begin{pmatrix} 0.955942 \\ -2.06928 \\ 11.9428 \\ -58.4268 \\ 289.87 \\ -1428.35 \\ 7009.62 \\ -34364.8 \end{pmatrix}$$

U-MATRIX ELEMENTS

$$\begin{pmatrix} 0.312282 \\ 0.284779 \\ 0.0360325 \\ 0.292765 \\ -0.00308689 \\ 0.112265 \\ -0.0270048 \\ -2.49743 \end{pmatrix}$$

F-MATRIX ELEMENTS

$$\begin{vmatrix} 0.298524 \\ 1.11865 \\ 0.304837 \\ -2.66147 \\ 0.0325721 \\ 2.24125 \\ 0.976995 \\ -5.31615 \end{vmatrix}$$

```
For[k=2, k≤n, k++,
  bt[[k]]=Delete[bt[[k−1]], −1];
    bt[[k]]=Prepend[bt[[k]], 0];
      bt[[k]]=Flatten[bt[[k]]]];
Transpose[bt]];
cronk[t_]:=Module[{}, ckv=Table[1, {nt}];
For[i=t+1, i≤nt, i++, ckv=ReplacePart[ckv, 0, i]]];
pushdown[a_, c_]:=
  Module[{pdt, n, k}, n=Length[a]; pdt=Table[0,
  {nt}]; pdt=Flatten[a];
For[k=1, k≤c, k++,
  Pbt=Delete [pbt, −1]; pbt=Prepend[pbt,0]]; pbt=Flatten[pbt]];

qmgsv=Flatten[Table[qm[[i]], {i, ns}]];
mgs=ns nt; ngs=2 nt;
agsvtran=Table[0, {i, ngs}, { j, mgs}];

For[i=1, i≤nt, i++, agsvtran [[i]]=
  Flatten[Table[pushdown[pg[[j]], i−j], {j, ns}]]];

For[i=1, i≤ns, i++, c=kri[[i]]; fmv[[i]]=Table[−1, {nt}];
For[kd=1, kd≤nt−c, kd++, fmv[[i]]=Delete[fmv[[i]],
−1]]; For[ka=1, ka≤nt−c,
ka++, fmv[[i]]=Append[fmv[[i]], 0] ]; fmv[[i]]=Flatten[fmv[[i]]]];

For[i=nt+1, i≤2 nt, i++, agsvtran[[i]]=
  Flatten[Table[pushdown[fmv[[j]], i−nt−1], {j, ns}]]];

agsv=Transpose[agsvtran];
a=agsv; at=Transpose[a]; b=qmgsv;
{u, md, v}=SingularValues[a]; ut=Transpose[u];
apsu=PseudoInverse[a];
xcalc=apsu.b;
uov=Take[xcalc, nt];
uo=toeplitzexp[uov];
fov=Take[xcalc, −nt];
fo=toeplitzexp[fov];
tivcalc=Table[0, {i, ns}, {j, nt}];
For[st=1, st≤ns, st++, k=kri[[st]]; cronk[k];
xv=qm[[st]]+(fo.ckv); nt=Length[xv];
yv=uo.pg[[st]];
yr=Delete[yv, 1]; yr=Flatten[Append[yr, 0]];
uvcalc=Table[0, {i, nt}, {j, 1}];
```

$$\text{uvcalc}[[1]]=\frac{xv[[1]]}{yv[[1]]};\ \text{utoep}=\text{toeplitzexp}[\text{uvcalc}];$$

For[s=2, s≤nt, s++,

$$\text{uvcalc}[[s]]=N\left[\frac{xv[[s]]-(utoep[[s-1]]yr)}{yv[[1]]}\right];\ \text{utoep}$$

$$=\text{toeplitzexp}[\text{uvcalc}]];$$

The built-in function Singular Values decomposes the a matrix into three factored matrices (u^{T}, m_{D}, and v), where u^{T} is the transpose of u and m_{D} is a square diagonal matrix with singular valued elements. The built-in function pseudo inverse then constructs a^{-1}, the pseudo inverse of a, using the following equation:

$$a^{-1}=v^{\mathrm{T}}m_{\mathrm{D}}^{-1}u \tag{7}$$

where m_{D}^{-1} is the inverse of m_{D}.

Then

$$<\mathbf{N}>=a^{-1}b \tag{8}$$

The unique, calibrated $[U_0]$ matrix and the $[F_{\mathrm{M}}]$ matrix defined by the set $<\mathbf{N}>$ is used for each storm in the determination of correction matrices as described in the next section.

5. Computation of correction matrices

Eq. (1), with B=0, is then used individually for each storm event to calculate the $[T^i]$ correction matrix for the ith storm; i=1, 2, ..., ns. Namely,

$$\underset{\sim}{Q}^i=[T^i][U_0]\underset{\sim}{P}_{\mathrm{g}}^i-[F_{\mathrm{M}}]\underset{\sim}{\delta}_{k_i}^i \tag{9}$$

The $\underset{\sim}{Q}^i$, $\underset{\sim}{P}_{\mathrm{g}}^i$, $\underset{\sim}{\delta}_{k_i}^i$ vectors in Eq. (9) are the standardized length runoff, rainfall, and unit pulse vectors for the ith storm which were formed from the historical storm data input. The $[U_0]$ Toeplitz matrix and the $[F_{\mathrm{M}}]$ Toeplitz matrix computed in the previous subsection are used for each storm. Rearrangement of Eq. (9) yields:

$$\underset{\sim}{Q}^i+[F_{\mathrm{M}}]\underset{\sim}{\delta}_{k_i}^i=[T^i][U_0]\underset{\sim}{P}_{\mathrm{g}}^i \tag{10}$$

Letting $\underset{\sim}{A}^i$=$\underset{\sim}{Q}^i$+$[F_{\mathrm{M}}]$, $\underset{\sim}{\delta}_{k_i}^i$ and $\underset{\sim}{B}^i$=$[U_0]\underset{\sim}{P}_{\mathrm{g}}^i$ then $\underset{\sim}{A}^i$=$\underset{\sim}{B}^i$ can readily be calculated for each storm as their elements are known, and the above equation can be rewritten as:

$$\underset{\sim}{A}^i=[T^i]\underset{\sim}{B}^i \tag{11}$$

As discussed in Hromadka II and Whitley (2000), the $[T]$ correction matrix for each storm is a Toeplitz matrix, and the $[T^i]$ matrix is completely defined by its first column vector, $\underset{\sim}{T}^i$.

$[\underset{\sim}{T}^i]$, in Eq. (11), can be solved for each storm using forward substitution with the following iterative algorithm:

$$\text{For } s=1\ \underset{\sim}{T}_1^i=\left(\frac{\underset{\sim}{A}_1^i}{\underset{\sim}{B}_1^i}\right) \tag{12}$$

$$\text{For } s=2,\ldots,nt\ \underset{\sim}{T}_s^i=\frac{\underset{\sim}{A}_s-(\underset{\sim}{T}_{s-1}^i\underset{\sim}{B}_r^i)}{\underset{\sim}{B}_1^i} \tag{13}$$

where $\underset{\sim}{T}_s^i$ is the Toeplitz expansion of $\underset{\sim}{T}_{s-1}^i$ (from the prior iteration), and $\underset{\sim}{B}_r^i$ is $\underset{\sim}{B}^i$ with the first element deleted and a zero appended as the last element. All $[T]$ matrices are calculated in this manner.

The set of calculated $[U_0]$, $[F_M]$, and $[T]$ matrices collectively define the "best estimator model" for the subject watershed. The *Mathematica* code for the above process is given below.

5.1. Description of the computer code

All specific built-in *Mathematica* code objects used are completely described in Wolfram (1996). Each built-in *Mathematica* code object begins with a capital letter, whereas all variable names used in the computer code begin with a lower case letter.

The following portion of the computer code reads in the values of *ns* and *nt*, as well as the precipitation and runoff data for each storm. Zero elements are added, as needed, to each precipitation and runoff vector to produce the $\underset{\sim}{P}_g^i$ and $\underset{\sim}{Q}_M^i$ vectors, respectively. Each of these vectors has a standardized length equal to *nt*. The values of k_i (the non-zero length of the rainfall precipitation vector) and $\underset{\sim}{\delta}_{k_i}^i$ (the nt×1 unit pulse vector for storm i) are also computed for each storm and stored.

```
nt=8;
Stormdata= OpenRead["historicstormdata1.txt"]
Skip[stormdata, Word]; ns=Read[stormdata, Number];

pg=Table[0, {i, ns}, {j, nt}];
qm=Table[0, {i, ns}, {j, nt}];
kri=Table{0, {ns}];
fmv=Table[0, {i, ns}, {j, nt}];
For[h=1, h≤ns, h++,
Skip[stormdata, {Word, Word}]; st=Read[stormdata, Number];
Skip[stormdata, Word]; ni=Read[stormdata, Number];
Skip[stormdata, {Word, Word, Word}];
temp=ReadList[stormdata, {Number, Number, Number} ni];
Skip[stormdata, {Word, Word, Number}];
trantemp=Transpose[temp];
pg[[h]]=Flatten[trantemp[[2]]];
For[i=ni+1, i≤nt, i++, pg[[h]]=Append[pg[h]], 0]];
pg[[h]]=Flatten[pg[[h]]];
lis=pg[[h]]; mar=h; kr=1;
  Label[beginning]; If[lis[kr]]≠0, kr=kr+1, Goto[nextline]];
If[kr≤nt, Goto[beginning], Goto[nextline]];

Label[nextline]; kri[[mar]]=kr−1;
qm[[h]]=Flatten[trantemp[[3]]];
For[i=ni+1, i≤nt, i++, qm[[h]]=Append[qm[[h]],0]];
qm[[h]]=Flatten[qm[[h]]]];
Close[stormdata];
```

The following portion of the computer code defines three program patterns which perform the following functions, respectively:

1. expand a column vector to form its associated Toeplitz matrix (the original column vector becomes the first column vector of its Toeplitz matrix);
2. compute $\underset{\sim}{\delta}_{t_i}^i$ (the next nt×1 unit pulse vector for storm i) for an input value of t_i;
3. reforms a column vector by deleting its last c elements and adding c elements with value 0 at the top of the vector.

These patterns are used in the subsequent computer code sections.

```
Teoplitzexp[a_]:=
Module[{bt,n}, n=Length[a]; bt=IdentityMatrix[n];
bt[[1]]=Flatten[a];
For[k=2, k≤n, k++,
  bt[[k]]=Delete[bt[[k−1]], −1];
    bt[[k]]=Prepend[bt[[k]], 0];
      bt[[k]]=Flatten[bt[[k]]]];
Transpose[bt]];

cronk[t_]:=Module[{ }, ckv=Table[1, {nt}];
For[i=t+1, i<nt, i++, ckv=ReplacePart[ckv, 0, i]]];

pushdown[a_,c_]:=
  Module[{pdt, n, k}, n=Length[a]; pdt=Table[0,{nt}];
  pbt=Flatten[a];
  For[k=1, k≤c, k++,
pbt=Delete[pbt, −1]; pbt=Prepend[pbt, 0]]; pbt=Flatten[pbt]];
```

The following portion of the code computes matrices *a* and *b* which are used to set up the least squares ›lution.

```
qmgsv=Flatten[Table[qm[[i]], {i,ns}]];
mgs=ns nt; ngs=2nt;
agsvtran=Table[0,{i,ngs}, {j,mgs}];

For[i=1, i≤nt, i++, agsvtran[[i]]=
  Flatten[Table[pushdown[pg[[j]], i−1], {j, ns}]]];

For[i=1, i≤ns, i++, c=kri[[i]]; fmv[[i]]=Table[−1, {nt}];
For[kd=1, kd≤nt−c, kd++, fmv[[i]]=Delete[fmv[[i]],
−1]]; For[ka=1, ka≤nt−c,
  ka++, fmv[[i]]=Append[fmv[[i]], 0]; fmv[[i]]=Flat-
  ten[fmv[[i]]]]];

For[i=nt+1, i≤2nt, i++, agsvtran[[i]]=
Flatten[Table[pushdown[fmv[[j]], i−nt −1], {j, ns}]]];

agsv=Transpose[agsvtran];
a=agsv; at=Transpose[a]; b=qmgsv;
```

The following portion of the code uses a least squares singular-value decomposition computation to solve for the first column vector of the $[U_0]$ matrix and the $[F_M]$ matrix.

```
 u, md, v}=SingularValues[a]; ut=Transpose[u];
apsu=PseudoInverse[a];
xcalc=apsu.b;
uov=Take[xcalc, nt];
uo=toeplitzexp[uov];
fov=Take[xcalc, −nt];
fo=toeplitzexp[fov];
```

The following portion of the code computes $[T^i]$, the first column vector of the correction matrix for each storm, using forward substitution.

```
tivcalc=Table[0, {i, ns}, {j, nt}];
For[st=1, st≤ns, st++, k=kri[[st]]; cronk[k];
xv=qm[[st]]+(fo.ckv); nt=Length[xv];
yv=uo.pg[[st]];
yr=Delete[yv,1]; yr=Flatten[Append[yr, 0]];
uvcalc=Table[0, {i nt}, {j, 1}];

uvcalc[[1]]=xv[[1]]/yv[[1]]; utoep=toeplitzexp[uvcalc];

For[st=2, s≤nt, s++,
  uvcalc[[s]]=N[(xv[[s]]−(utoep[[s−1]].yr))/yv[[1]]];
  utoep=toeplitzexp[uvcalc]];

'vcalc[[st]]=uvcalc];
ıvcalctran=Transpose[tivcalc];
```

The following portion of the code prints the output.

```
Print["Output"];
Print["\n\n U-MATRIX ELEMENTS"]; Print[Matrix-
Form[uov]];
Print["\n\n F-MATRIX ELEMENTS"]; Print[Matrix-
Form[fov]];
Print["\n\n"];
For[i=1, i≤ns, i++, Print["T-MATRIX ELEMENTS:
STORM", i];
Print[MatrixForm[tivcalc[[i]]]]];
```

6. Use of the best estimator model

The estimated runoff, Q_M, for an assumed rainfall vector, P_g^* (and its associated unit pulse vector, δ_k^*), for a specific watershed is then determined using the watershed's "best estimator model" and the following equations:

$$Q_M=\{Q^1, Q^2, \ldots, Q^{ns}\} \tag{14}$$

where

$$Q^i=[T^i][U_0]P_g^*-[F_M]\delta_k^* \text{ for } i=1, 2, , \ldots, ns$$

Thus, the estimated runoff Q_M is a discrete stochastic distribution of runoff hydrograph estimates based on the use of all correction matrices. In Eq. (14), the superscript asterisk indicates a forecast storm event.

7. Conclusion

A *Mathematica* computer program has been prepared which calibrates a unit hydrograph rainfall-runoff model based on its rainfall-runoff data, and calculates the watershed's "best estimator model" parameters, matrices $[U_0]$, $[F_M]$, and $[T^i]$.

The "best estimator model" can then be used to approximate the watershed's runoff hydrograph distribution for an assumed forecast rainfall event. This post processing could also be programmed with *Mathematica*, and the output includes graphs.

Acknowledgements

Acknowledgments are paid to the County of San Bernardino, California, and the 1996 Masters Project Team for the Applied Mathematics Department at California State University, Fullerton. This paper is a summary of the 1996 Masters Project.

Appendix A

In order to run the computer code using the following explicit instructions, the following files and folders must be used.

The historical storm data is placed in a *MicroSoft Notepad* text file, entitled "historicstormdata1.txt", which is stored in a folder entitled "My Documents" on the "C" drive. The prescribed format for the text file is illustrated in the text cases. (Note: it is very convenient to initially prepare a *MicroSoft Excel* spreadsheet.xls file, and then name it as a *Notepad* text file).

The *Mathematica* file with the computer code is then opened.

The following two preliminary setup cells, then must be individually evaluated (by pressing shift and enter with the cursor positioned at the end of the cell to be evaluated).

When the second setup cell is evaluated correctly, the cell output will be a line input listing. (Note: sometimes the cell wording must be deleted and retyped in order to activate the cell output).

Change the value of *nt* on the first line of the computer code from 8 to the proper value for the input (highest value of intervals for all of the input storms).

Execute the computer code by evaluating the cell in which the computer code is placed (by pressing shift and enter with the cursor positioned at the end of the cell).

Preliminary setup

```
SetDirectory["C:\My Documents"]
!! historicstormdata1.txt
```

Computer code

```
nt=8;
Stormdata=OpenRead["historicstormdata1.txt"];
Skip[stormdata, Word]; ns=Read[stormdata, Number];

pg=Table[0, {i, ns}, {j, nt}];
qm=Table[0, {i, ns}, {j, nt}];
kri=Table{0, {ns}];
fmv=Table[0, {i, ns}, {j, nt}];
For[h=1, h≤ns, h++,
Skip[stormdata, {Word, Word}]; st=Read[stormdata,
Number];
Skip[stormdata, Word]; ni=Read[stormdata, Number];
Skip[stormdata, {Word, Word, Word}];
temp=ReadList[stormdata, {Number, Number, Number},
ni];
Skip[stormdata, {Word, Word, Number}];
trantemp=Transpose [temp];

uov=Take[xcalc, nt];
uo=toeplitzexp[uov];
fov=Take[xcalc, −nt];
fo=toeplitzexp[fov];

tivcalc=Table[0, {i, ns}, {j, nt}];
For[st=1, st≤ns, st++, k=kri[[st]]; cronk[k];
xv=qm[[st]]+(fo.ckv); nt=Length[xv];
yv=uo.pg[[st]];
yr=Delete[yv,1]; yr=Flatten[Append[yr, 0]];
uvcalc=Table[0, {i nt}, {j, 1}];

uvcalc[[1]]=xv[[1]]/yv[[1]];
utoep=toeplitzexp[uvcalc]];

For[s=2, s≤nt, s++,
  uvcalc[[s]]=N[(xv[[s]]−utoep[[s−1]].yr)/yv[[1]]];
  utoep=toeplitzexp[uvcalc]];

tivcalc[[st]]=uvcalc];
tivcalctran=Transpose[tivcalc];

Print["Output"];
Print["\n\n U-MATRIX ELEMENTS"]; Print[Matrix-
Form[uov]];
Print["\n\n F-MATRIX ELEMENTS"]; Print[Matrix-
Form[fov]];
Print["\n\n"];
For[i=1, i≤ns, i++, Print["T-MATRIX ELEMENTS:
STORM", i];
  Print[MatrixForm[tivcalc[[i]]]]];
```

References

Hromadka II, T.V., Whitley, R.J., 2000. On formalization of HEC-1 and related systems. ASCE Journal of Hydrologic Engineering (submitted for publication).

Wolfram, S., 1996. The *Mathematica* Book, 3rd ed. Wolfram Media (*Mathematica* Version 3).

Hromadka II, T.V., Whitley, R.J., 1989. Stochastic Integral Equations in Rainfall-Runoff Modeling. Springer-Verlag.