Modelling of a dimensionless synthetic unit hydrograph

Gaddi Ngirane-Katashaya[1]* and Robert Cowen[2]

[1] Dept Earth Sciences, University of Venda, P/Bag X5050, Thohoyandou 0950, South Africa

[2] Department of Mathematics, University of Botswana, P/Bag 0022, Gaborone, Botswana

Abstract

When use of a synthetic unit hydrograph on ungauged and non-comparable catchments is required, a further complication is how to choose a suitable time-step to use. The computed time-step is rarely an integral multiplier or divider of the rainfall time-step thus necessitating reanalysing the rainfall chart, at the new time-step. It would be easier if the time-step of the unit hydrograph could be expressed as a continuous function.

In February 1995 there was a flash flood in mid-Eastern Botswana which resulted in all gauges being washed away. The only available data were from upland recording rain gauges from which the reconstruction of the flood hydrograph had to be done. Use of a dimensionless synthetic unit hydrograph (DSUH) was indicated. This is the first part of a two-part paper which involves analyticising (transforming a numerical/graphical function into a continuous one i.e. rendering it analytical) the dimensionless unit hydrograph i.e. developing a piecewise/zonal continuous function of time vs. the runoff.

A DSUH was piecewise modelled using usual statistical methodology and special mathematics software. In spite of having had to use several and basically different functions from piece/zone to piece/zone, the final goodness of fit was exceptional. Demonstrating the application results is shown in the unit hydrograph-based mathematical model which was developed to solve for the above flood. This will be presented in a follow-up paper, which will dwell more on the hydrology of the exercise.

Introduction

Using a unit hydrograph (UH) to compose a catchment runoff hydrograph has always relied on working from the catchment's typical hydrographs produced from rainfall of known intensity and duration, one of the principles on which unit hydrographs are based (Sherman, 1932; Linsley et al., 1982/1988). From this one can work out instantaneous UHs and UHs of various durations to suit rainfall of various durations. Although it is known that modelling has taken over from use of UH whose basic assumptions are gross, for economical and urgency reasons where accuracy is not so crucial, one may quickly get an approximate result which might serve the purpose.

Normally, one resorts to synthetic unit hydrographs (SUHs) if there are no observed discharge hydrographs. This requires that the catchment characteristics (Snyder, 1938; Linsley et al., 1982/1988) be obtained or determined which are then used to adapt the SUH to suit a particular catchment. The catchment/basin lag-time and other catchment characteristics are required to customise the UHs to that particular catchment (Taylor et al., 1952; Mockus, 1957; Wilson, 1983). Extensive use of several catchment characterisation parameters can produce a reasonable and largely reliable runoff hydrograph suitable for design purposes (FSR, 1975/85).

The time-step of rainfall in a rainfall-runoff hydrograph is important as it is the parameter which determines the time-step of the hydrograph and the runoff input. Irrespective of the duration of the rainfall, the time-step goes a long way in determining the apparent intensity of the rainfall as experienced by the catchment, compared to the real intensity. Only accurate representation of the rainfall intensity relative to the catchment can give a true reflection of the resulting discharge. It is imperative that the time-step used in rainfall, the UH, and eventually the constituted discharge, are the same in size and reflect the true relativeness of the rainfall, UH and runoff. Starting with an SUH, this is usually the cause of doubt and difficulty, as will be explained later.

This paper, therefore, is about analyticising a UH thus rendering it easy to adjust its time-step to conform with that of rainfall thus eliminating one source of uncertainty in UH-rainfall-runoff interaction.

* To whom all correspondence should be addressed.

☎ 082 200 5584; fax 0838 269 1843; e-mail gaddin-k@mailexcite.com

Received 11 September1997; accepted in revised form 23 July 1998.

The flood of Mahalapye-Palapye catchments

In February 1995 there was a severe flood in the Mahalapye and Palapye catchments of Botswana (200 km NE of Gaborone and 100 km SE of Francistown - the Lotsane and Mahalapye Rivers are south-flowing tributaries of the Limpopo River). The river-gauging stations were washed away as were some houses near the rivers. The only data available were the rain-gauge records and of flood marks on trees and bushes in the floodways.

The condition of the catchment was such that the floods came within a day of the start of the rainfall and since the catchments were reasonably steep, with little infiltration capacity, the rivers were back to nil discharge within a few days afterwards. A flash flood situation was thus indicated. This showed that the solution would be obtained by using a unit hydrograph, specifically an SUH as there were no prior records of similar circumstances.

The necessity for modelling the flood was to obtain a flood hydrograph thus being able to reconstruct and delineate the flood zone, flood duration, and thus work out assistance and compensation programmes. Mathematical modelling, deterministic simulation-based (for more accuracy and representativity), would have taken longer and would have been more expensive, contrary to the basic objective of the exercise.

The problem: Use of a dimensionless synthetic unit hydrograph

The commonly used dimensionless synthetic unit hydrograph (DSUH) is the US Soil Conservation Service (USSCS) type (Linsley et al., 1982/1988). It is usually presented as a graph thus leaving the user to determine its dimensions from a graphical representation. This allows the user to set the time-step for sampling to one which will suit the rainfall data and catchment characteristics at hand. Bearing in mind that the dimensional time-steps of the UH are determined using the catchment characteristics and rainfall intensity (in some formulations) the UH characteristics are determined after the rainfall, thus making it impossible to determine the time-step in advance for a one-off event as the present case.

If one is dealing with an instantaneous unit hydrograph (IUH), it is possible to determine the desired time-step mathematically. It is desirable that the extraction of DSUH ordinates be done at a relative time-step which, when multiplied by T_p, gives a time-step equal to that of rainfall. The time-step thus determined is usually in non-integral multiples/dividers of the time to peak T_p.

Transformation of DSUH to catchment unit hydrograph (CUH) requires that each ordinate of DSUH be multiplied by the peak discharge QPK in order to dimensionalise the CUH ordinates (e.g. in m^3/s) and multiply the relative time-steps by T_P in real time units. It is thus imperative that dimensionless time-steps have to be at intervals which, when multiplied by T_P give a time-step similar to that of rainfall. Thus a need to be able to manipulate the time-step is presented.

The solution: Stepless time increment scale

It was realised that in order to achieve the requirements, one had to have a DSUH which is continuous (analytical), thus able to assume any time-step that might be imposed on it. The solution thus lay in the ability of the DSUH to assume the analytical state i.e. UHy = Fn(Xt), where UHy (the DSUH ordinate) is a function of the time-step Xt. In this case n represents the fact that one may need to use several functions in order to represent the SUH curve properly.

With this analytical SUH, and ordinates UHy, one only needs to specify at what time(Xt) one requires UHy. One would get a UHy(Xt) which is the dimensionless SUH ordinate at relative time Xt. Thus if the rainfall time-step is 1 h then the relative time-step (Xut) will be 1 h divided by Tp and the UHy, corresponding dimensionless value would be UHyPk, which would be equal to 1.0 (at Xut is equal to 1.0 i.e. at the peak real time (Tp)). The corresponding CUH ordinates and time-steps would be UHq=UHy.Qpk and Xtq = Xt.Tp (Wilson, 1983).

Subsequent presentation will dwell exclusively on the procedure for analyticising the DSUH as the main objective of this paper, the flood simulation will be in a subsequent paper.

Development and validation of a model for the relationship between UHy and UHxt

The graphical DSUH was digitised as closely as possible, thus deriving 101 time-steps from it. At each of these time-steps there was a corresponding UH ordinate. It was reasonable to assume that a relationship which would reproduce the digital equivalent of the UH would represent the required continuous DSUH function.

The resulting hydrograph closely resembles f(x) = x.exp(x) Walpole and Myers (1990) which indicates that one should try and model the data by functions whose terms are products of powers of x and exponentials. The data were divided into the following four regions:

Region A: initial rise of the hydrograph
Region B: peak region (from mid-rise, peak, to upper-recession - an inverted bell shape).
Region C: mid-recession to mid-lower recession
Region D: beyond mid-lower recession.

Subsequent use of the software *Mathematica* (Wolfram Research Inc., 1993) was to determine the values of the constants when the functions had been determined.

Modelling Region A

We proposed that a model of the form F(x) [as in Eq. (1)] should be fitted to the data points (a cubic polynomial with a little exponential growth added).

$$F(x) = \alpha_1 x + \alpha_2 x e^{kx} + \alpha_3 x^2 + \alpha_4 x^3 \quad (1)$$

where:
α_i and k are constants to be determined

Let the data be represented by $\{(X_i, Y_i): 1 \leq i \leq 101\}$. As solving simultaneous equations involving transcendental functions is extremely difficult, the constant k (multiplier of exponent x) was estimated using computer simulation to be k = 1.57. The constants $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ were then estimated by the method of least squares which we briefly describe:

Choosing the data pairs $\{(X_{2i}, Y_{2i})$ in zone: $0 \leq i \leq 6\}$ we define:

$$J(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = \Sigma(Y - f(X))^2 \quad (2)$$

From: $\delta J/\delta\alpha_1 = \delta J/\delta\alpha_2 = \delta J/\delta\alpha_3 = \delta J/\delta\alpha_4 = 0$, we obtain:

$$\Sigma XY = \Sigma X f(X)$$
$$\Sigma X e^{1.57x} Y = \Sigma X e^{1.57} f(X)$$
$$\Sigma X^2 Y = \Sigma X^2 f(X)$$
$$\Sigma X^3 Y = \Sigma X^3 f(X)$$

which give (*Mathematica*, 1993):
$\alpha_1 = 2.0876$ $\alpha_2 = -1.93324$ $\alpha_3 = 4.23726$ $\alpha_4 = 3.69121$

Modelling Region B

We attempted to model this region by a polynomial and computer simulation showed that at least a quintic (a 5th power polynomial) was necessary. Hence we proposed the model g(x) such that:

$$g(x) = \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \beta_5 x^5 \quad (3)$$

By considering the data points $\{(X_i, Y_i)$ in zone:$15 \leq i \leq 32\}$ and using the method of least squares we obtained (*Mathematica*, 1993):

$\beta_1 = -1.91541$ $\beta_2 = 9.70054$ $\beta_3 = -9.9143$
$\beta_4 = 3.42073$ $\beta_5 = -0.289851$

Modelling Region C

We proposed an exponential model of the form h(x) such that:

$$h(x) = \alpha e^{\beta x} \quad (4)$$

where:
x is a time variable, and α and β are constants.

Pairs of data were chosen and inserted into the equation. This was then solved to obtain the constants. We found that one obtained the closest fit when the equation was solved for $\{X_i, Y_i)$ in zone: $46 < i < 74\}$ which gave (*Mathematica*, 1993):

$\alpha = 6.603689528 \quad \beta = -1.554251744$

Modelling Region D

We proposed a negative exponential model {function K(x)} of the form:

$$k(x) = \alpha e^{\beta x} \tag{4}$$

where x, α, β have the same functional meanings as before and used the same method as for region B. Using $\{(X_i, Y_i)$ in zone: $74 < i < 101\}$ we obtained the following (*Mathematica*, 1993):

$\alpha = 116.9078954 \qquad \beta = -2.252735315$

The overall model

By comparing the value of the obtained functions to the actual data we were able to deduce their respective domain. Thus, this gave us the following model: ϕ

ϕ = {f(x): xε[0.0,< 0.7]} valid in Xt range (zone) 0.0 to < 0.7
= {g(x): xε[0.7, < 1.6]} valid in Xt range (zone) 0.7 to < 1.6
= {h(x): xε[1.6, < 4.15]}valid in Xt range (zone) 1.6 to < 4.15
= {k(x): xε[4.15, 5.0]}valid in Xt range (zone) 4.15 to 5.0

Transition zones

As is the normal practice in engineering, the jump discontinuities of the model (from one zone to another) are amended in such a way that the values for the model in the regions [0.65, 0.75], [1.55, 1.65], and [4.1, 4.2] which are referred to as "shared zones" in the program, are determined by taking fractions of function values which lie on either side of the jump (This aspect was programmed in such a way that unequal fractions of the functions, in the shared zone, could be used to minimise the error within the zone).

Validation of the model

Let:
Y_n = the data; y = mean value of the data;
Z_n = estimate from model z = mean value of estimate from model;

The coefficient of correlation, r, is defined by

$$r = \{\Sigma(Y_n\text{-}y)\,(Z_n\text{-}z)\} \,/\, \{\Sigma(Y_n\text{-}y)^2\, \Sigma(Z_n\text{-}z)^2\} \tag{5}$$

where: $-1.0 \leq r \leq 1.0$

The coefficient of correlation can be interpreted in the following way:

Range of correlation coefficient	0.0 to 0.2	0.2 to 0.4	0.4 to 0.7	0.7 to 0.9	0.9 to 1.0
Strength of a model	very weak model	weak model	moderate model	strong model	very strong model

In our case r = 0.99915, which is considered to be an almost perfect model.

The standard error of estimate, SEE is defined in Eq. (6) which, for our results, is equal to 0.00131.

$$SEE = \{1\text{-}\ r^2\} \,/\, \{N - 2\} \tag{6}$$

Thus our 99.7% confidence interval is r ± SEE = [0.995985, 1.003845]. Due to the large number of data points, it was not necessary to use Fishers Z-transformation (*Mathematica*, 1993). This confidence interval can be interpreted in the following way: If another 101 data points were randomly chosen and this was done 1000 times, we would expect that for 997 times the new value of the coefficient of correlation would exceed 0.99598. See the **Appendices**.

Appendix 1 gives the numerical results obtained after applying typical values to the unit hydrograph function. Note that at this stage our objective was to fit the developed function to the known DUH ordinates.

Appendix 2 gives the subroutine EXPOLYFT of the function which was used to test its performance and as contained in the simulation model.

Conclusions

- An analytical/functional unit hydrograph facilitates the continuous adjustment of time to peak with changing rainfall onset times and subsequently rainfall durations. This conforms more to the assumptions on which use of a UH is based.
- A function of a unit hydrograph can be obtained using various curve fitting techniques thus making a unit hydrograph a continuous function of time and rainfall amount. This facilitates proper setting of the hydrograph time-step according to the conditions prevailing when it was raining.
- The complete hydrological simulation program or only the EXPOLYFT subroutine can be made available by e-mail, on request.

Acknowledgement

We would like to thank Dr E Lungu (University of Botswana) for his assistance on computer modelling of the mathematics involved and Mr B Jay, a government engineer, who supplied the data and other necessary information.

References

FSR (1975/1985) Flood Studies Report, Institute of Hydrology, Natural Environment Research Council, UK 1975, & Review :10-years-on 1985.

LINSLEY RK, KOHLER MA and PAULHUS JLH (1982/1988) - *Hydrology for Engineers* (3rd edn.). ISE McGraw Hill, New York.

MOCKUS V (1957) Use of Storm and Watershed Characteristics in Synthetic Hydrograph Analysis and Application. US Soil Conservation Service [As cited by Linsley et al.].

SHERMAN LK (1932) Streamflow from rainfall by the Unitgraph method. *Eng. News Record* **108** (1932) 501 (cited by Wilson EM)
SNYDER F F (1938) Synthetic unitgraphs *Trans. Am. Geophys. Union*, Part **2**, p 447 [As cited by Linsley et al. and Wilson EM].
TAYLOR AB and SCHWARZ HE (1952) A unit hydrograph lag and peak flow related to basin characteristics. *Trans. Am. Geophys. Union* **33** 235.
WALPOLE R and MYERS R (1990) *Probability and Statistics for Engineers and Scientists* (4th edn.) Macmillan.
WILSON EM (1983) *Engineering Hydrology* (3rd edn). ELBS Macmillan, London.
WOLFRAM RESEARCH INC. (1993) *Mathematica* (Version 2.2.3), USA.

Appendix 1
Results of unit hydrograph data fitted to the model (performance of expolyft)

Old values of X vs old and estimated values of Y, showing errors after fitting to functions FX,GX,HX,KX which describe the unit hydrograph.

XNo.	X[old]	Yo[old]	Ye[estmat]	dY[Ye-Yo]	Err[dY/Yo]	Er[dY/YoMx]	FuncsUsed
1	.0000	.0000	.0000	.0000	-.0001%	-.0001%	[100FX]
2	.0500	.0150	.0109	-.0041	-27.4746%	-.4121%	[100FX]
3	.1000	.0300	.0286	-.0014	-4.5481%	-.1364%	[100FX]
4	.1500	.0500	.0539	.0039	7.8938%	.3947%	[100FX]
5	.2000	.0850	.0873	.0023	2.6607%	.2262%	[100FX]
6	.2500	.1250	.1288	.0038	3.0219%	.3777%	[100FX]
7	.3000	.1800	.1784	-.0016	-.8801%	-.1584%	[100FX]
8	.3500	.2300	.2358	.0058	2.5188%	.5793%	[100FX]
9	.4000	.3000	.3002	.0002	.0638%	.0191%	[100FX]
10	.4500	.3750	.3705	-.0045	-1.1941%	-.4478%	[100FX]
11	.5000	.4450	.4453	.0003	.0655%	.0292%	[100FX]
12	.5500	.5400	.5226	-.0174	-3.2296%	-1.7440%	[100FX]
13	.6000	.6000	.5999	-.0001	-.0176%	-.0105%	[100FX]
14	.6500	.6850	.6843	-.0007	-.0951%	-.0652%	[70F+30G]
15	.7000	.7800	.7803	.0003	.0328%	.0256%	[10F+90G]
16	.7500	.8500	.8510	.0010	.1127%	.0958%	[0F+100G]
17	.8000	.9150	.9060	-.0090	-.9783%	-.8952%	[100GX]
18	.8500	.9550	.9490	-.0060	-.6329%	-.6044%	[100GX]
19	.9000	.9750	.9792	.0042	.4339%	.4230%	[100GX]
20	.9500	.9920	.9967	.0047	.4788%	.4749%	[100GX]
21	1.0000	1.0000	1.0017	.0017	.1709%	.1709%	[100GX]
22	1.0500	.9900	.9946	.0046	.4657%	.4610%	[100GX]
23	1.1000	.9770	.9763	-.0007	-.0766%	-.0748%	[100GX]
24	1.1500	.9500	.9477	-.0023	-.2405%	-.2285%	[100GX]
25	1.2000	.9100	.9104	.0004	.0393%	.0358%	[100GX]
26	1.2500	.8700	.8658	-.0042	-.4827%	-.4200%	[100GX]
27	1.3000	.8200	.8159	-.0041	-.4984%	-.4087%	[100GX]
28	1.3500	.7600	.7628	.0028	.3696%	.2809%	[100GX]
29	1.4000	.7100	.7088	-.0012	-.1643%	-.1166%	[100GX]
30	1.4500	.6500	.6565	.0065	1.0075%	.6549%	[100GX]
31	1.5000	.6100	.6087	-.0013	-.2088%	-.1274%	[100GX]
32	1.5500	.5700	.5683	-.0017	-.2912%	-.1660%	[100G+0H]
33	1.6000	.5300	.5385	.0085	1.6116%	.8541%	[100G+0H]
34	1.6500	.5000	.5082	.0082	1.6393%	.8197%	[0G+100H]
35	1.7000	.4650	.4702	.0052	1.1180%	.5199%	[100HX]
36	1.7500	.4350	.4350	.0000	.0097%	.0042%	[100HX]
37	1.8000	.4000	.4025	.0025	.6286%	.2514%	[100HX]
38	1.8500	.3750	.3724	-.0026	-.6884%	-.2582%	[100HX]
39	1.9000	.3500	.3446	-.0054	-1.5506%	-.5427%	[100HX]
40	1.9500	.3250	.3188	-.0062	-1.9049%	-.6191%	[100HX]
41	2.0000	.3000	.2950	-.0050	-1.6760%	-.5028%	[100HX]
42	2.0500	.2750	.2729	-.0021	-.7575%	-.2083%	[100HX]
43	2.1000	.2550	.2525	-.0025	-.9760%	-.2489%	[100HX]
44	2.1500	.2350	.2336	-.0014	-.5826%	-.1369%	[100HX]
45	2.2000	.2200	.2162	-.0038	-1.7444%	-.3838%	[100HX]

46	2.2500	.2000	.2000	.0000	.0000%	.0000%	[100HX]
47	2.3000	.1850	.1850	.0000	.0249%	.0046%	[100HX]
48	2.3500	.1750	.1712	-.0038	-2.1656%	-.3790%	[100HX]
49	2.4000	.1650	.1584	-.0066	-3.9946%	-.6591%	[100HX]
50	2.4500	.1500	.1466	-.0034	-2.2902%	-.3435%	[100HX]
51	2.5000	.1400	.1356	-.0044	-3.1385%	-.4394%	[100HX]
52	2.5500	.1280	.1255	-.0025	-1.9790%	-.2533%	[100HX]
53	2.6000	.1200	.1161	-.0039	-3.2619%	-.3914%	[100HX]
54	2.6500	.1100	.1074	-.0026	-2.3581%	-.2594%	[100HX]
55	2.7000	.1000	.0994	-.0006	-.6247%	-.0625%	[100HX]
56	2.7500	.0920	.0919	-.0001	-.0597%	-.0055%	[100HX]
57	2.8000	.0850	.0851	.0001	.0828%	.0070%	[100HX]
58	2.8500	.0750	.0787	.0037	4.9463%	.3710%	[100HX]
59	2.9000	.0700	.0728	.0028	4.0351%	.2825%	[100HX]
60	2.9500	.0650	.0674	.0024	3.6608%	.2380%	[100HX]
61	3.0000	.0550	.0623	.0073	13.3483%	.7342%	[100HX]
62	3.0500	.0520	.0577	.0057	10.9237%	.5680%	[100HX]
63	3.1000	.0500	.0534	.0034	6.7352%	.3368%	[100HX]
64	3.1500	.0460	.0494	.0034	7.3419%	.3377%	[100HX]
65	3.2000	.0420	.0457	.0037	8.7747%	.3685%	[100HX]
66	3.2500	.0400	.0423	.0023	5.6737%	.2269%	[100HX]
67	3.3000	.0370	.0391	.0021	5.7000%	.2109%	[100HX]
68	3.3500	.0345	.0362	.0017	4.8836%	.1685%	[100HX]
69	3.4000	.0318	.0335	.0017	5.2808%	.1679%	[100HX]
70	3.4500	.0292	.0310	.0018	6.0824%	.1776%	[100HX]
71	3.5000	.0270	.0287	.0017	6.1482%	.1660%	[100HX]
72	3.5500	.0256	.0265	.0009	3.5824%	.0917%	[100HX]
73	3.6000	.0242	.0245	.0003	1.3820%	.0334%	[100HX]
74	3.6500	.0227	.0227	.0000	.0000%	.0000%	[100HX]
75	3.7000	.0213	.0210	-.0003	-1.3957%	-.0297%	[100HX]
76	3.7500	.0199	.0194	-.0005	-2.3500%	-.0468%	[100HX]
77	3.8000	.0184	.0180	-.0004	-2.2858%	-.0421%	[100HX]
78	3.8500	.0169	.0166	-.0003	-1.5675%	-.0265%	[100HX]
79	3.9000	.0155	.0154	-.0001	-.7014%	-.0109%	[100HX]
80	3.9500	.0141	.0142	.0001	.9964%	.0140%	[100HX]
81	4.0000	.0128	.0132	.0004	2.9354%	.0376%	[100HX]
82	4.0500	.0115	.0122	.0007	6.0051%	.0691%	[100HX]
83	4.1000	.0104	.0113	.0009	8.4529%	.0879%	[100H+0K]
84	4.1500	.0093	.0102	.0009	9.4476%	.0879%	[0H+100K]
85	4.2000	.0084	.0091	.0007	8.2661%	.0694%	[0H+100K]
86	4.2500	.0076	.0081	.0005	6.9155%	.0526%	[100KX]
87	4.3000	.0068	.0073	.0005	6.7646%	.0460%	[100KX]
88	4.3500	.0061	.0065	.0004	6.3382%	.0387%	[100KX]
89	4.4000	.0055	.0058	.0003	5.3753%	.0296%	[100KX]
90	4.4500	.0050	.0052	.0002	3.5653%	.0178%	[100KX]
91	4.5000	.0045	.0046	.0001	2.8144%	.0127%	[100KX]
92	4.5500	.0040	.0041	.0001	3.3448%	.0134%	[100KX]
93	4.6000	.0036	.0037	.0001	2.5957%	.0093%	[100KX]
94	4.6500	.0033	.0033	.0000	.0000%	.0000%	[100KX]
95	4.7000	.0029	.0029	.0000	1.6713%	.0048%	[100KX]
96	4.7500	.0026	.0026	.0000	1.3224%	.0034%	[100KX]
97	4.8000	.0024	.0024	.0000	-1.9269%	-.0046%	[100KX]
98	4.8500	.0021	.0021	.0000	.1439%	.0003%	[100KX]
99	4.9000	.0019	.0019	.0000	-1.1054%	-.0021%	[100KX]
100	4.9500	.0017	.0017	.0000	-1.2448%	-.0021%	[100KX]
101	5.0000	.0015	.0015	.0000	.0000%	.0000%	[100KX]

Yo =YOLD; Yom = yo.mean; Ye =y.estm; Yem = y.estmean] [[10F+90G means in transition zone 10% of functn F value added to 90% of funct G value give the best Ye value]]

```
Yom(mean)=.259465     Yem(mean)=0.259456  TOTVAR(TotVar) S[(Yo-Yom)**2]   =10.25252
 VARMDE(VarMode) S[(Ye-Yem)**2]=10.23095  EXPVAR(ExpctVar) S[(Ye-Yom)**2]=10.23096
ESTMER(EstmErr)  S[(Yo-Ye)**2]=0.001288  TOTERR S[(Yo-Yom)*(Ye-Yem)]     =10.24110
```

Appendix 2

```
   Subroutine EXPOLYFT

C Subroutine EXPOLYFT  evaluates the UH ordinates
C using ExponPolynomial based on USSCS UnitHydrog
C
   SUBROUTINE      EXPOLYFT(XT,YOLD,UHDT,FUNC)
   INTEGER         FNC
   REAL            KX
   CHARACTER       FUNC*9
   DATA FNC,F5,F1 /1,0.5,0.1/
   DATA F2,F3,F4  /0.2,0.3,0.4/
   DATA F5,F6,F7  /0.5,0.6,0.7/
   DATA F8,F9,FX  /0.8,0.9,0.0/
   DATA GX,HX,KX  /0.0,0.0,0.0/
C--Setting XT range & go to appropriate equations
8000  IF(XT.ge.0.0.and.XT.lt.0.65) THEN
      FNC= 1
      GOTO 8010
   ENDIF
   IF(XT.ge.0.65.and.XT.le.0.75) THEN
      FNC   = 12
      GOTO 8010
   ENDIF
   IF(XT.gt.0.75.and.XT.lt.1.55) THEN
      FNC= 2
      GOTO 8020
   ENDIF
   IF(XT.ge.1.55.and.XT.le.1.65) THEN
   NC= 23
      GOTO 8020
   ENDIF
   IF(XT.gt.1.65.and.XT.lt.4.10) THEN
      FNC= 3
      GOTO 8030
   ENDIF
   IF(XT.ge.4.10.and.XT.le.4.20) THEN
      FNC= 34
      GOTO 8030
   ENDIF
   IF(XT.gt.4.20) THEN
      FNC= 4
      GOTO 8040
   ENDIF

C==Function FX models initial rise of hydrograph
C--Range XT:0->le .65 FX; share:>.65->lt.75 GX
8010  CONTINUE
   FXT1  = 2.08760*XT
   FXT2  = 1.93324*XT*EXP(1.57*XT)
   FXT3  = 4.23726*XT**2
   FXT4  = 3.69121*XT**3
   FX    = FXT1 - FXT2 + FXT3 + FXT4
   IF(FNC.eq.12) GOTO 8020
   GOTO 8110

C==Functn GX models mid-rise-peak-mid~recession
C--Range XT:ge.75-le 1.55 GX;share:>.65-<.75FX
C--&  share in range: gt 1.55->lt 1.65 with HX
8020  CONTINUE
   GX1   = 1.915410*XT
   GX2   = 9.700540*XT**2.0
   GX3   = 9.914300*XT**3.0
   GX4   = 3.420730*XT**4.0
   GX5   = 0.289851*XT**5.0
   GX    = - GX1 + GX2 - GX3 + GX4 - GX5
   IF(FNC.eq.12) GOTO 8112
   IF(FNC.eq.23) GOTO 8030
   GOTO 8120
```

```
C==Function HX models midrec->lower-mid~recession
C--Range XT:ge 1.65->le 4.1 HX;share:>1.55- <1.65
C--with GX: & share: > 4.10- < 4.20 with KX
8030  CONTINUE
   HX    = 6.603689528*EXP(-1.554251744*XT)
   IF(FNC.eq.23) GOTO 8123
   IF(FNC.eq.34) GOTO 8040
   GOTO 8130
C==Function KX models the  lower recession (tail)
C--Range XT:ge 4.20 KX:share:>4.10-<4.20 with HX
8040  CONTINUE
   KX    = 116.9078954*EXP(-2.252735315*XT)
   IF(FNC.eq.34) GOTO 8134
   GOTO 8140
C================================================C
C The following routine tests combinations of func
C %ages in SHARED ZONES which give UHT with the
C least error
C---Function FX only
8110  IF(FNC.eq.1) THEN
      UHDT  = FX
      FUNC  = '[100FX]'
      GOTO 8200
   ENDIF
C
C--SHORTHAND FOR EXPLAINING RECURRING STATEMENTS
C  Sh-zone/UnSh>Shared/transition zone; Unshared
C  FValUnShZ---->Function value in unshared zone
C  ErDuModSh---->Error due to mode of share/assign
C  Comp50%ShZ--->Compute at 50% funcs in Sh-zone
C  Z%FVCompUHT-->In Sh-zone % of FVals to make UHT
C  ErDu%Sh------>Error due to  percent  of sharing
C
C==Functions   FX  and GX=======================C
8112  IF(FNC.eq.12) THEN
      SH1 = FX                   ! FValUnShZ
      U1  = ABS((SH1-YOLD)/YOLD) ! ErDuModSh
      SH2 = GX                   ! FValUnShZ
      U2  = ABS((SH2-YOLD)/YOLD) ! ErDuModSh
      SH12= F5*FX+(1.0-F5)*GX    ! Comp50%ShZ
      U12 = ABS((SH12-YOLD)/YOLD)! ErDuModSh

      SHL1= F1*FX+(1.0-F1)*GX    ! Z%FVCompUHT
      UL1 = ABS((SHL1-YOLD)/YOLD)! ErDu%Sh
      SHL2= F2*FX+(1.0-F2)*GX    ! Z%FVCompUHT
      UL2 = ABS((SHL2-YOLD)/YOLD)! ErDu%Sh
      SHL3= F3*FX+(1.0-F3)*GX    ! Z%FVCompUHT
      UL3 = ABS((SHL3-YOLD)/YOLD)! ErDu%Sh
      SHL4= F4*FX+(1.0-F4)*GX    ! Z%FVCompUHT
      UL4 = ABS((SHL4-YOLD)/YOLD)! ErDu%Sh

      SHH6= F6*FX+(1.0-F6)*GX    ! Comp50%ShZ
      UH6 = ABS((SHH6-YOLD)/YOLD)! ErDuModSh
      SHH7= F7*FX+(1.0-F7)*GX    ! Comp50%ShZ
      UH7 = ABS((SHH7-YOLD)/YOLD)! ErDuModSh
      SHH8= F8*FX+(1.0-F8)*GX    ! Comp50%ShZ
      UH8 = ABS((SHH8-YOLD)/YOLD)! ErDuModSh
      SHH9= F9*FX+(1.0-F6)*GX    ! Comp50%ShZ
      UH9 = ABS((SHH9-YOLD)/YOLD)! ErDuModSh

      UM1 = MIN(U1,U2,U12,UL1,UL2,UL3,UL4)
      UMIN= MIN(UM1,UH6,UH7,UH8,UH9)

      IF(UMIN.eq.U1) THEN
      UHDT = SH1
         FUNC = '[100F+0G]'
         GOTO 8200
      ENDIF
```

```
      IF(UMIN.eq.U12) THEN
         UHDT = SH12
         FUNC = '[50F+50G]'
         GOTO 8200
      ENDIF
      IF(UMIN.eq.U2) THEN
         UHDT = SH2
         FUNC = '[0F+100G]'
         GOTO 8200
      ENDIF

      IF(UMIN.eq.UL1) THEN
         UHDT = SHL1
         FUNC = '[10F+90G]'
         GOTO 8200
      ENDIF
      IF(UMIN.eq.UL2) THEN
         UHDT = SHL2
         FUNC = '[20F+80G]'
         GOTO 8200
      ENDIF
      IF(UMIN.eq.UL3) THEN
         UHDT = SHL3
         FUNC = '[30F+70G]'
         GOTO 8200
      ENDIF
      IF(UMIN.eq.UL4) THEN
         UHDT = SHL4
         FUNC = '[40F+60G]'
         GOTO 8200
      ENDIF
      IF(UMIN.eq.UH6) THEN
         UHDT = SHH6
            FUNC = '[60F+40G]'
            GOTO 8200
         ENDIF
         IF(UMIN.eq.UH7) THEN
            UHDT = SHH7
            FUNC = '[70F+30G]'
            GOTO 8200
         ENDIF
         IF(UMIN.eq.UH8) THEN
            UHDT = SHH8
            FUNC = '[80F+20G]'
            GOTO 8200
         ENDIF
         IF(UMIN.eq.UH9) THEN
            UHDT = SHH9
            FUNC = '[90F+10G]'
            GOTO 8200
         ENDIF
         FNC   = 2
         GOTO  8120
      ENDIF
C--Function GX only
8120  IF(FNC.eq.2) THEN
         UHDT  = GX
         FUNC  = '[100GX]'
         GOTO 8200
      ENDIF
C--Functions  GX and HX
8123  IF(FNC.eq.23) THEN
         SH2 = GX                    ! FValUnShZ
         U2  = ABS((SH2-YOLD)/YOLD) ! ErDuModSh
         SH3 = HX                    ! FValUnShZ
         U3  = ABS((SH3-YOLD)/YOLD) ! ErDuModSh
         SH23= F1*GX+(1.0-F1)*HX    ! Comp50%ShZ
         U23 = ABS((SH23-YOLD)/YOLD)! ErDuModSh

         SHL1= F1*GX+(1.0-F1)*HX    ! Z%FVCompUHT
         UL1 = ABS((SHL1-YOLD)/YOLD)! ErDu%Sh
```

```
         SHL2= F2*GX+(1.0-F2)*HX    ! Z%FVCompUHT
         UL2 = ABS((SHL2-YOLD)/YOLD)! ErDu%Sh
         SHL3= F3*GX+(1.0-F3)*HX    ! Z%FVCompUHT
         UL3 = ABS((SHL3-YOLD)/YOLD)! ErDu%Sh
         SHL4= F4*GX+(1.0-F4)*HX    ! Z%FVCompUHT
         UL4 = ABS((SHL4-YOLD)/YOLD)! ErDu%Sh

         SHH6= F6*GX+(1.0-F6)*HX    ! Comp50%ShZ
         UH6 = ABS((SHH6-YOLD)/YOLD)! ErDuModSh
         SHH7= F7*GX+(1.0-F7)*HX    ! Comp50%ShZ
         UH7 = ABS((SHH7-YOLD)/YOLD)! ErDuModSh
         SHH8= F8*GX+(1.0-F8)*HX    ! Comp50%ShZ
         UH8 = ABS((SHH8-YOLD)/YOLD)! ErDuModSh
         SHH9= F9*GX+(1.0-F6)*HX    ! Comp50%ShZ
         UH9 = ABS((SHH9-YOLD)/YOLD)! ErDuModSh

         UM1 = MIN(U2,U3,U23,UL1,UL2,UL3,UL4)
         UMIN= MIN(UM1,UH6,UH7,UH8,UH9)

         IF(UMIN.eq.U2) THEN
            UHDT = SH2
            FUNC = '[100G+0H]'
            GOTO 8200
         ENDIF
         IF(UMIN.eq.U23) THEN
            UHDT = SH23
            FUNC = '[50G+50H]'
            GOTO 8200
         ENDIF
         IF(UMIN.eq.U3) THEN
            UHDT = SH3
            FUNC = '[0G+100H]'
            GOTO 8200
         ENDIF

         IF(UMIN.eq.UL1) THEN
            UHDT = SHL1
            FUNC = '[10G+90H]'
            GOTO 8200
         ENDIF
         IF(UMIN.eq.UL2) THEN
            UHDT = SHL2
            FUNC = '[20G+80H]'
            GOTO 8200
         ENDIF
         IF(UMIN.eq.UL3) THEN
            UHDT = SHL3
            FUNC = '[30G+70H]'
            GOTO 8200
         ENDIF
         IF(UMIN.eq.UL4) THEN
            UHDT = SHL4
            FUNC = '[40G+60H]'
            GOTO 8200
         ENDIF

         IF(UMIN.eq.UH6) THEN
            UHDT = SHH6
            FUNC = '[60G+40H]'
            GOTO 8200
         ENDIF
         IF(UMIN.eq.UH7) THEN
            UHDT = SHH7
            FUNC = '[70G+30H]'
            GOTO 8200
         ENDIF
         IF(UMIN.eq.UH8) THEN
            UHDT = SHH8
            FUNC = '[80G+20H]'
            GOTO 8200
         ENDIF
```

```
        IF(UMIN.eq.UH9) THEN
           UHDT = SHH9
           FUNC = '[90G+10H]'
           GOTO 8200
        ENDIF
        FNC   = 3
        GOTO  8130
     ENDIF
C--Function HX only
8130  IF(FNC.eq.3) THEN
      UHDT  = HX
      FUNC  = '[100HX]'
      GOTO 8200
   ENDIF
C--Functions  HX and KX
8134  IF(FNC.eq.34) THEN
        SH3 = HX                   ! FValUnShZ
        U3  = ABS((SH3-YOLD)/YOLD) ! ErDuModSh
        SH4 = KX                   ! FValUnShZ
        U4  = ABS((SH4-YOLD)/YOLD) ! ErDuModSh
        SH34= F1*HX+(1.0-F1)*KX    ! Comp50%ShZ
        U34 = ABS((SH34-YOLD)/YOLD)! ErDuModSh

        SHL1= F1*HX+(1.0-F1)*KX    ! Z%FVCompUHT
        UL1 = ABS((SHL1-YOLD)/YOLD)! ErDu%Sh
        SHL2= F2*HX+(1.0-F2)*KX    ! Z%FVCompUHT
        UL2 = ABS((SHL2-YOLD)/YOLD)! ErDu%Sh
        SHL3= F3*HX+(1.0-F3)*KX    ! Z%FVCompUHT
        UL3 = ABS((SHL3-YOLD)/YOLD)! ErDu%Sh
        SHL4= F4*HX+(1.0-F4)*KX    ! Z%FVCompUHT
        UL4 = ABS((SHL4-YOLD)/YOLD)! ErDu%Sh

        SHH6= F6*HX+(1.0-F6)*KX    ! Comp50%ShZ
        UH6 = ABS((SHH6-YOLD)/YOLD)! ErDuModSh
        SHH7= F7*HX+(1.0-F7)*KX    ! Comp50%ShZ
        UH7 = ABS((SHH7-YOLD)/YOLD)! ErDuModSh
        SHH8= F8*HX+(1.0-F8)*KX    ! Comp50%ShZ
        UH8 = ABS((SHH8-YOLD)/YOLD)! ErDuModSh
        SHH9= F9*HX+(1.0-F6)*KX    ! Comp50%ShZ
        UH9 = ABS((SHH9-YOLD)/YOLD)! ErDuModSh

        UM1 = MIN(U3,U4,U34,UL1,UL2,UL3,UL4)
        UMIN= MIN(UM1,UH6,UH7,UH8,UH9)

        IF(UMIN.eq.U3) THEN
           UHDT = SH3
           FUNC = '[100H+0K]'
           GOTO 8200
        ENDIF
        IF(UMIN.eq.U34) THEN
           UHDT = SH34
           FUNC = '[50H+50K]'
           GOTO 8200
        ENDIF
        IF(UMIN.eq.U4) THEN
           UHDT = SH4
           FUNC = '[0H+100K]'
           GOTO 8200
        ENDIF
        IF(UMIN.eq.UL1) THEN
           UHDT = SHL1
           FUNC = '[10H+90K]'
           GOTO 8200
        ENDIF
        IF(UMIN.eq.UL2) THEN
           UHDT = SHL2
           FUNC = '[20H+80K]'
           GOTO 8200
        ENDIF
       IF(UMIN.eq.UL3) THEN
           UHDT = SHL3

           FUNC = '[30H+70K]'
           GOTO 8200
        ENDIF
        IF(UMIN.eq.UL4) THEN
           UHDT = SHL4
           FUNC = '[40H+60K]'
           GOTO 8200
        ENDIF


        IF(UMIN.eq.UH6) THEN
           UHDT = SHH6
           FUNC = '[60H+40K]'
           GOTO 8200
        ENDIF
        IF(UMIN.eq.UH7) THEN
           UHDT = SHH7
           FUNC = '[70H+30K]'
           GOTO 8200
        ENDIF
        IF(UMIN.eq.UL8) THEN
           UHDT = SHL8
           FUNC = '[80H+20K]'
           GOTO 8200
        ENDIF
        IF(UMIN.eq.UH9) THEN
           UHDT = SHH9
           FUNC = '[90H+10K]'
           GOTO 8200
        ENDIF
        FNC   = 4
        GOTO  8140
     ENDIF


C--Function KX only

8140  IF(FNC.eq.4) THEN
        UHDT  = KX
        FUNC  = '[100KX]'
        GOTO 8200
     ENDIF
C================================================C
8200  CONTINUE
8201  RETURN
      END
C==END OF SUBROUTINE EXPOLYFT==================
C
```